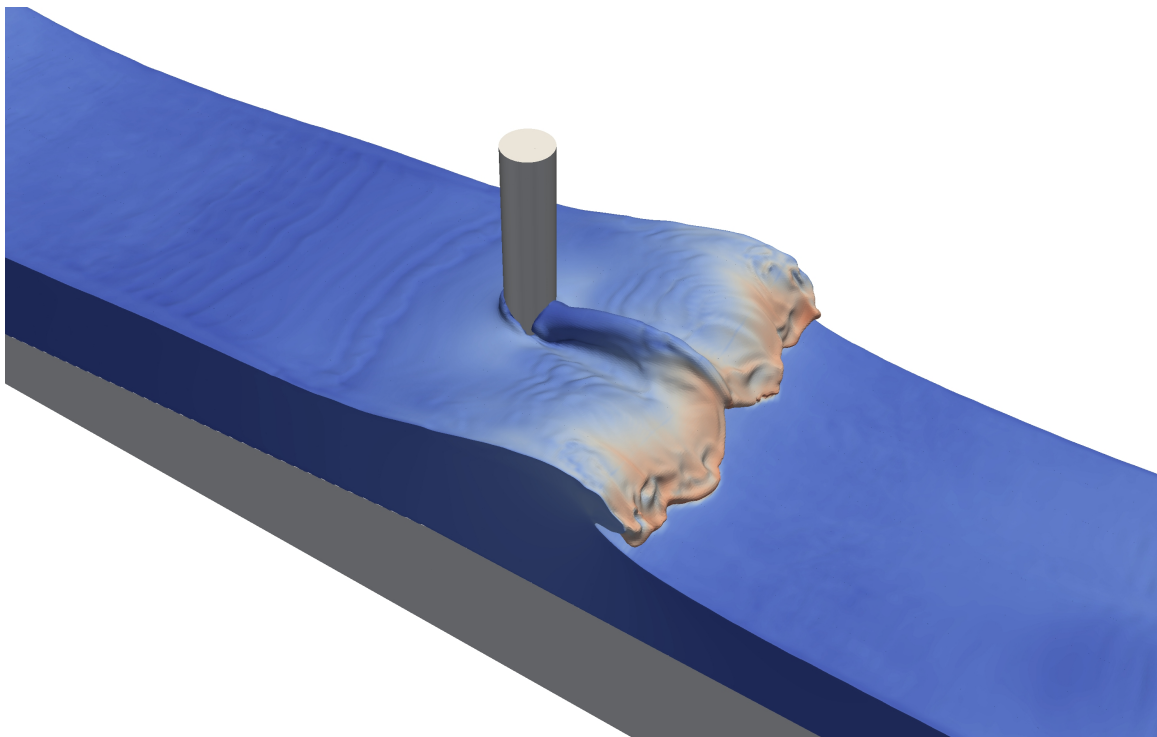


REEF3D :: User Guide



Contact

Hans Bihs

Associate Professor
Marine Civil Engineering
NTNU Trondheim

hans.bihs@ntnu.no

Team

Arun Kamath
Mayilvahanan Alagan Chella
Nadeem Ahmad
Ankit Aggarwal
Weihzi Wang
Adria Moreno Miquel

Contents

1	Introduction	1
1.1	User Expectations & Feedback	2
2	Key Modeling Features of REEF3D	3
3	Numerical Model Overview	7
3.1	Governing Equations	7
3.2	Turbulence Modeling	7
3.3	Convection Discretization	9
3.4	Time Treatment	10
3.5	Pressure	10
3.6	Level Set Method	11
4	Functions	13
4.1	B :: Boundary Conditions	13
4.2	C :: Concentration	26
4.3	D :: Discretization	28
4.4	F :: Free Surface	30
4.5	G :: Grid Options	39
4.6	H :: Heat Transfer	39
4.7	I :: Initialization	41
4.8	M :: MPI	42
4.9	N :: Numerical Options	42
4.10	P :: Printer	47
4.11	S :: Sediment	55
4.12	T :: Turbulence	61
4.13	W :: Water and Material Properties	65
4.14	X :: 6DOF	66
5	Compiling the Code	71
5.1	Installation: Overview	71
5.1.1	Quick Installation Guide	71
5.1.2	User Installation: Makefile	71
5.1.3	Developer Installation: CodeLite	71
5.2	External Libraries	72
5.2.1	MPI	72
5.2.2	HYPRE	72

5.3	Installing REEF3D on Mac OS X and Linux	72
5.3.1	Installing GCC GNU compiler on a Mac	72
5.3.2	Installing OpenMPI	73
5.3.3	Installing HYPRE	73
5.3.4	User version: Building REEF3D and DiveMesh	74
5.3.5	Developer Version: Settings in Codelite	74
5.4	Installing REEF3D on Windows 7	76
5.5	Installing REEF3D on Windows 8	80
5.6	Installing REEF3D on Windows 10	80
5.6.1	Getting Unix Terminal on Windows 10	80
5.6.2	Installing GCC compilers, OpenMPI and Hype	80
6	Post-Processing	83
6.1	Paraview	83
6.1.1	Generating screenshots and animations from Paraview	87
7	Tutorial	89
7.1	2D Dam Break	90
7.1.1	DIVEMesh: control.txt	90
7.1.2	REEF3D: ctrl.txt	90
7.1.3	Results	90
7.2	3D Dam Break with Obstacle	92
7.2.1	DIVEMesh: control.txt	92
7.2.2	REEF3D: ctrl.txt	92
7.2.3	Results	92
7.3	Flow through a Narrow Contraction	94
7.3.1	DIVEMesh: control.txt	94
7.3.2	REEF3D: ctrl.txt	94
7.3.3	Results	95
7.4	Flow around a Circular Pier	97
7.4.1	DIVEMesh: control.txt	97
7.4.2	REEF3D: ctrl.txt	97
7.4.3	Results	98
7.5	Rectangular Wave Tank	98
7.5.1	DIVEMesh: control.txt	99
7.5.2	REEF3D: ctrl.txt	99
7.5.3	Results	100
7.6	Wave Propagation over a Submerged Bar	100
7.6.1	DIVEMesh: control.txt	101
7.6.2	REEF3D: ctrl.txt	101
7.6.3	Result	102
8	Publications with REEF3D	105
8.1	Journal Articles	105
8.2	Conference Papers	106
8.3	Reports	109

Chapter 1

Introduction

Today's 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 [13] 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.

1.1 User Expectations & Feedback

REEF3D is still a fairly young project, compared to other CFD programs out there. Up to now there have been only few external users. For us, the code works well gives good results, especially when it comes to the numerical wave tank. But of course we have been developing and using the code for a while. For new users, some aspects of REEF3D may be difficult to learn at first, but we are working hard to improve the code as well as documentation and tutorials available online. If you like, drop us some lines of feedback here: hans.bihs@reef3d.com

Chapter 2

Key Modeling Features of REEF3D

- **Numerical Grid**

- Cartesian Grid
- Ghostcell Immersed Boundary
- STL file import
- natural topography with xyz points
- basic geometric shapes

- **Convection Discretization with Conservative Finite Differences**

- WENO (5th-order weighted essentially non-oscillatory scheme)
- TVD (2nd-order TVD limiter schemes)
- SMART (2nd-order non-TVD limiter scheme)
- QUICK (2nd-order quadratic upwinding)
- CDS (2nd- and 4th-order central differences)
- FOU (1st-order upwinding)

- **Convection Discretization with Conservative Finite Differences**

- 3rd-order TVD Runge-Kutta scheme
- 2nd-order TVD Runge-Kutta scheme
- 4th-order Runge-Kutta scheme
- 2nd-order Adams-Bashforth scheme
- 2nd-order Implicit Euler scheme
- 1st-order Implicit Euler scheme
- steady state with local time stepping

- **Free Surface Calculation**

- Level Set Method
- Particle Level Set Method
- Volume-of-Fluid Method

- **Pressure**

- Projection Method
- SIMPLE
- PISO
- Incompressible
- Compressible

- **Parallel Iterative Solver**

- Block-Jacobi
- SIP
- Preconditioned BiCGStab
- HYPRE's conjugate gradient solvers [3][5]
- HYPRE's geometric multigrid [3][1]
- HYPRE's algebraic multigrid [3][6]

- **Parallelization**

- Domain Decomposition with MPI

- **Parallel Output**

- ParaView parallel binary file format (.pvtu)
- free surface extraction (.pvtp)
- wave gauges
- free surface lines
- probe points
- probe lines
- flow and wave forces
- binary state files (.r3d)

- **Numerical Wave Tank**

- shallow water wave generation
- intermediate water wave generation
- deep water wave generation
- 2nd-order Stokes wave generation
- 5th-order Stokes wave generation

- 1st-order Cnoidal Waves
- 5th-order Cnoidal Waves
- 1st-order Solitary Wave
- 3rd-order Solitary Wave
- 1st-order Irregular Waves
- 1st-order Focussed Waves
- 2nd-order Irregular Waves
- 2nd-order Focussed Waves
- Wavemaker theory
- Numerical beach with velocity damping
- Numerical beach with relaxation method
- Active wave absorption
- Calculation of wave forces

- **Turbulence Modeling**

- k - ω model
- k - ϵ model
- URANS
- EARSM
- LES

- **Floating Body**

- 6DOF Algorithm
- Immersed Boundary: no remeshing

- **Sediment Transport**

- Bed Load Transport
- Suspended Load Transport
- Bed shear stress on sloping bed
- Sand slide algorithm

- **Multiphase Flow**

- immiscible: multiple level set functions
- two- and three-phase flow
- miscible: concentration functions

- **Surface Tension**

- CSF-Model

- **Heat Transfer**

- Convection-diffusion equation
- in combination with free surface flow
- **Initialization**
 - Potential Flow Solver

Chapter 3

Numerical Model Overview

In this chapter a basic overview over the numerical methods used in REEF3D is given. The goal is to give a rough idea about the relevant numerical approaches and references for further information for the interested reader. The feature set supported by REEF3D is quite large as of today, and for this user's guide only some of the core technologies will be presented.

3.1 Governing Equations

The Navier-Stokes equations describe the behavior of viscous and incompressible fluids. Their formulation is based upon the momentum conservation principle. The left hand side contains the transient and the convective terms. The right hand side contains volume and surface forces. The viscous and the pressure terms are surface forces and gravity is a volume force. The Reynolds stress term requires separate modeling in order to close the system of equations. The turbulence models are further discussed in section 3.2. The governing equations for the mass and momentum conservation are the continuity and the Reynolds-averaged Navier-Stokes (RANS) equations:

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (3.1)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[\nu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \overline{u_i u_j} \right] + g_i \quad (3.2)$$

u is the velocity averaged over the time t , x is the spatial geometrical scale, ρ is the water density, ν is the kinematic viscosity, P is the pressure, g is the gravity, u is the velocity fluctuation over time with $\overline{u_i u_j}$ representing the Reynolds stresses.

Because of REEF3D's focus on CFD in hydraulic and marine engineering, the encountered fluid velocities are mostly too low to have a significant influence on the compressibility on the water and also the air. The governing equations can thus be simplified as incompressible RANS and continuity equations.

3.2 Turbulence Modeling

For many simulations scenarios, RANS turbulence modeling is very relevant, because for the spatial and temporal scales encountered in applications in marine and hydraulic engineering LES (Large-Eddy Simulations) is either not feasible or not necessary. REEF3D has several

RANS turbulence models to choose from, namely the $k-\epsilon$ [9] and the $k-\omega$ [17] turbulence models. From our experience and due to the transient complex nature of the flows we focus on, the $k-\omega$ has shown to give the most robust and accurate results in those cases. The governing equations of the $k-\omega$ model are given. The Reynolds stress term in the the RANS equations is replaced with the Boussinesq-approximation:

$$-\overline{u_i u_j} = \nu_t \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) - \frac{2}{3} k \delta_{ij} \quad (3.3)$$

with

$$\nu_t = c_\mu \frac{k}{\omega} \quad (3.4)$$

Here, k is the turbulent kinetic energy and ω the specific turbulent dissipation. Those two parameters determine the eddy viscosity ν_t and their solutions is found by solving the following transport equations:

$$\frac{\partial k}{\partial t} + u_j \frac{\partial k}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_k} \right) \frac{\partial k}{\partial x_j} \right] + P_k - \beta_k k \omega \quad (3.5)$$

$$\frac{\partial \omega}{\partial t} + u_j \frac{\partial \omega}{\partial x_j} = \frac{\partial}{\partial x_j} \left[\left(\nu + \frac{\nu_t}{\sigma_\omega} \right) \frac{\partial \omega}{\partial x_j} \right] + \frac{\omega}{k} \alpha P_k - \beta \omega^2 \quad (3.6)$$

where P_k is the turbulent production rate, the coefficients have the values $\alpha = \frac{5}{9}$, $\beta_k = \frac{9}{100}$, $\beta = \frac{3}{40}$, $\sigma_k = 2$ and $\sigma_\omega = 2$.

At solid boundaries the surface roughness is accounted for by using Schlichting's rough wall law [14]:

$$u^+ = \frac{1}{\kappa} \ln \left(\frac{30y}{k_s} \right) \quad (3.7)$$

where u^+ is the dimensionless wall velocity, κ a constant of 0.4, y the water depth and k_s the equivalent sand roughness. For the turbulence model the assumption is made, that near the wall the turbulent production is equal to the dissipation of k [17]. The wall function for the specific turbulent dissipation ω for a bed cell with the distance Δy_p from the wall to the center of the cell is then:

$$\omega_{wall} = - \frac{c_\mu^{3/4} k_w^{1/2} U_w^+}{\Delta y_p} \quad (3.8)$$

The formula gives the value for ϵ directly. The turbulent kinetic energy k at the wall is treated by integrating the source terms of Eq.(3.5) over the bed cell:

$$\int (P_k - \epsilon_{wall}) \rho = \left[\frac{\tau_w U_w}{\Delta y_p} - \frac{\rho c_\mu^{3/4} k_w^{3/2} U_w^+}{\Delta y_p} \right] \quad (3.9)$$

The rough wall law is then used to determine the wall shear stress τ_w and the dimensionless u^+ . The terms from Eq. (3.9) are discretized as source terms in the transport equation for k .

3.3 Convection Discretization

The calculation of wave propagation based on the solution of the Navier-Stokes equations can be challenging, because it can suffer from numerical diffusion which consequently leads to an unphysical damping of the waves. So, in the context of CFD models, the order of accuracy of the discretization methods requires special attention. At the same time, numerical stability and efficiency cannot be neglected. In REEF3D, these requirements are balanced by the implementation of the fifth-order WENO (weighted essentially non-oscillatory) scheme in the conservative finite-difference framework [8] for the convection discretization of the flow velocities. The level set function ϕ is discretized by the Hamilton-Jacobi version of the WENO scheme [15]. This scheme can handle large gradients right up to the shock very accurately by taking local smoothness into account. These stencils are weighted depending on their smoothness, with the smoothest stencil contributing the most significantly.

$$u_i \frac{\partial u_i}{\partial x_i} \approx \frac{1}{\Delta x} (\tilde{u}_{i+1/2} u_{i+1/2} - \tilde{u}_{i-1/2} u_{i-1/2}) \quad (3.10)$$

The convection velocity \tilde{u} is obtained at the cell faces through simple interpolation. For the cell face $i + 1/2$, $u_{i+1/2}$ is reconstructed with the WENO scheme based on the weighted sum of three ENO stencils:

$$U_{i+1/2}^{\pm} = \omega_1^{\pm} U_{i+1/2}^{1\pm} + \omega_2^{\pm} U_{i+1/2}^{2\pm} + \omega_3^{\pm} U_{i+1/2}^{3\pm} \quad (3.11)$$

The \pm sign indicates the upwind direction. U^1 , U^2 and U^3 represent the three possible ENO stencils. For upwind direction in the positive i -direction, they are:

$$\begin{aligned} U_{i+1/2}^{1-} &= \frac{1}{3}u_{i-2} - \frac{7}{6}u_{i-1} + \frac{11}{6}u_i, \\ U_{i+1/2}^{2-} &= -\frac{1}{6}u_{i-1} + \frac{5}{6}u_i + \frac{1}{3}u_{i+1}, \\ U_{i+1/2}^{3-} &= \frac{1}{3}u_i + \frac{5}{6}u_{i+1} - \frac{1}{6}u_{i+2} \end{aligned} \quad (3.12)$$

Here, ω_n^{\pm} are non-linear weighting factors. They are determined for each ENO stencil and calculated based on the smoothness indicators IS [8]. The ENO stencil with the smoothest solution will be assigned the largest weight. For stencils with larger gradients, the weights will be smaller.

$$\omega_1^{\pm} = \frac{\alpha_1^{\pm}}{\alpha_1^{\pm} + \alpha_2^{\pm} + \alpha_3^{\pm}}, \quad \omega_2^{\pm} = \frac{\alpha_2^{\pm}}{\alpha_1^{\pm} + \alpha_2^{\pm} + \alpha_3^{\pm}}, \quad \omega_3^{\pm} = \frac{\alpha_3^{\pm}}{\alpha_1^{\pm} + \alpha_2^{\pm} + \alpha_3^{\pm}}, \quad (3.13)$$

and

$$\alpha_1^{\pm} = \frac{1}{10} \frac{1}{(\tilde{\epsilon} + IS_1^{\pm})^2}, \quad \alpha_2^{\pm} = \frac{6}{10} \frac{1}{(\tilde{\epsilon} + IS_2^{\pm})^2}, \quad \alpha_3^{\pm} = \frac{3}{10} \frac{1}{(\tilde{\epsilon} + IS_3^{\pm})^2} \quad (3.14)$$

with the regularization parameter $\tilde{\epsilon} = 10^{-6}$ and the following smoothness indicators:

$$\begin{aligned} IS_1^\pm &= \frac{13}{12}(q_1 - 2q_2 + q_3)^2 + \frac{1}{4}(q_1 - 4q_2 + 3q_3)^2, \\ IS_2^\pm &= \frac{13}{12}(q_2 - 2q_3 + q_4)^2 + \frac{1}{4}(q_2 - q_4)^2, \\ IS_3^\pm &= \frac{13}{12}(q_3 - 2q_4 + q_5)^2 + \frac{1}{4}(3q_3 - 4q_4 + q_5)^2 \end{aligned} \quad (3.15)$$

3.4 Time Treatment

Most complex free surface flow problems can be characterized by their transient nature. Accordingly, also for the discretization in time, high numerical accuracy and stability are required. The explicit third-order TVD Runge-Kutta scheme is an appropriate choice [16]. This method can be applied to the velocities but also to all scalar values, with the exception of the equations of the RANS turbulence models. Here, due to the time step constraining large source terms, an implicit time scheme is recommended. The equations of the third-order Runge-Kutta scheme for the generic variable ϕ are:

$$\begin{aligned} \phi^{(1)} &= \phi^n + \Delta t L(\phi^n) \\ \phi^{(2)} &= \frac{3}{4}\phi^n + \frac{1}{4}\phi^{(1)} + \frac{1}{4}\Delta t L(\phi^{(1)}) \\ \phi^{n+1} &= \frac{1}{3}\phi^n + \frac{2}{3}\phi^{(2)} + \frac{2}{3}\Delta t L(\phi^{(2)}) \end{aligned} \quad (3.16)$$

When used for the time treatment of the momentum equation, the velocities of each Euler sub-step need to fulfill the continuity equation. In order to make them divergence free, the correct pressure is required for the velocity correction. As a result, the Poisson equation for pressure needs to be solved after each Euler sub-step of the scheme. The Poisson equation solution typically takes a significant amount of time. For most cases, except the most demanding in regards of accuracy, the fractional step version of the third-order TVD Runge-Kutta scheme can be used [10]. With this approach, the pressure from the previous time step can be used to correct the velocities after the first two Euler sub-steps. Then, the pressure for the current time step is only computed after the last sub-step. The fractional step method should not be used for cases, where the solid-fluid boundary is dynamic. Otherwise, the situation can occur that the location for the pressure was a solid cell in the previous time step.

3.5 Pressure

In contrast to the other variables of the Navier-Stokes equations, there is no obvious equation which delivers a solution for the pressure. Instead, the solution strategy for the pressure involves the continuity equation. The solution for the pressure is sought in such a way, that the resulting flow field is divergence free. In the context of explicit time treatment, where the solution of the velocity field is based on values from previous time steps, Chorin's projection method [4] can be used. Here, the momentum equation is solved for an intermediate velocity field u_i^* without the pressure gradient:

$$\frac{\partial u_i^*}{\partial t} + u_j^n \frac{\partial u_i^n}{\partial x_j} = \frac{\partial}{\partial x_j} \left[(\nu + \nu_t) \left(\frac{\partial u_i^n}{\partial x_j} + \frac{\partial u_j^n}{\partial x_i} \right) \right] + g_i \quad (3.17)$$

The intermediate velocity u_i^* is not divergence free, or:

$$\frac{\partial u_i^*}{\partial x_i} \neq 0 \quad (3.18)$$

In order for the pressure gradient to create a divergence free flow field, the pressure needs to fulfil the following equation:

$$\frac{\partial}{\partial x_i} \left(\frac{1}{\rho(\phi^n)} \frac{\partial p^{n+1}}{\partial x_i} \right) = \frac{1}{\Delta t} \frac{\partial u_i^*}{\partial x_i} \quad (3.19)$$

Equation 3.19 results in a Poisson equation, which requires the solution of a linear equation system. As the coefficients of the matrix are anisotropic due to the spatially varying density in the case of two-phase flow, the iterative solution of the pressure system is computationally quite demanding. REEF3D offers several iterative solvers. Due to the HYPRE integration (see section 5.3.3), it is possible to combine a conjugate gradient solver with multigrid preconditioning. This results in a fast and scalable solution of the Poisson equation. When the pressure for the new time step p^{n+1} is found, it can be used to correct the intermediate velocity field u_i^* , making it divergence free:

$$u_i^{n+1} = u_i^* - \frac{\Delta t}{\rho(\phi^n)} \frac{\partial p^{n+1}}{\partial x_i} \quad (3.20)$$

3.6 Level Set Method

For most of the areas in which REEF3D is used, the flow is modelled as two-phase system. With the interface capturing level set method [11] complex moving fluid interfaces can be modelled on a fixed Eulerian mesh. The interface between fluids is defined as the zero-contour of the level set function. Away from the interface, the level set function is a signed distance function. This results in the following properties:

$$\phi(\vec{x}, t) \begin{cases} > 0 & \text{if } \vec{x} \in \text{phase 1} \\ = 0 & \text{if } \vec{x} \in \Gamma \\ < 0 & \text{if } \vec{x} \in \text{phase 2} \end{cases} \quad (3.21)$$

The level set function is coupled to the flow field with a pure convection equation:

$$\frac{\partial \phi}{\partial t} + u_j \frac{\partial \phi}{\partial x_j} = 0 \quad (3.22)$$

The spatial discretization of the level set function is mostly performed with the Hamilton-Jacobi version of the WENO scheme [7]. Other options exists and can be selected by the user. For time stepping, the third-order TVD Runge-Kutta scheme can be used [16]. When the level set function is transport by the flow, it can become distorted and can loose its signed distance property. In order to avoid this, it is reinitialized after every time step. The reinitialization scheme is based on the solution of a partial differential equation [12]:

$$\frac{\partial \phi}{\partial t} + S(\phi) \left(\left| \frac{\partial \phi}{\partial x_j} \right| - 1 \right) = 0 \quad (3.23)$$

Chapter 4

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.

4.1 B :: Boundary Conditions

B 5 `int` Rough/Smooth Wall Law

- 1** rough wall law
- 2** smooth wall law
- default:** 1

B 10 `int` Wall functions for the velocities

- 0** OFF
- 1** ON
- default:** 0

B 11 `int` Wall functions for turbulence model

Wall functions can only be used for the RANS turbulence models.

- 0** OFF
- 1** ON
- default:** 0

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

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

B 21 [int](#) Slip or no-slip boundary conditions for k , ω and ϵ

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

B 22 [int](#) Immersed boundary extrapolation

- 1 ON
- 2 OFF
- default:** 1

B 23 [int](#) Immersed boundary for velocity boundary conditions

- 1 ON
- 2 OFF
- default:** 1

B 24 [int](#) Immersed boundary for turbulence boundary conditions

- 1 ON
- 2 OFF
- default:** 1

B 26 [int](#) Boundary condition implementation for level set method

- 1 Neumann
- 2 Extend
- 3 Extend at the bed only
- default:** 1

B 27 [int](#) Wall function formulation for velocities

- 1 including kinetic turbulent energy
- 2 velocities only

default: 1

B 28 **int** Extend pressure along the bed boundary

0 OFF

1 ON

default: 0

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 hydrograph inflow option, make sure that a file with the name "hydrograph.dat" is present in the simulation folder. When selecting the hydrograph 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 62 [int](#) Initialization for rans turbulence variables

0 OFF

1 constant, based on inflow velocity

2 logarithmic profile bed

3 logarithmic profile sidewalls and bed

4 constant with wall function values

5 constant values for NWT

default: 3

B 63 [int](#) Inflow profile for rans turbulence variables

1 zero values

2 logarithmic profile

3 constant

default: 3

B 64 **int** Eddy viscosity damping near inflow

0 OFF

1 ON

default: 1

B 65 **double** Factor for logarithmic inflow profile

default: 1.0

B 66 **double** Factor 1 and **double** Factor 2 for inflow and outflow velocities extending into phase 2

default: 1.6 ; 3.6

B 67 * **double** factor for value, **double** distance, **double** line angle, **double** line x-origin, **double** line y-origin; for use of relaxation method for fixed kinetic turbulent energy

default: na

B 68 * **double** factor for value, **double** distance, **double** line angle, **double** line x-origin, **double** line y-origin; for use of relaxation method for fixed turbulent dissipation

default: na

B 69 * **double** factor for value, **double** distance, **double** line angle, **double** line x-origin, **double** line y-origin; for use of relaxation method for fixed turbulent eddy viscosity

default: na

B 70 * **double** value, **double** distance, **double** line angle, **double** line x-origin, **double** line y-origin; for use of relaxation method for fixed water level

default: na

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](#) Fix pressure at boundaries for constant inflow and wave Generation

- 0 OFF
- 1 inlet
- 2 outlet
- 3 inlet and outlet
- 4 bed
- 5 inlet, outlet and bed
- default:** 1

B 77 [int](#) Pressure outflow boundary conditions

For this parameter to be activated, B 76 needs to be selected in such a way that the pressure outlet boundary condition gets prescribed.

- 1 controlled outflow
- 2 free stream outflow
- 3 controlled outflow for $z \leq B_{80}$, free stream outflow for $z > B_{80}$
- 4 controlled outflow for $z \leq B_{80}$, free stream outflow for $z > B_{80}$; with smoothing
- 5 controlled outflow for water, free stream for air
- default:** 1

B 78 [int](#) Pressure at active wave absorption, based on:

- 1 free surface
- 2 still water level
- 3 zero gradient
- default:** 1

B 79 [double](#) factor for pressure at outlet

default: 1.0

B 80 [double](#) height threshold for pressure at outlet

default: -1.0e8

B 81 **double** x-location for focussed wave, time of wave focussing

default: 0.0 ; 0.0

B 82 **int** Type of focusing amplituted calculation

1 NewWave

2 Spectrum

default: 1

B 85 **int** Wave spectrum for irregular waves

1 Pierson-Moskowitz

2 JONSWAP

3 Torsethaugen

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** Give the pressure from wave theory in the wave relaxation zone

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 amplitude, Wave length, Wave Phase change

The main wave parameters are given here. Alternatively B93 can be used, when the wave period is known.

default: 0.0 ; 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. Wavemaker, irregular and focused waves are in beta.

1 Shallow Water

2 Intermediate

3 Deep Water

4 2nd-order Stokes Waves

5 5th-order Stokes Waves (Felton)

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)

21 Piston Wavemaker

22 Flap Wavemaker

31 1st-order Irregular Waves

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

33 2nd-order Irregular Waves (Schäffer v1)

34 2nd-order Irregular Waves (Schäffer v2)

41 1st-order Focused Waves

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

43 *2nd-order Focused Waves (Schäffer v1)*

44 *2nd-order Focused Waves (Schäffer v2)*

default: 0

B 93 *double* Wave amplitude, Wave period, Wave Phase change

The main wave parameters are given here. Alternatively B91 can be used, when the wave length is known.

default: 0.0 ; 0.0 ; 0.0

B 96 *double* Wave Generation and Beach parameters, dist1, dist2, dist3

When B 98 2 is used, the dist2 still needs to be given, but will not be used by the code. Dist1 and dist2 are the distances from the beginning of the wave flume, dist3 is the distance measured from the end of the wave flume.

default: 0.0 ; 0.0 ; 0.0

B 97 *double* Wave Generation Direction in degree °

default: 0.0

B 98 *int* Wave Generation Method

Relaxation Method 2 ramps up the values for the input wave within one zone. Relaxation Method 1 uses two zones to generate the waves. For the Dirichlet type, waves are generated by assigning the values for the free surface and the velocity in the inflow boundary only. The relaxation type generation methods have shown to result in higher quality waves.

0 OFF

1 Relaxation Method 1

2 Relaxation Method 2

3 Dirichlet

4 Dirichlet with Active Absorption, based on shallow water theory

5 Dirichlet with Active Absorption, based on intermediate water theory

6 Dirichlet with Absorption

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. On the other hand the relaxation methods dissipate the wave energy much better.

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

default: 0

B 100 `double` Exponential factor for beach relaxation method 2

default: 3.5

B 101 `int` Ramp function for wave generation

0 OFF

1 ON

default: 0

B 102 `double` Ramp function factor for duration in wave periods

default: 1.0

B 103 `double` Factor for thickness of relaxation region in the air phase

default: 0.6

B 104 `double` Smoothing factor for thickness of relaxation region in the air phase

default: 3.0

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** angle β , x and y coordinates for the distance origin of numerical beach relaxation zones

default: 0.0 ; 0.0 ; 0.0

B 110 **double** water depth d for wave theory

If not given, the water depth is calculated automatically based on the assumption that the sea bed level is located at $z = 0.0$ m.

default: na

B 118 **double** Exponential Factor for relaxation function method 1

default: 6.0

B 119 **double** Exponential Factor for relaxation function method 2

default: 3.5

B 121 **int** Enable air velocity in wave relaxation zone

0 OFF

1 ON

default: 1

B 126 **double** Z-coordinate for no paddle movement threshold for solitary waves

default: $-1.0e^{20}$

B 140 **double** Numerical Beach: start transition x_1 , start full damping x_2 , β damping factor

default: 0.0 ; 0.0 ; 0.0

B 180 **int** Enable ioGravity for gravity waves

0 OFF

1 ON

default: 0

B 181 **double** x-Gravity Wave frequency, x-Gravity Wave amplitude, x-Gravity Wave Phase change

default: 0.0 ; 0.0 ; 0.0

B 182 **double** y-Gravity Wave frequency, y-Gravity Wave amplitude, y-Gravity Wave Phase change

default: 0.0 ; 0.0 ; 0.0

B 183 **double** z-Gravity Wave frequency, z-Gravity Wave amplitude, z-Gravity Wave 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** Enable periodic boundary conditions

0 OFF

1 ON

default: 0

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 ; 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 265 `double` α resistance coefficient for VRANS

default: 1000.0

B 266 `double` β resistance coefficient for VRANS

default: 1.0

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

default: 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: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}, n, d_{50}$

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

4.2 C :: Concentration

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 2nd-order TVD Runge-Kutta

3 3rd-order TVD Runge-Kutta

11 1st-order Euler Implicit

12 2nd-order Three Time Level Method

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 an 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

4.3 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.

D 10 `int` convection discretization

0 OFF

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: 4

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 time 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 (automatic for implicit convection discretization)**3** implicit diffusion and implicit source term treatment**default:** 1**D 21** [int](#) print out diffusion solver time and iteration for implicit diffusion and explicit convection discretization**0** OFF**1** ON**default:** 0**D 22** [int](#) molecular diffusion in diffusion discretization for momentum equations**0** OFF**1** ON**default:** 1**D 23** [int](#) molecular diffusion in diffusion discretization for RANS turbulence model equations**0** OFF**1** ON**default:** 1**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**11** SIMPLE

12 PISO**default:** 1**D 31** [int](#) Number of correction steps for PISO algorithm**default:** 1**D 32** [int](#) density calculation at the cell faces inside poisson equation

- 1** linear interpolation at cell edges
- 2** Heaviside function at cell edges
- 3** Heaviside function at cell centers with linear interpolation
- 4** Heaviside function at cell centers with 4th-order interpolation

default: 2**D 33** [int](#) density calculation at the cell faces for velocity correction in pressure method

- 1** linear interpolation at cell edges
- 2** Heaviside function at cell edges
- 3** Heaviside function at cell centers with linear interpolation
- 4** Heaviside function at cell centers with 4th-order interpolation

default: 2

4.4 F :: Free Surface

F 30 [int](#) Time scheme Level Set

- 1** 2nd-order Adams-Bashfort
- 2** 2rd-order TVD Runge-Kutta
- 3** 3rd-order TVD Runge-Kutta
- 4** 4th-order Runge-Kutta
- 11** 1st-order Euler Implicit
- 12** 2nd-order Three Time Level Method

default: 0**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 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: 5

F 39 [double](#) Relaxation factor for reini volume constraint

default: 0.5

F 40 [int](#) Reinitialization time scheme

- 1 2nd-order Adams-Bashfort
 - 3 3rd-order TVD Runge-Kutta
 - 4 4th-order Runge-Kutta
 - 5 3rd-order Runge-Kutta with local volume correction
 - 7 3rd-order TVD Runge-Kutta with volume constraint
 - 8 4th-order Runge-Kutta with volume constraint
 - 13 Geometric reinitialization with 3rd-order TVD Runge-Kutta
 - 14 Geometric reinitialization with 4th-order Runge-Kutta
- default:** 0

F 41 `int` Reinitialization after i^{th} time step

default: 1

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 48 `int` Reinitialization after each RK step

0 OFF

1 ON

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: $1.0 \cdot 10^7 \text{ m}$

F 56 `double` Phase 1 end, z-direction

default: $1.0 \cdot 10^7 \text{ m}$

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

default: 0.0 ; 0.0 ; 0.0 ; 0.0

F 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

F 60 `double` Phase 1 end, z-direction for whole domain

default: $-1.0 \cdot 10^{20} \text{ m}$

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

default: $-1.0 \cdot 10^{20} \text{ m}$

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

default: $-1.0 \cdot 10^{20} \text{ m}$

F 63 `double` x_{start} for level set interpolation

default: $-1.0 \cdot 10^{20} \text{ m}$

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

11 1st-order Euler Implicit

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 101 `int` Air entrainment model

0 OFF

1 ON

default: 0

F 150 [int](#) Benchmarks

This will initialize the level set function and the velocity field according to the benchmark cases.

0 OFF

1 Vortex

2 Slotted disc

3 3D Deformation

default: 0

F 300 [int](#) Time scheme level set for multiphase flow

When this parameter is activated, three phase flow will be used. The two level set functions are employed to capture the interfaces between the three different phases.

1 2nd-order Adams-Bashfort

2 2nd-order TVD Runge-Kutta

3 3rd-order TVD Runge-Kutta

4 4th-order Runge-Kutta

11 1st-order Euler Implicit

12 2nd-order Three Time Level Method

default: 0

F 305 [int](#) convection discretization for level set for multiphase flow

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:** 5

F 310 **int** Reinitialization time scheme for multiphase flow

- 1 2nd-order Adams-Bashfort
- 3 3rd-order TVD Runge-Kutta
- 4 4th-order Runge-Kutta
- 5 3rd-order Runge-Kutta with local volume correction
- 7 3rd-order TVD Runge-Kutta with volume constraint
- 8 4th-order Runge-Kutta with volume constraint
- 13 Geometric reinitialization with 3rd-order TVD Runge-Kutta
- 14 Geometric reinitialization with 4th-order Runge-Kutta
- default:** 3

F 321 **double** Factor for the calculation of the Interface Thickness ϵ between Phase 1 and 2

default: 1.6

F 322 **double** Factor for the calculation of the Interface Thickness ϵ between Phase 1 and 3

default: 1.6

F 323 **double** Factor for the calculation of the Interface Thickness ϵ between Phase 2 and 3

default: 1.6

F 350 **int** Fix level set for inflow or outflow for multiphase flow

- 1 Inflow fixed
- 2 Outflow fixed
- 3 Fix Both
- 4 Fix None
- default:** 0

F 360 **double** x-coordinate for interface for level set function 1

default: $-1.0 \cdot 10^{20} \text{ m}$

F 361 **double** y-coordinate for interface for level set function 1

default: $-1.0 \cdot 10^{20} \text{ m}$

F 362 **double** z-coordinate for interface for level set function 1

default: $-1.0 \cdot 10^{20} \text{ m}$

F 370 **double** Rectangular box for positive level set function 1: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

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

F 371 **double** Rectangular box for negative level set function 1: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

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

F 380 **double** x-coordinate for interface for level set function 2

default: $-1.0 \cdot 10^{20} \text{ m}$

F 381 **double** y-coordinate for interface for level set function 2

default: $-1.0 \cdot 10^{20} \text{ m}$

F 382 **double** z-coordinate for interface for level set function 2

default: $-1.0 \cdot 10^{20} \text{ m}$

F 390 **double** Rectangular box for positive level set function 2: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

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

F 391 **double** Rectangular box for negative level set function 2: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

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

4.5 G :: Grid Options

G 50 *int* look for geodata points in grid file

When point coordinates with xyz componets exist for the solid boundary representation, such as for example for a river or coastal region, the interpolation for the topography will take place in DIVEMesh. Then the location of the solid boundary is written into each of the grid files.

0 OFF

1 ON

default: 1

G 52 *int* number of smoothing iterations for geodat points

default: 0

G 60 * *double* Topo 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 ; 0 ; 0

G 61 * *double* Topo ini as wedge in x-direction: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

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

G 95 * *double* Bottom outflow flow: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

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

4.6 H :: Heat Transfer

H 1 *double* Thermal Diffusivity Phase 1

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

H 2 *double* Thermal Diffusivity Phase 2

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

H 10 *int* Time Scheme Heat Transfer

0 OFF

1 2nd-order Adams-Bashfort
3 3rd-order TVD Runge-Kutta
4 4th-order Runge-Kutta
11 1st-order Euler Implicit
12 2nd-order Three Time Level Method
default: 0

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

4.7 I :: Initialization

I 10 [int](#) Initialize Everything

Turning this parameter on will invoke I11, I12 and I13.

0 OFF

1 ON

default: 0

I 11 [int](#) Initialize Velocities with Potential Flow Solver

0 OFF

1 ON

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 20 [int](#) Iterations of the Potential Flow Solver

default: 2500

I 21 [int](#) Set Phase 2 velocities to zero

0 OFF

1 ON

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 P 40 for state file print out.

0 OFF

1 ON

default: 0

I 41 **int** Number of state file (hotstart)

default: 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

4.8 M :: MPI

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

4.9 N :: Numerical Options

N 5 **int** True 2D

0 OFF

1 ON

default: 1

N 8 [int](#) Iterative solver for velocities and scalars

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 Poisson equation with a jump in the matrix coefficients, as is the case for multiphase flow. SIP is an optimized implementation of ILU. This solver is used for the implicit solution of the momentum equations and other scalar equations like the implicit level set function of the implicit solution of the equations of the turbulence model.

1 Block Jacobi

2 SIP

3 BiCGStab

default: 3

N 9 [int](#) Preconditioner

Here the preconditioner for the solvers in CDS matrix format can be specified.

0 OFF

1 Jacobi Scaling

2 Block Jacobi

3 SIP

default: 1

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 Poisson 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 HYPRE 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 [3].

1 Block Jacobi (internal)

2 SIP (internal)

3 BiCGStab (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 Multigrid)

- 21** PCG (HYPRE AIJ)
- 22** GMRES (HYPRE AIJ)
- 23** LGMRES (HYPRE AIJ)
- 24** BiCGStab (HYPRE AIJ)
- 25** AMG (HYPRE AIJ, BoomerAMG)
- 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
- 1** Jacobi Scaling (for internal solvers)
- 2** Block Jacobi (for internal solvers)
- 3** SIP (for internal solvers)
- 11** PFMG (for HYPRE Struct Krylov solvers)
- 12** SMG (for HYPRE Struct Krylov solvers)
- 21** AMG (for HYPRE AIJ Krylov solvers, BoomerAMG)
- default:** 11

N 13 [int](#) Inner Iterations Solver

default: 10

N 15 [int](#) Boundary condition update in between iterations of the solver

- 0** OFF
- 1** ON
- default:** 0

N 17 `double` Relaxation factor for the iterative solver

default: 0.5

N 18 `double` Stopping criteria preconditioner

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

N 40 `int` Time scheme for the momentum equations

The explicit time schemes (option 1-9) require the projection method for the pressure (D 30). Implicit time discretization (option 11 and 12) only works together with the SIMPLE-type schemes.

1 2nd-order Adams-Bashforth

2 2nd-order TVD Runge-Kutta

3 3rd-order Runge-Kutta

4 4th-order Runge-Kutta

6 Fractional Step Method based on 3rd-order TVD Runge-Kutta

7 Fractional Step Method based on 4th-order Runge-Kutta

8 Modified 3rd-order Runge-Kutta

9 Modified Fractional Step Method based on 3rd-order TVD Runge-Kutta

11 1st-order Euler Implicit

12 2nd-order Three Time Level Method

default: 1

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

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: 5250

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

3 ON, using separate velocity and turbulence timestep

4 ON, using minimum velocity and turbulence timestep

11 Local time stepping for steady-state calculation

default: 1

N 49 `double` Timestep Size for Fixed Timestepping

default: 1.0 sec

N 50 `int` Number of Outer Iterations for SIMPLE-type Algorithms

default: 1

N 51 `double` Stopping Criteria for Outer Iterations of SIMPLE-type Algorithms

default: $1.0 \cdot 10^{-5}$

N 52 `int` Include Scalar Transport in Outer Iterations of SIMPLE-type Algorithms

0 OFF

1 ON

default: 1

N 53 `double` Relaxation Factor for the Pressure

default: 1.0

N 54 [double](#) Relaxation Factor for the Velocities

default: 1.0

N 55 [double](#) Relaxation Factor for the Turbulence Model

default: 1.0

N 56 [double](#) Relaxation Factor for the Level Set Method

default: 1.0

N 57 [int](#) Starting Iteration for Increasing N47, [int](#) Ending Iteration for Increasing N47, [double](#) Final N47

default: 0 ; 0 ; 0.3

N 58 [int](#) Enable Timestep Dependend Relaxation Factors

0 OFF

1 ON

default: 0

N 71 [int](#) Order for Extend Value Ghost Cell Extrapolation

default: 2

N 72 [int](#) Order for Dirichlet Ghost Cell Extrapolation

default: 2

4.10 P :: Printer

P 10 [int](#) Print ParaView binary format

0 OFF

1 ON

default: 1

P 11 `int` Log print frequency

default: 10

P 14 `int` Print files to folders

0 OFF

1 ON

default: 1

P 15 `int` Print out file numbering

1 print out based

2 iteration based

default: 1

P 17 `int` Print out warning for incomplete paraview cut cells

0 OFF

1 ON

default: 0

P 18 `int` Algorithm type for level set paraview print out

1 Standard

2 Node Fill

default: 2

P 19 `int` CCIPO algorithm type for level set paraview print out

1 Standard

2 Facet Fill

default: 1

P 20 `int` Print Results every i^{th} Iteration

Choose between either P20, P30, P34 or P35.

default: -10

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 scalar velocity magnitude 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 fb 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, End t, 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 functionality

See I 40 for full initialization from state file. The 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 continous files

default: 0

P 41 **int** Print state file every i^{th} Iteration

default: na

P 42 **double** Print state file every i^{th} second

default: na

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 wsline file

0 OFF

1 ON

default: 0

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

default: 10

P 55 **double** Print wsf files every i^{th} second

default: -1.0

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

default: na

P 59 **int** Print runup for conical island

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 66 **int** print out discharge cross section to console

0 OFF

1 ON

default: 0

P 67 * **double** x-location of discharge in cross section

default: na

P 75 **int** print out vorticity to vtU file

0 OFF

1 ON

default: 0

P 79 **int** print out bed shear stress to vtU file

0 OFF

1 ON

default: 0

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 82 **double** force calculation x-width and y-width

default: 0.0 ; 0.0

P 83 **double** Interface threshold ϵ for force calculation

default: 0.0

P 84 **double** Diameter for normalized vertical force calculation

default: 1.0

P 85 **double** force calculation box for rectangular objects: $x_{start}, x_{end}, y_{start}, y_{end}, z_{start}, z_{end}$

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

P 86 **double** force calculation x-width and y-width

default: 0.0 ; 0.0

P 87 **double** Interface threshold ϵ for force calculation

default: -1.6

P 88 **double** Diameter for normalized vertical force calculation

default: 1.0

P 89 **double** morison force calculation c_M and c_D

default: 0.0 ; 0.0

P 90 **double** Distance from floor to truncated obstacle

default: 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 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 125 `double` x-location and `double` y-location of bed shear stress gauges

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 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

4.11 S :: Sediment

S 10 [int](#) Sediment Transport Module

0 OFF

1 ON

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 morphodynamic calculations is by default decoupled from the hydrodynamics timestep (see S15), which typically is determined through adaptive timestep-ping (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 17 `int` Inner iterations for topo in order to get to full sediment timestep

- 0** OFF
 - 1** ON
- default:** 0

S 18 `int` Maximum inner iterations for topo

default: 100

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 Fall Velocity

default: 0.000001 *m/s*

S 24 `double` Porosity of the Sediment Layer

default: 0.5

S 25 **double** Angle of Repose

default: 30.0 °

S 28 **double** Factor for new fluid cell velocity interpolation

default: 0.5

S 29 **double** Factor for new solid cell velocity interpolation

default: 0.5

S 30 **double** Shields Parameter

default: 0.047

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 38 **int** Reinitialization for interface nodes

0 OFF

1 ON

default: 1

S 39 **int** Time discretization of bedchanges

1 1-st order Euler

2 2nd-order Adams-Bashforth

3 3rd-order Adams-Bashforth

default: 1

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 51 **double** Sediment start, x-direction

default: 0.0 *m*

S 52 **double** Sediment start, y-direction

default: 0.0 *m*

S 53 **double** Sediment start, z-direction

default: 0.0 *m*

S 54 **double** Sediment end, x-direction

default: $1.0 \cdot 10^7$ *m*

S 55 **double** Sediment end, y-direction

default: $1.0 \cdot 10^7$ *m*

S 56 **double** Sediment end, z-direction

default: $1.0 \cdot 10^7$ *m*

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

default: $-1.0 \cdot 10^{20}$ *m*

S 60 **int** Time Scheme Suspended Sediments

1 2nd-order Adams-Bashfort

2 2nd-order TVD Runge-Kutta

3 3rd-order TVD Runge-Kutta

11 1st-order Euler Implicit

12 2nd-order Three Time Level Method

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 fixed water level

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

0 off

1 Parker and Kovacs

2 Dey empirical

3 Dey analytical

default: 0

S 81 **double** Midphi parameter

default: 35.0

S 82 **double** Deltaphi parameter

default: 5.0

S 90 **int** Sandslide algorithm

0 off

1 on

default: 0

S 91 **int** Number of sandslide iterations

default: 1

S 92 **double** Safety factor for sandslide corrections

default: 1.0

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

S 102 **int** Number of outer spatial filter iterations for the bed load discharge for the predictor step (larger numbers increase the filter effect, recommended value: 0)

default: 0

S 103 **int** Number of inner spatial filter iterations for the bed load discharge for the corrector step (larger numbers move the bed load discharge closer to the uncorrected values, recommended value: 0)

default: 0

4.12 T :: Turbulence

T 10 **int** Turbulence Model

0 OFF

1 k- ϵ model

2 k- ω model

11 EARSM with k- ϵ model

12 EARSM with k- ω model

21 URANS with k- ϵ model

22 URANS with k- ω model

31 LES with Smagorinsky SGS model

default: 0

T 11 [int](#) Time Scheme for 2-equation turbulence models

1 2nd-order Adams-Bashfort

2 2nd-order TVD Runge-Kutta

3 3rd-order TVD Runge-Kutta

11 1st-order Euler Implicit

12 2nd-order Three Time Level Method

default: 11

T 12 [int](#) convection discretization for turbulence model

0 OFF

1 FOU

2 CDS

3 QUICK

4 WENO FLUX

5 WENO 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

T 30 [int](#) Type of limiter for eddy viscosity

0 no limiter

1 eddy viscosity is kept bounded

2 limiter for eddy viscosity in addition to boundedness

default: 2

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 33 **double** Factor for free surface turbulence damping in the $k - \omega$ model

Choose between T33 and T36.

default: 0.0

T 34 **int** Influence of mesh size on free surface turbulence damping

1 Δx

2 Δx^2

default: 2

T 35 **double** factor for eddy viscosity limiter near-wall

default: 0.816

T 36 **int** FSF boundary condition for turbulent dissipation

Choose between T33 and T36.

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 39 **double** dirac function factor for free surface turbulence damping

default: 1.0

T 40 **int** Type of limiter for k , ω and ϵ source terms

0 no limiter

1 k , ω and ϵ are kept bounded

2 limiter for turbulent production in k equation in addition to boundedness

3 limiter for turbulent production in ω and ϵ equation in addition to boundedness

default: 1

T 41 **int** Source term definition for ω in $k - \omega$ model

1 Wilcox

2 Durbin

default: 1

T 42 **double** Factor for turbulent production limiter

default: 10.0

T 43 **int** Turbulent production term with strain (standard) or vorticity

1 Strain

2 Vorticity

default: 1

T 51 **double** turbulent intensity at inflow boundary

default: 0.0

T 52 **double** turbulent intensity at generation zone for NWT

default: 0.00001

T 53 **double** factor for turbulent dissipation at generation zone for NWT

default: 1.0

4.13 W :: Water and Material Properties

W 1 *double* Density Phase 1

default: 998.2 kg/m^3

W 2 *double* Viscosity Phase 1

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

W 3 *double* Density Phase 2

default: 1.205 kg/m^3

W 4 *double* Viscosity Phase 2

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

W 5 *double* Surface Tension

default: 0.0 N/m

W 6 *double* Density Phase 3

default: 840.0 kg/m^3

W 7 *double* Viscosity Phase 3

default: $1.5 \cdot 10^{-3} \text{ m}^2/\text{s}$

W 10 *double* Discharge

default: $0.0 \text{ m}^3/\text{s}$

W 20 *double* Gravity, x-component

default: 0.0 m/s^2

W 21 *double* Gravity, y-component

default: 0.0 m/s^2

W 22 **double** Gravity, z-component

default: 0.0 m/s^2

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°

4.14 X :: 6DOF

X 10 **int** Turn on 6DOF algorithm for floating bodies

0 OFF

1 ON

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 turned off, then they can be prescribed with an external linear or angular velocity.

0 OFF

1 ON

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

X 18 **int** Relaxation method for flow velocities in the vicinity of the floating body.

0 OFF

1 ON

default: 0

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 C for rotational motion around 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 solid velocities

default: 3

X 33 **int** Boundary conditions on floating solid for pressure

1 Neumann

2 Extend

default: 2

X 34 **int** Boundary treatment for new floating solid velocity

0 OFF

1 ON

default: 1

X 101 **double** Initial Euler angles: ϕ_0, θ_0, ψ_0

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 120 **double** Sphere floating body: $radius, x_{center}, y_{center}, z_{center}$

default: 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: $x_{length}, y_{length}, z_{length}, x_{center}, y_{center}, z_{center}, x_{roll}, y_{roll}$

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

Chapter 5

Compiling the Code

5.1 Installation: Overview

5.1.1 Quick Installation Guide

1. get a compiler / install cygwin
2. install MPI
3. install HYPRE
4. compile DIVEMesh
5. compile REEF3D

5.1.2 User Installation: Makefile

A makefile is included together with the source code files. After installing MPI and HYPRE, go the location of the source code within the terminal. This is also where the makefile can be found. Then, just type 'make' in order to compile and build the executable. This method of installation is straightforward and is the recommended way for those who want to use REEF3D.

5.1.3 Developer 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 Mac OS, 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 Mac OS X, Windows and Linux
- it works with a variety of compilers
- it works with the MPI library

- support for parallel builds
- no make files are needed

CodeLite can be downloaded here: <http://www.codelite.org>. 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. This method of installation is recommended for those who would like to read and modify the source code, because the source code is nicely organized in a folder structure inside the CodeLite IDE.

5.2 External Libraries

5.2.1 MPI

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 official MPI homepage: <http://www.mpi-forum.org/>. Depending on the operation system, several MPI distributions exist. We recommend OpenMPI for Mac OS X and Linux and MPICH for Windows.

5.2.2 HYPRE

HYPRE [3][5] is a library for high-performance linear solvers and preconditioners and can be downloaded here: http://computation.llnl.gov/project/linear_solvers/. With its underlying structured grid, REEF3D can take advantage of using HYPRE's geometric multigrid options. The geometric multigrid preconditioner PFMG [1] together with one of the conjugate gradient accelerators results in extremely small solution times for the Poisson equation for the pressure.

5.3 Installing REEF3D on Mac OS X and Linux

Here, the installation of REEF3D on Mac OS X is shown. As it is a UNIX based system, installations on Linux should be quite similar. On supercomputers, which are typically UNIX driven, also use the makefile for compilation.

5.3.1 Installing GCC GNU compiler on a Mac

This step is Mac only, Linux should have the compiler installed already.

- Install Xcode.
- Install the command line tools in Xcode, which include gcc and g++:
`xcode-select --install`
- Install gfortran: go this link <https://gcc.gnu.org/wiki/GFortranBinariesMacOS> and download the installer. Install the binary .dmg file.

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

```
gcc -v
g++ -v
gfortran -v
```

5.3.2 Installing OpenMPI

- Download source code - <http://www.open-mpi.org/software/> and unzip the archive.
- Open 'Terminal' and navigate to the location of the archive and type:
(To ensure all the compilers are given MPI support:)
`./configure CC=gcc CXX=g++ F77=gfortran FC=gfortran --prefix=/usr/local/openmpi`
- Next:
`make all` (which will compile and configure)
- Finally, install by:
`sudo make install`
- Add the following openmpi locations to the PATH:
`/usr/local/openmpi`
`/usr/local/openmpi/bin`

- this can be done for example with the pico editor and the following command:
`sudo pico /etc/paths`

Check if added to path using:
`echo $PATH` (should return installation directory).

- If the pico editor does not work, the PATH variable can be appended through the bash profile or bashrc using the following command:
First find out which of the files exist on your machine using:
`ls -a`
Look for files starting with '.bash'. Proceed depending on which file exists.
An example to include the path is shown for the '.bashrc' file below:
`echo 'export PATH=/usr/local/openmpi/bin/:/usr/local/openmpi/lib/:$PATH' >> ~/.bashrc`
- Close the terminal and open a new terminal to check if added to path using:
`echo $PATH` (should return installation directory).

5.3.3 Installing HYPRE

- 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`

note: if HYPRE does not find the FORTRAN compiler, use this configure command:
`./configure --prefix=/usr/local/hypre F77=/usr/local/bin/gfortran`

- Next:
make all (which will compile and configure)
- Finally, install by:
sudo make install
- Add the following HYPRE locations to the PATH:
/usr/local/hypre
/usr/local/hypre/lib

5.3.4 User version: Building REEF3D and DiveMesh

For users who are interested in running the code only, the easiest route is the compilation of REEF3D and DIVEmesh with the makefiles given in the code repository. The makefile contains a so called wildcard option, compiling all source files.

Open terminal and go to the REEF3D directory and type:
make

The code will compile and the executable will be produced in the bin directory. Follow the same steps to compile DiveMesh.

5.3.5 Developer Version: Settings in Codelite

The following instructions are somewhat more involved than the makefile installation. The REEF3D and DIVEmesh compilation is for programmers who are interested in working with and modifying the source code. Codelite works with an internal makefile system, which enables selective compilation of the code. This means, after modifying the source code, only relevant parts of the code are recompiled.

The following path settings are to be set in Codelite to compile REEF3D. First, go to settings > build settings and add new compiler named mpicxx. Rest of the settings 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)

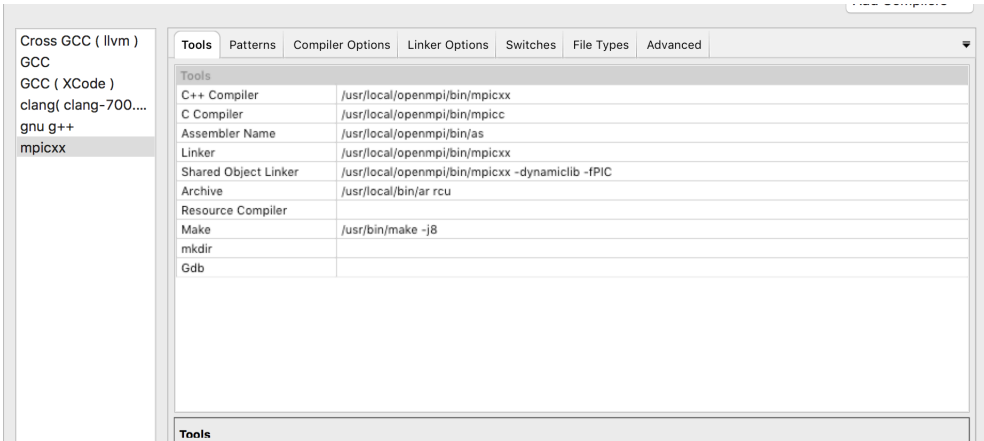


Figure 5.1: Setup of the mpicxx compiler in CodeLite.

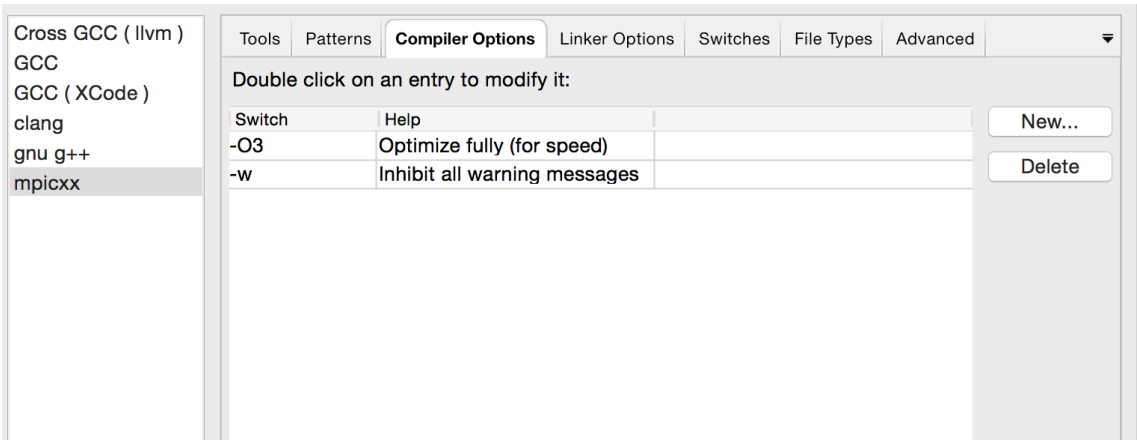


Figure 5.2: Compiler options for mpicxx.

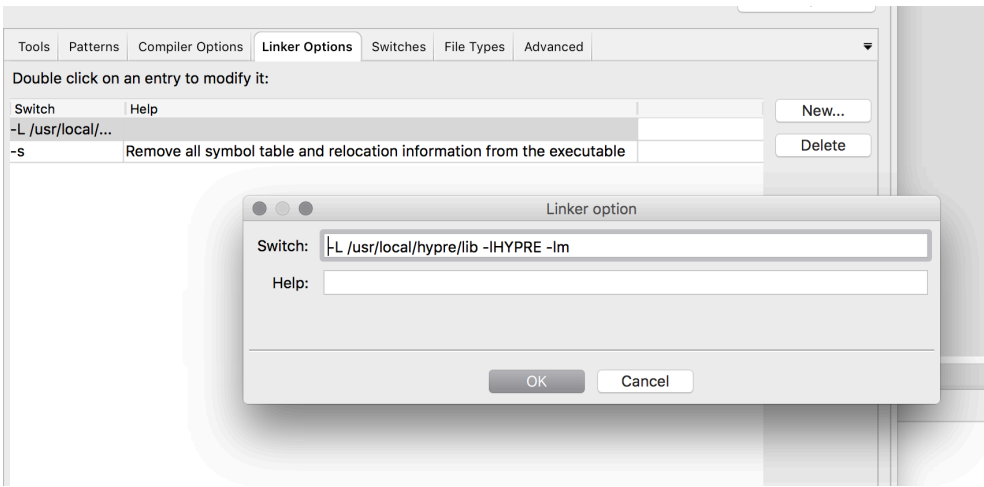


Figure 5.3: Linker options for mpicxx, link to HYPRE.

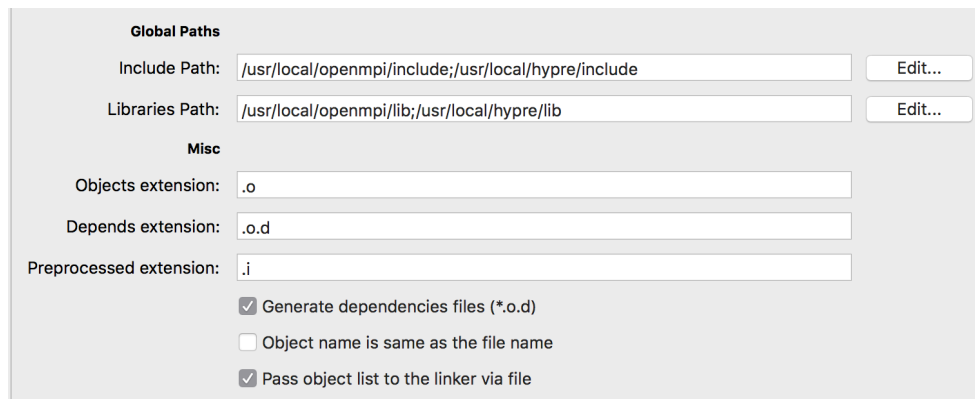


Figure 5.4: Add the Include and Libraries Paths in the advanced tab of the mpicxx compiler

- Compile REEF3D and DiveMesh using Codelite
- Paste the applications REEF3D and DiveMesh to your home folder
- Place the control.txt and ctrl.txt files in the same folder
- Double click on DiveMesh to generate the grid files.
- Open terminal and type for example: `mpiexec -n 4 reef3d` to run the program. The number after `-n` is the number of processors and can be chosen based on the number of physical CPU cores available (NOT threads).

5.4 Installing REEF3D on Windows 7

IMPORTANT: From our experience, the installation of REEF3D on Windows 7 works in most of the cases. But sometimes it can fail due to complexities with Windows security or administration settings. As such, we strongly recommend to run REEF3D under a UNIX/LINUX based system. If that is not possible, the second best option is to update to Windows 10, where an officially supported Linux bash is available. Also keep in mind, that the performance of REEF3D will be much better on a UNIX/LINUX system or on Windows 10 through the Linux bash than on Windows 7.

At the heart of the Windows 7 installation lies the employment of cygwin, which offers a Unix-like terminal and compilers. As a result, the necessary commands to compile and run REEF3D and DIVEMesh are very similar to those for the Mac and Linux. There are also ways to install the code on Windows in a more native fashion, but experience has shown, that the success varies with different Windows versions and levels of administrator rights. Consequently, the cygwin route is the most consistent way to install REEF3D and DIVEMesh on Windows and is the only method explained in this user's guide.

The additional procedures for Windows OS is shown below (we have tested this procedure for Windows 7, 64-bit machine, but also works on Windows 8). Note that you need to have administrator access on the machine for these installations.

- Install cygwin for your architecture (64-bit, 32-bit). The executable for installation can be found at: <http://cygwin.com/>. Remember to install in a folder without a space in its name. For eg. C:\cygwin64 as shown in Fig.5.5

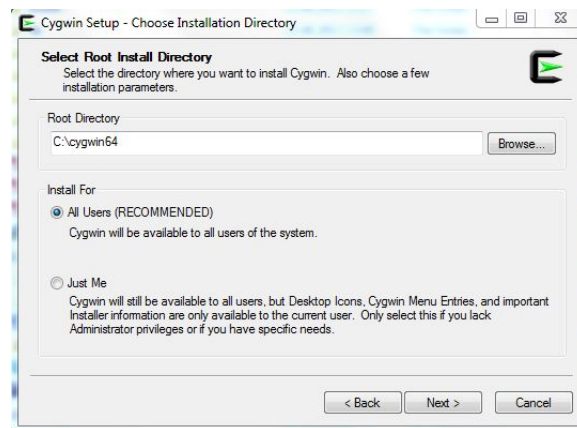


Figure 5.5: Cygwin installation path

- **Very important:** Make sure to install all packages for Admin, Base, Database, Debug, Devel and Libs in the package installation menu. This is done by clicking on 'default' next to the package name and going through the options until it says 'install' as shown in Fig.5.6.

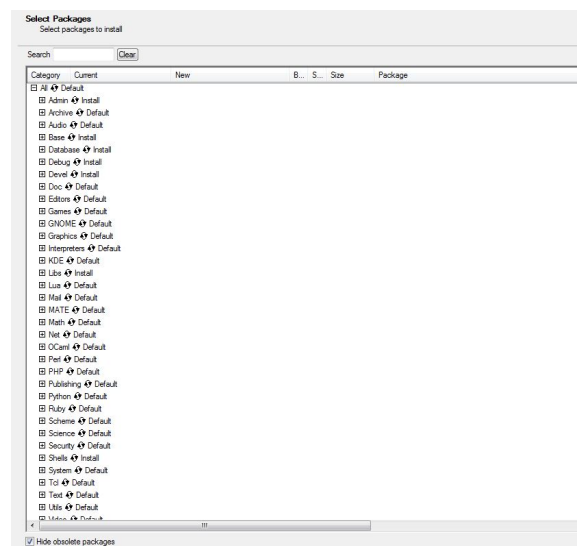


Figure 5.6: Cygwin installation packages

- To set the environment variables for cygwin, click on the start button and right click on My computer> Properties>Advanced (Fig.5.7) and click on Environment Variables at the bottom of the window (Fig.5.8)

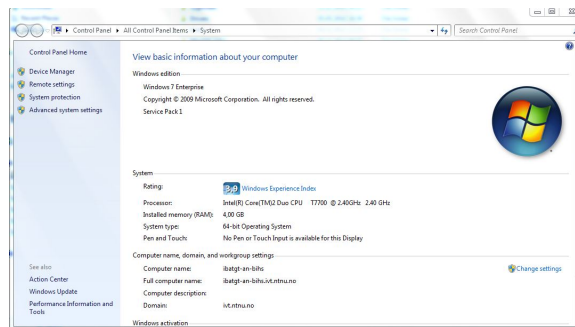


Figure 5.7: Setting environment variables in Windows 7: step1

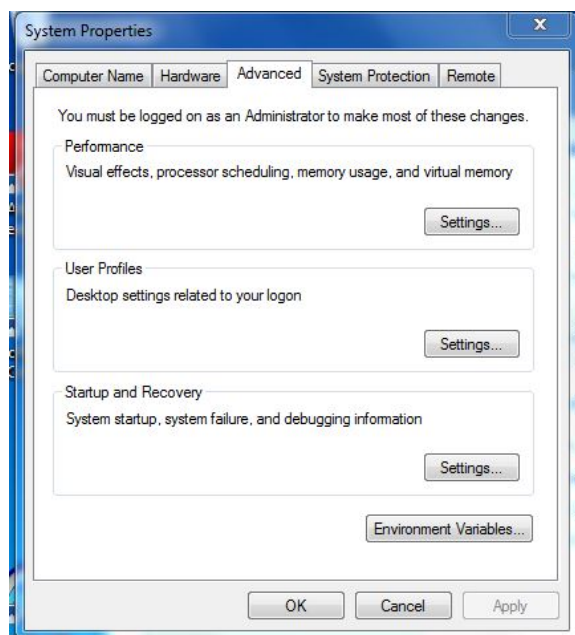


Figure 5.8: Setting environment variables in Windows 7: step2

- Click on PATH in the system variables list and add the following to the paths (Fig.5.9):
;C:\cygwin64\bin\;C:\cygwin64\lib (remember to use the correct location of your cygwin installation).
Remember to use semi-colon after every path (Fig.5.10).

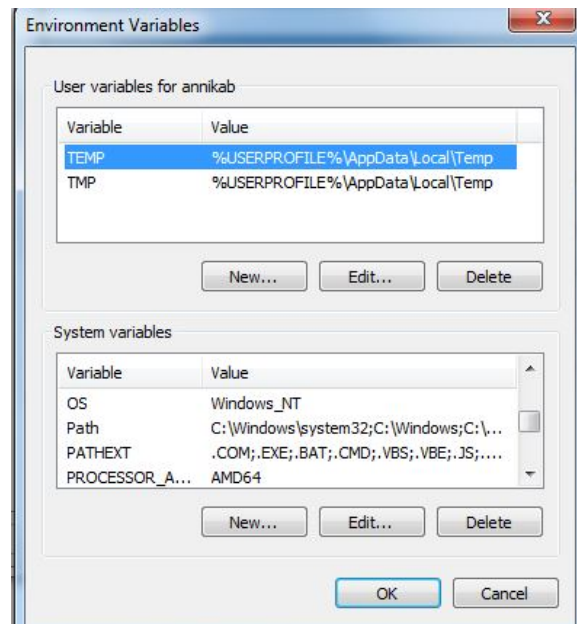


Figure 5.9: Setting environment variables in Windows 7: step3

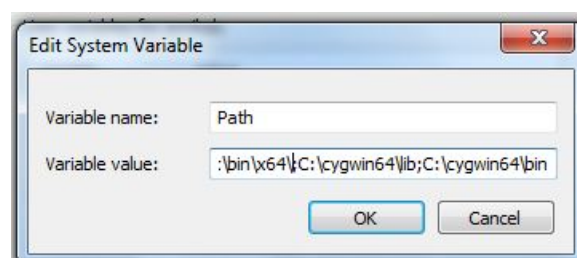


Figure 5.10: Setting environment variables in Windows 7: step4

Click OK.

- Go to section 5.3.2 and 5.3.3 and install OpenMPI and HYPRE.
- Now you are setup to compile DiveMESH and REEF3D.

Open the cygwin terminal. Navigate to the location of the REEF3D and DiveMESH folders. Access your windows system files by changing directory:
`cd/cygdrive/c/` (navigates to your C: drive on Windows)

- Run the Makefiles provided with the sourcecodes using:
`make -f Makefile.txt`
 for both REEF3D and DiveMESH.
 REEF3D and DiveMESH compiled.

IMPORTANT: There is a known issue with the 'mkdir' command on a windows platform.

Thus add P 14 0 to your ctrl.txt file in order to obtain output from reef3d.

5.5 Installing REEF3D on Windows 8

IMPORTANT: From our experience, the installation of REEF3D on Windows 8 following the procedure for Windows 7 works in most of the cases. But sometimes it can fail due to complexities with Windows security or administration settings. As such, we strongly recommend to run REEF3D under a UNIX/LINUX based system. If that is not possible, the second best option is to update to Windows 10, where an officially supported Linux bash is available. Also keep in mind, that the performance of REEF3D will be much better on a UNIX/LINUX system or on Windows 10 through the Linux bash than on Windows 8.

5.6 Installing REEF3D on Windows 10

Windows 10 offers the feature to run a Unix bash terminal, thus making the installation similar to that on a Unix machine on its anniversary update version. In order to be able to obtain the bash terminal on Windows 10, the following pre-requisites have to be met:

- Have administrator privileges on your 64-bit computer
- Install the anniversary update version of Windows 10 (64-bit)
Check the version of the Windows 10 by navigating to Settings> System> About.
You need to have version 1607. If you have an earlier version, check for updates and install the update giving you version 1607 or above. Restart the computer when prompted.

5.6.1 Getting Unix Terminal on Windows 10

- Turn on **Developer mode** by going to Settings> Update and Security> For Developers.
- From Start, search for 'Turn Windows features on or off' and select **Windows subsystem for Linux (Beta)** in the window and click OK.
- Restart the computer when prompted.
- Search of 'bash' 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.
- Your Unix home directory is located at C:\Users\<Windows user name>\AppData\Local\lxss. The folder lxss is a hidden system folder and you will need to enable 'show hidden folders' in the view folder options and also uncheck the box for 'hide system folders' to be able to see and access this folder.

5.6.2 Installing GCC compilers, OpenMPI and Hypre

- Open the bash terminal and update the apt-get package using:
sudo apt-get update

- To install GCC and fortran compilers, use:
sudo apt-get install gcc
sudo apt-get install g++
sudo apt-get install gfortran
- Use the following commands to find the path to the compilers and note down for later use:
which gcc
which g++
which gfortran
- Install Cmake using:
sudo apt-get install cmake
- Install libtools using:
sudo apt-get install libtool

Now the essential components are installed. Proceed with the installation of OpenMPI and Hydre as described in Sections 5.3.2 and 5.3.3.

Chapter 6

Post-Processing

6.1 Paraview

Paraview is an open-source visualization tool. It is cross-platform and works on Mac OS X, 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. The following procedure is used to visualize the results from a 2D simulation or the results along a plane from a 3D simulation.

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.7.15.

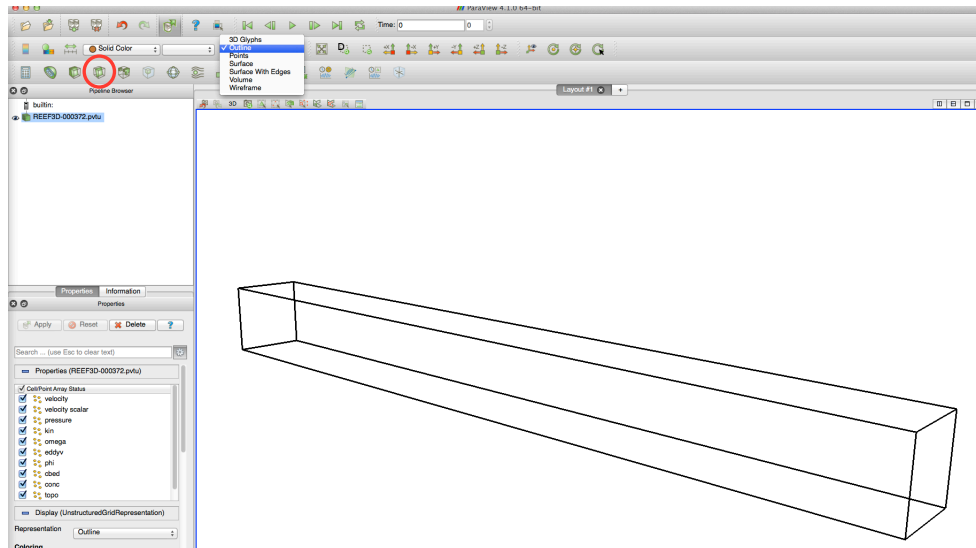


Figure 6.1: Outline of the numerical wave tank

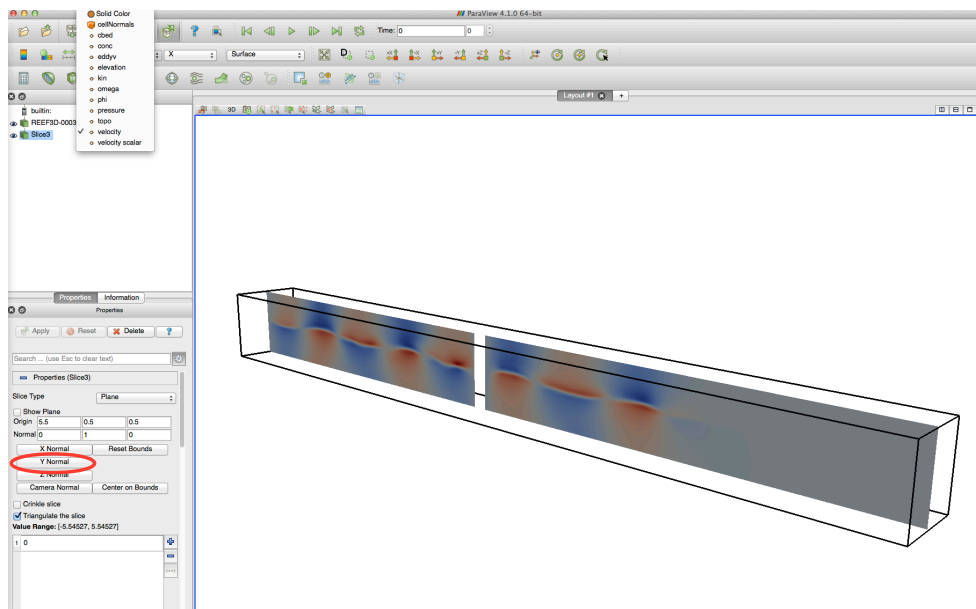


Figure 6.2: Creating a 2D slice

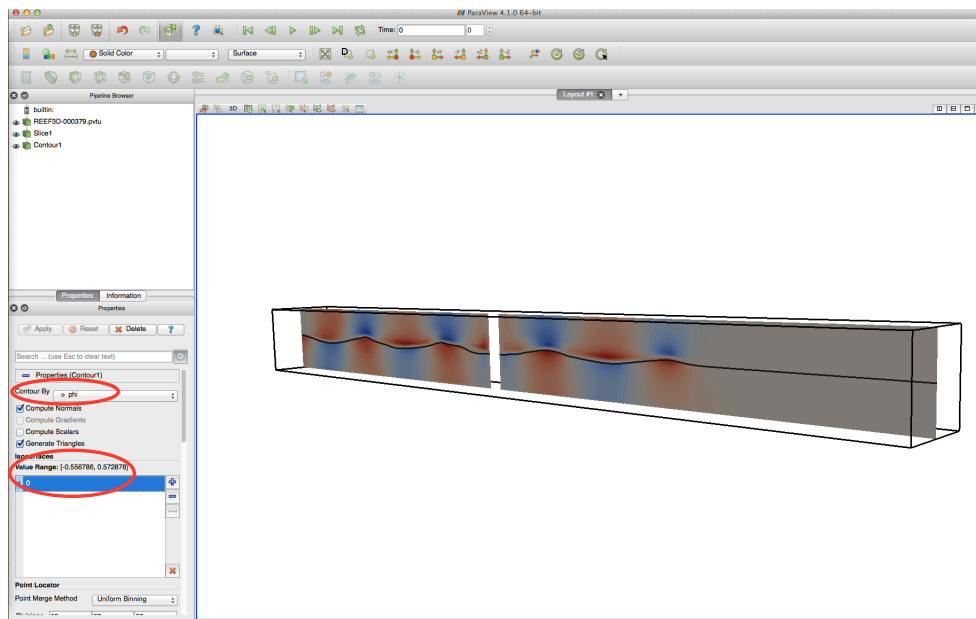


Figure 6.3: Generating the free surface

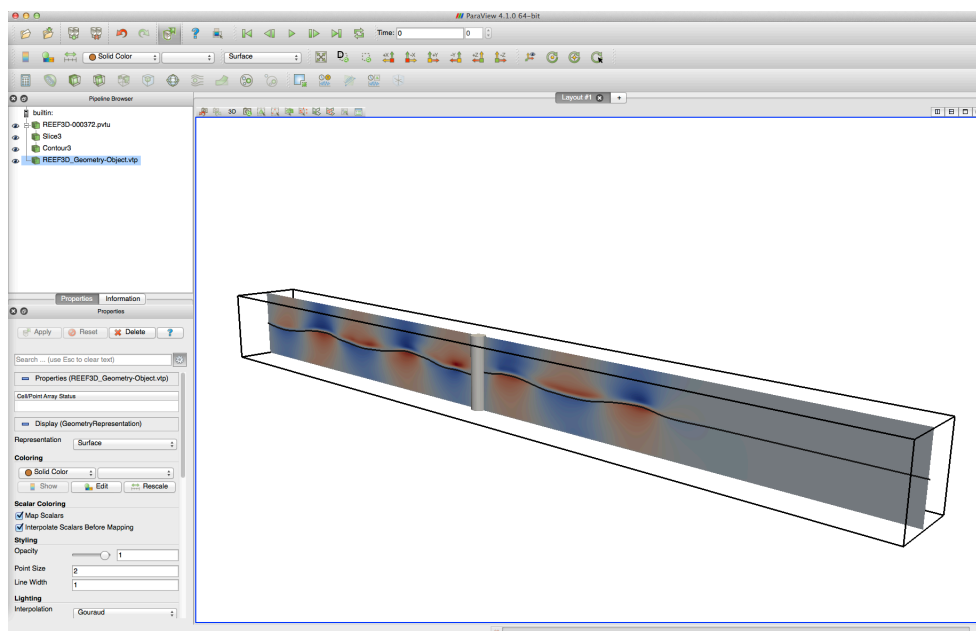


Figure 6.4: Including the geometry of the structure

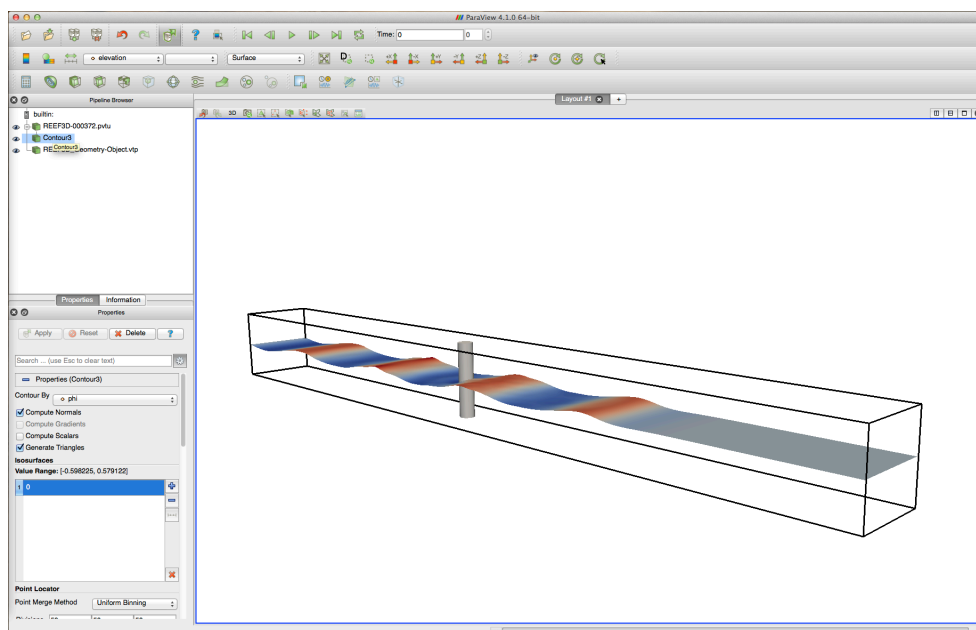


Figure 6.5: Visualizing the free surface in 3D

6.1.1 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.

Chapter 7

Tutorial

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 'control.txt' files describe the mesh and are read by DIVEMesh, a separate open-source mesh editor. 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.

7.1 2D Dam Break

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

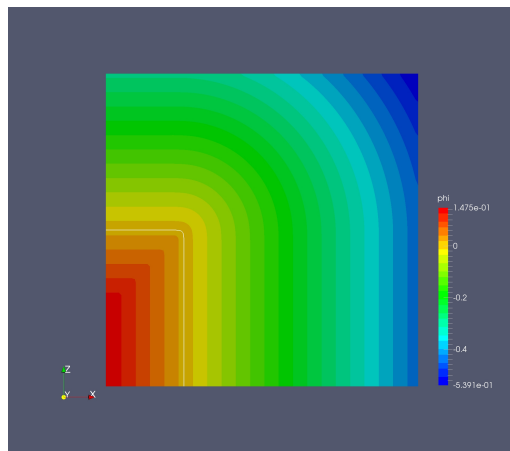
7.1.1 DIVEMesh: control.txt

```
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
```

7.1.2 REEF3D: ctrl.txt

```
B 5 1 // rough wall law
B 10 1 // turn on wall law for velocities
B 50 0.0001 // wall roughness  $k_s$ 
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 10 14 // HYPRE Struct BiCGStab pressure solver
N 11 11 // HYPRE Struct PFMG preconditioner
N 40 1 // 2nd-order Adams-Bashforth 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
```

7.1.3 Results



(a) Initial Condition with level set function

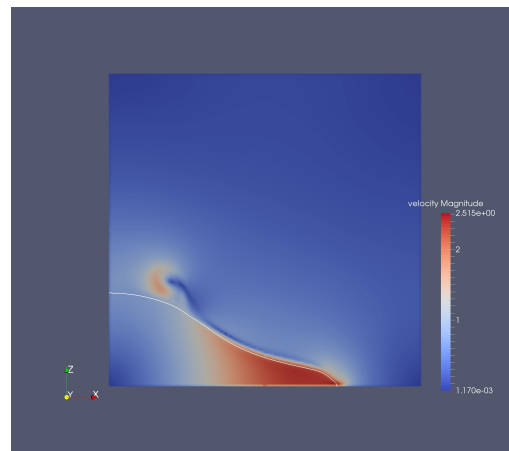
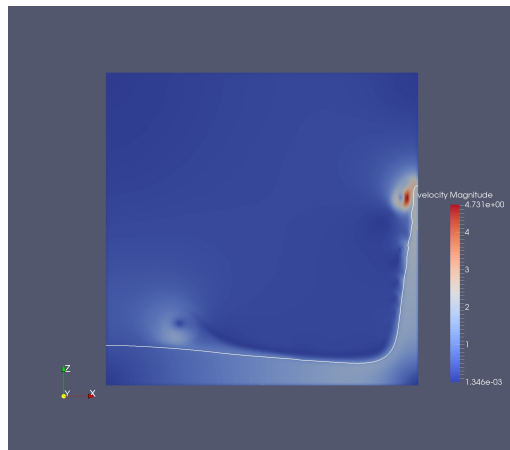
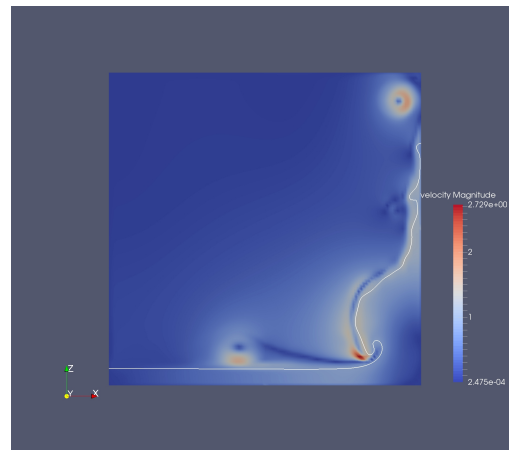
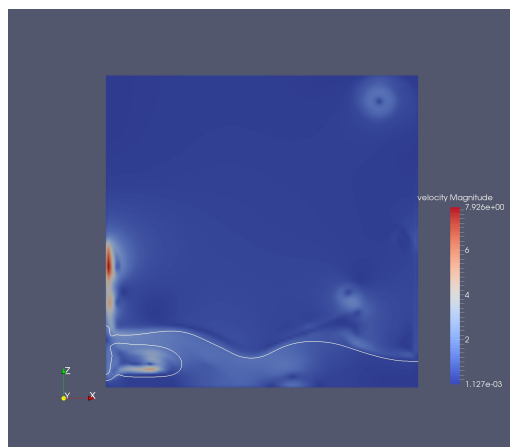
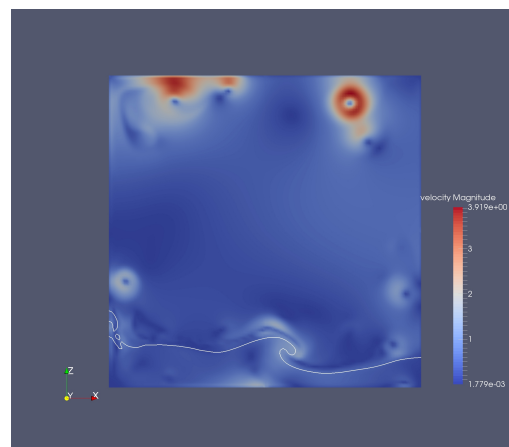
(b) after $t = 0.2$ sec with velocity magnitude(c) after $t = 0.4$ sec with velocity magnitude(d) after $t = 0.7$ sec with velocity magnitude(e) after $t = 1.0$ sec with velocity magnitude(f) after $t = 1.5$ sec with velocity magnitude

Figure 7.1: Dam Break

7.2 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.

7.2.1 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
```

7.2.2 REEF3D: ctrl.txt

```
B 5 1 // rough wall law
B 10 1 // turn on wall law for velocities
B 50 0.0001 // wall roughness  $k_s$ 
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 10 14 // HYPRE Struct BiCGStab pressure solver
N 11 11 // HYPRE Struct PFMG preconditioner
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 0.3 // 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
```

7.2.3 Results

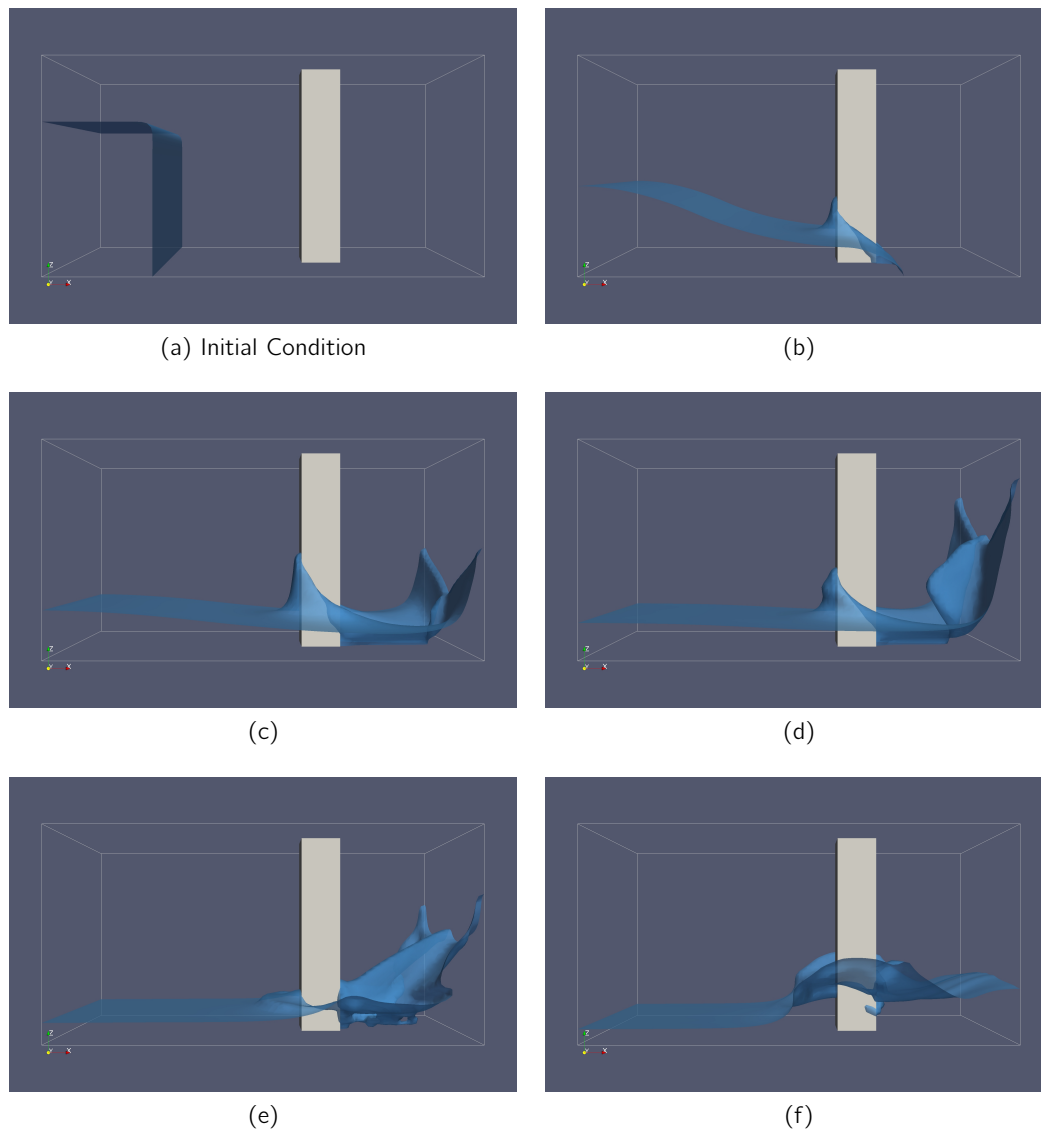


Figure 7.2: 3D Dam Break with Obstacle

7.3 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.

7.3.1 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 0.05 // mesh size dx
F 3 0.0 0.5 // Freecor: vertical extension of channel,  $z_s$  and  $z_e$ 
F 11 0.0 0.8 0.6 0.6 // Freecor:  $x_s$ ,  $x_e$ ,  $y_s$ ,  $y_e$  for first point of left side in flow direction
F 11 0.8 1.1 0.6 0.4
F 11 1.1 1.4 0.4 0.4
F 11 1.4 1.7 0.4 0.6
F 11 1.7 2.5 0.6 0.6
F 21 0.0 0.8 0.0 0.0 // Freecor:  $x_s$ ,  $x_e$ ,  $y_s$ ,  $y_e$  for first point of right side in flow direction
F 21 0.8 1.1 0.0 0.2
F 21 1.1 1.4 0.2 0.2
F 21 1.4 1.7 0.2 0.0
F 21 1.7 2.5 0.0 0.0
M 10 4
```

7.3.2 REEF3D: ctrl.txt

```
B 5 1 // use rough wall law
B 10 1 // use wall functions for the velocities
B 11 1 // use wall functions for the turbulence model
B 50 0.0001 // wall roughness  $k_s$ 
B 60 1 // turn on ioFlow
D 10 4 // Conservative WENO for velocities
D 30 11 // SIMPLE method for the pressure
F 30 12 // 2nd-order implicit time stepping 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
I 20 1500 // iterations for potential flow solver
N 10 14 // HYPRE Struct BiCGStab pressure solver
N 11 11 // HYPRE Struct PFMG preconditioner
```



```
N 40 12 // 2nd-order implicit time stepping 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
N 50 2 // number of outer iterations of the SIMPLE algorithm
N 53 0.3 // relaxation factor pressure
N 54 0.7 // relaxation factor for the velocities
N 55 0.7 // relaxation factor for the turbulence model
N 56 0.7 // relaxation factor for the level set function
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
```

7.3.3 Results

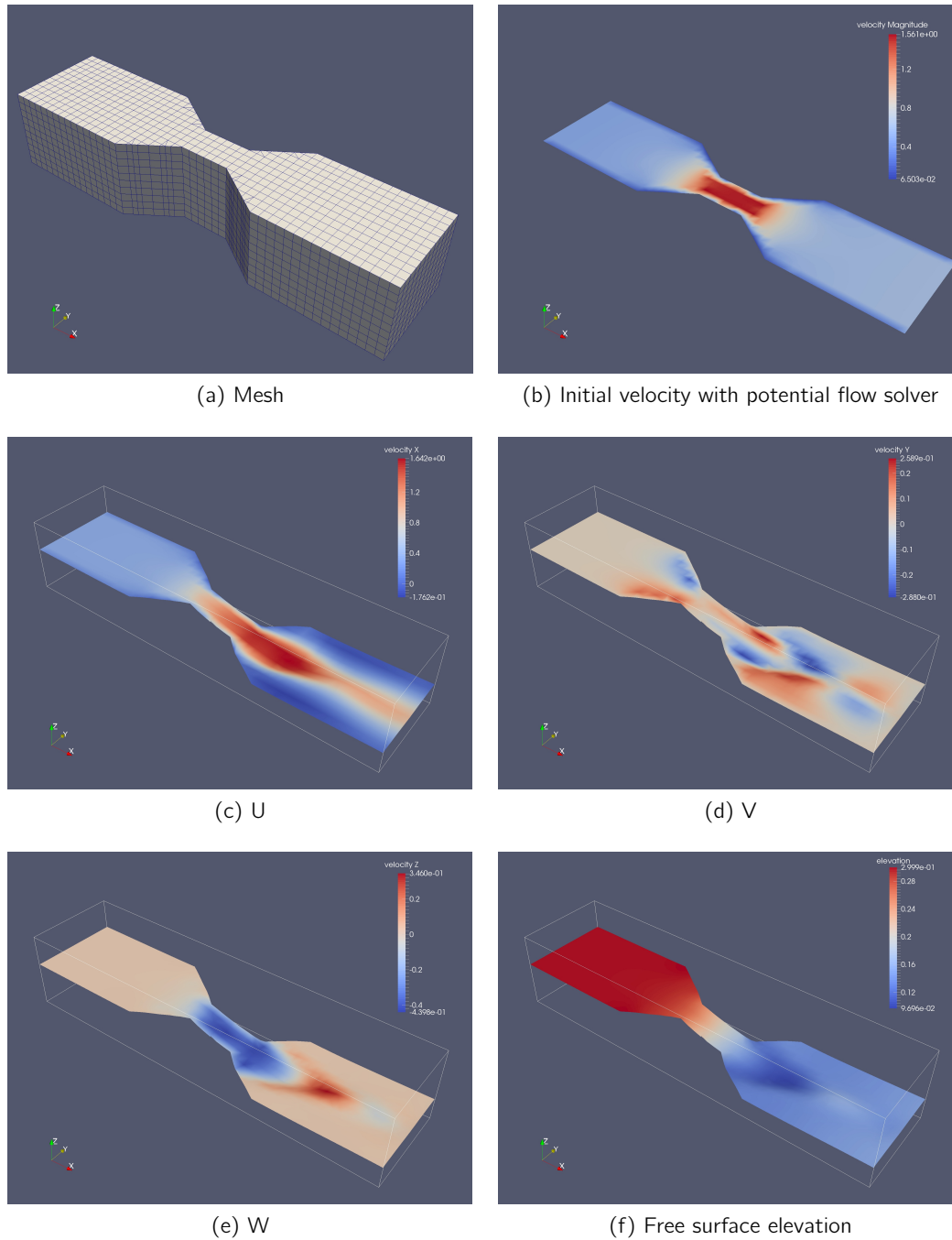


Figure 7.3: Narrow Contraction

7.4 Flow around a Circular Pier

7.4.1 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 1.5 0.0 1.0 0.0 0.4 // rectangular domain size
O 33 0.5 0.5 0.075 // vertical circular cylinder
M 10 4 // number of parallel processes
```

7.4.2 REEF3D: ctrl.txt

```
B 5 2 // use smooth wall law
B 10 1 // use wall functions on the velocities
B 11 1 // use wall functions for the turbulence model
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 11 // SIMPLE method for the pressure
F 30 12 // second-order implicit time treatment for the level set 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
I 20 1500 // iterations for potential flow solver
N 10 14 // HYPRE Struct BiCGStab pressure solver
N 11 11 // HYPRE Struct PFMG preconditioner
N 40 12 // 2nd-order implicit time treatment for the velocities
N 45 100000 // maximum number of iterations
N 47 0.3 // factor for CFL criterion
M 10 4 // number of parallel processes
N 50 2 // number of outer iterations of the SIMPLE algorithm
N 53 0.3 // relaxation factor for the pressure
N 54 0.7 // relaxation factor for the velocities
N 55 0.7 // relaxation factor for the turbulence model
N 56 0.7 // relaxation factor for the level set function
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
```

7.4.3 Results

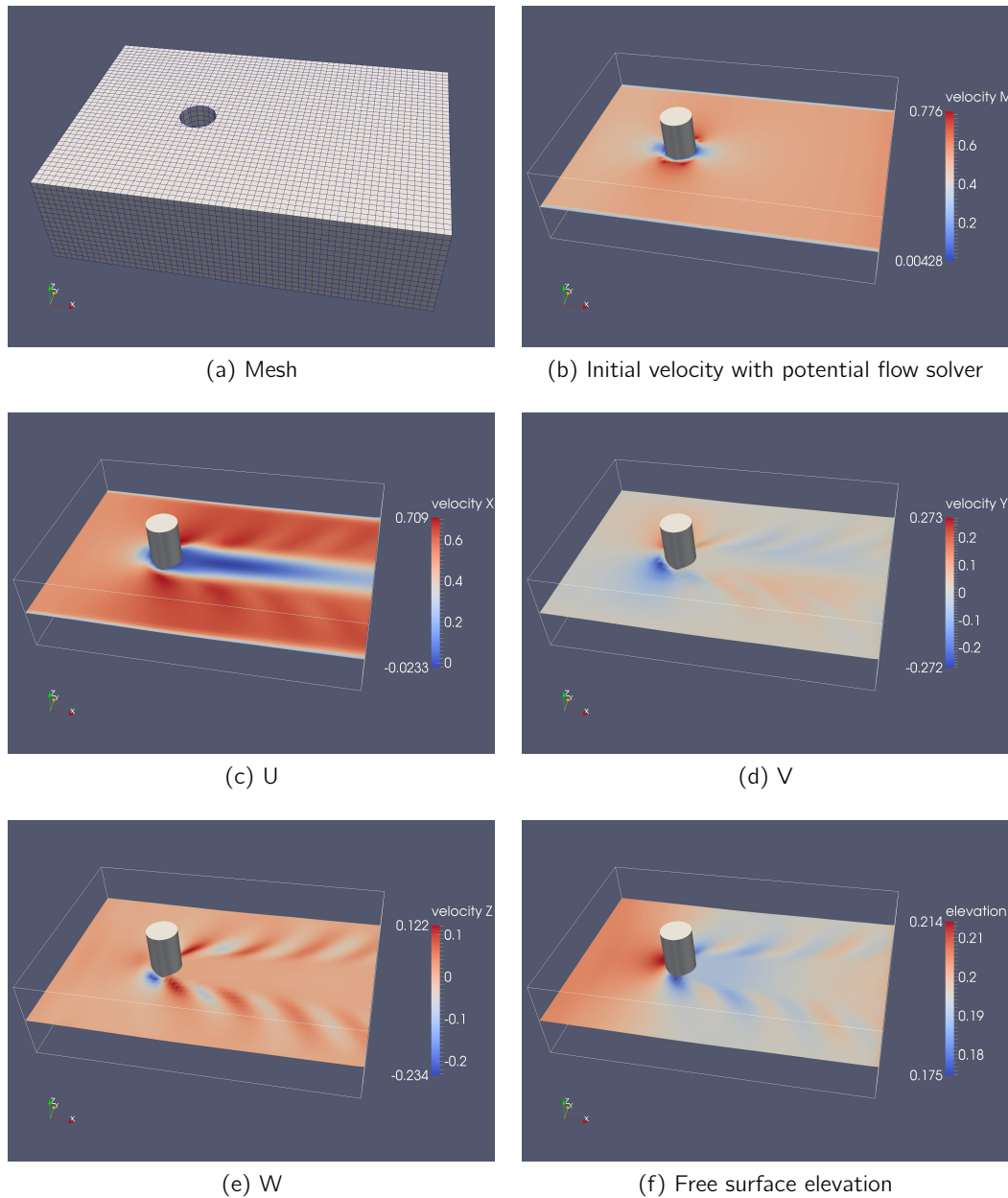


Figure 7.4: Flow around Circular Cylinder

7.5 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.

7.5.1 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
```

7.5.2 REEF3D: ctrl.txt

```
B 5 1 // use rough wall laws
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 0.01 4.0 0.0 // wave amplitude, wave length, phase shift
B 96 4.0 0.0 8.0 // relaxation length 1, relaxation length 2, 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 10 14 // HYPRE Struct BiCGStab pressure solver
N 11 11 // HYPRE Struct PFMG preconditioner
N 40 3 // 3rd-order Runge-Kutta Scheme for velocity time treatment
N 41 90.0 // Maximum simulation time
N 47 0.1 // 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 wsline to file
P 54 500 // print out interval based on iterations for wslines
```

7.5.3 Results

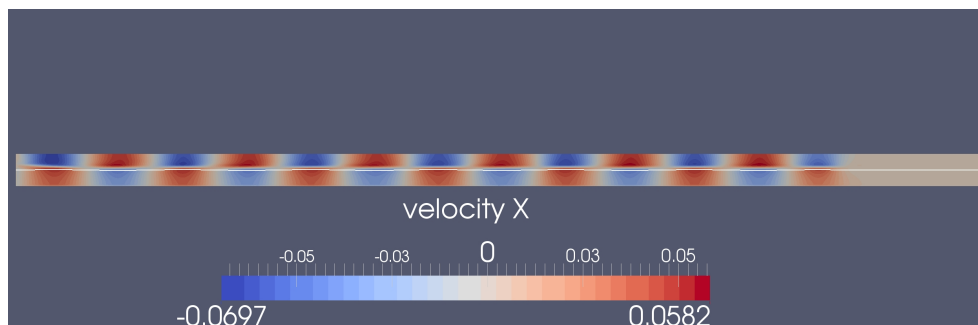


Figure 7.5: Wave tank with free surface and U

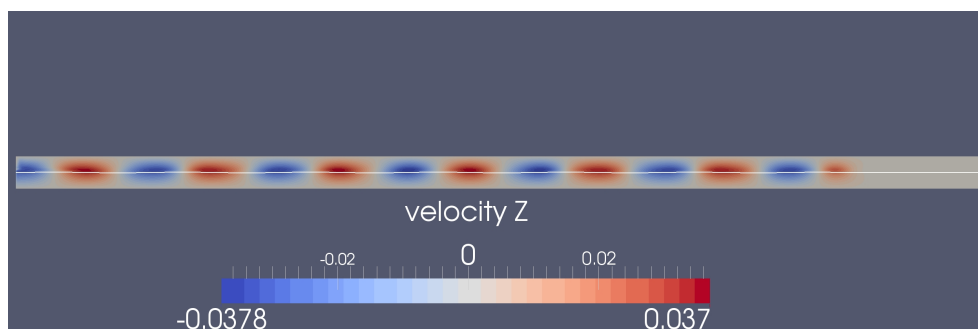


Figure 7.6: Wave tank with free surface and W

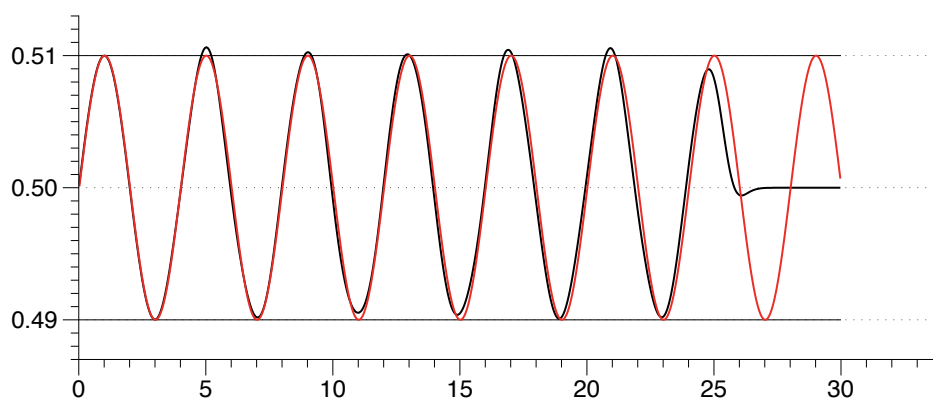


Figure 7.7: Free surface along the tank, numerical result (black line) vs theory (red line)

7.6 Wave Propagation over a 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 [2]. The simulations are run in 2D with wave gages at several locations along the wave tank.

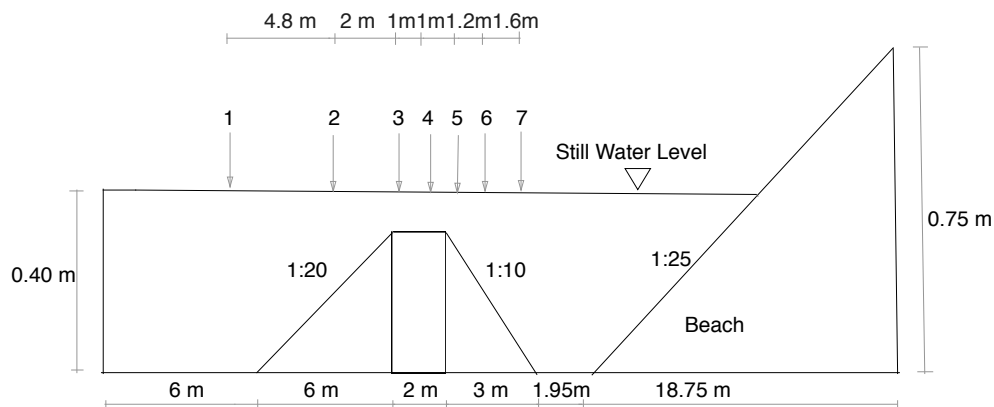


Figure 7.8: Experimental setup submerged bar

7.6.1 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 // 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 4 // number of parallel processes

```

7.6.2 REEF3D: ctrl.txt

```

B 5 1 // use rough wall laws
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 0.01 3.73 0.0 // wave amplitude, wave length, phase shift
B 96 3.73 0.0 3.73 // relaxation length 1, relaxation length 2, 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 10 14 // HYPRE Struct BiCGStab pressure solver
N 11 11 // HYPRE Struct PFMG preconditioner
N 40 3 // 3rd-order Runge-Kutta scheme for velocity time treatment
N 41 60.0 // Maximum simulation time
N 47 0.1 // 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 wsflne in x-direction
P 53 1 // add theoretical wsflne 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

```

7.6.3 Result

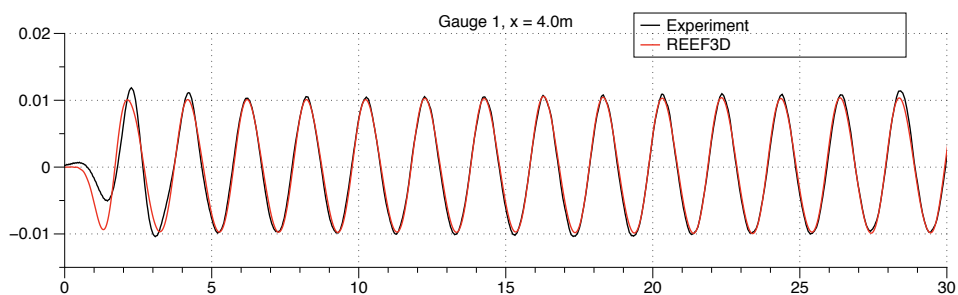


Figure 7.9: Wave gage 1

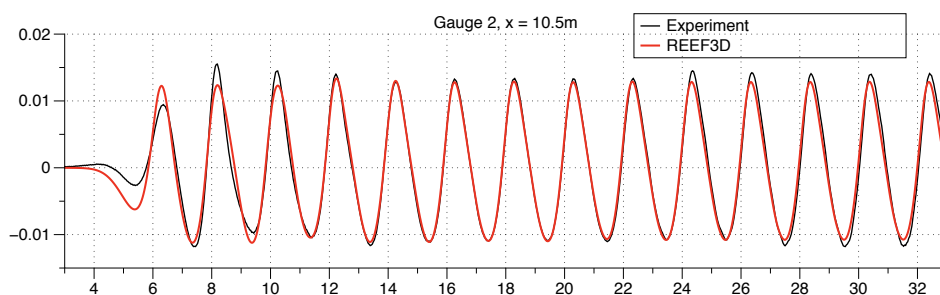


Figure 7.10: Wave gage 2

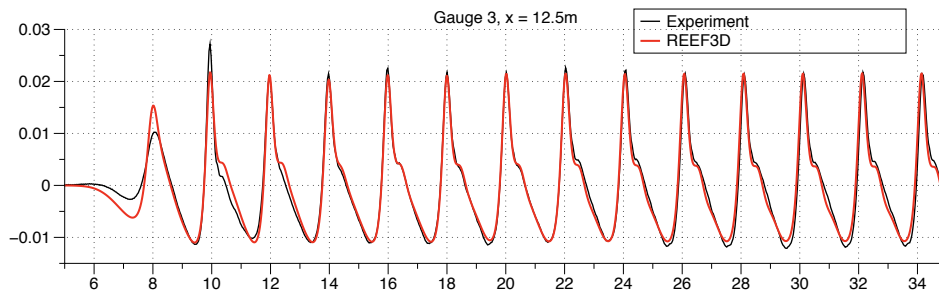


Figure 7.11: Wave gage 3

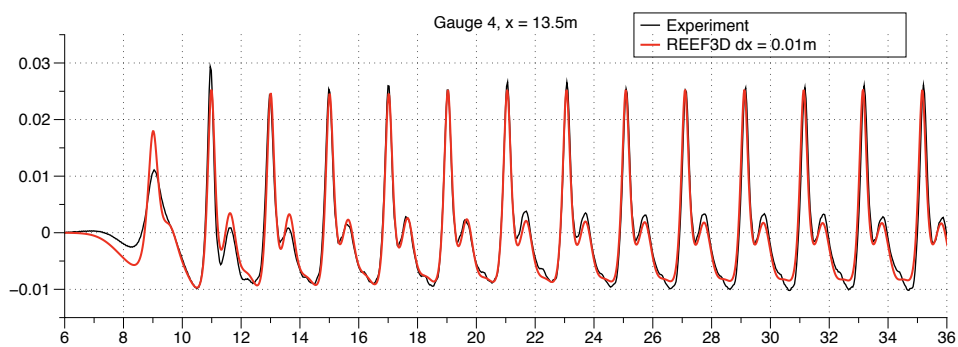


Figure 7.12: Wave gage 4

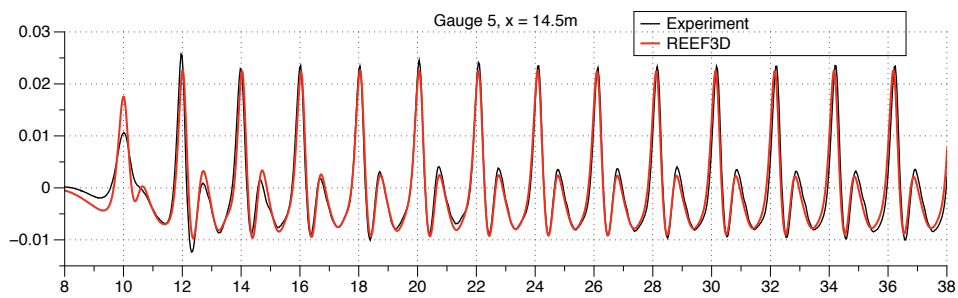


Figure 7.13: Wave gage 5

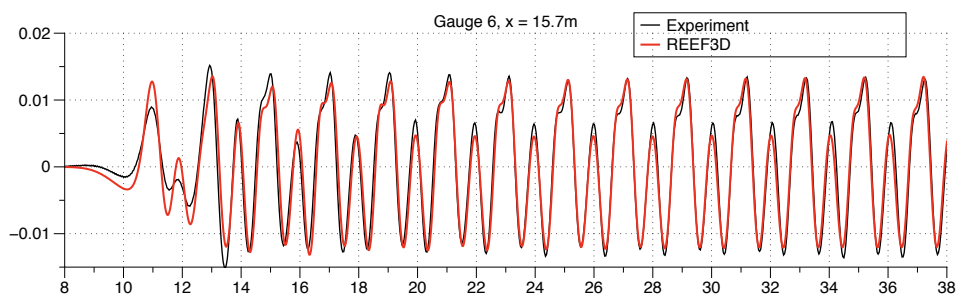


Figure 7.14: Wave gage 6

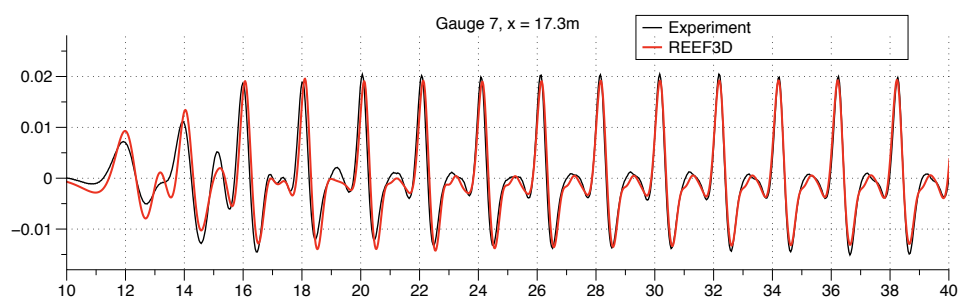


Figure 7.15: Wave gage 7

Chapter 8

Publications with REEF3D

8.1 Journal Articles

15. Bihs H., Kamath A. (2016), A Combined Level Set/Ghost Cell Immersed Boundary Representation for Simulations of Floating Bodies, accepted to *International Journal for Numerical Methods in Fluids*.
14. Kamath A., Bihs H., Arntsen Ø.A. (2016), Study of Water Impact and Entry of Free Falling Wedge Using CFD Simulations, accepted to *Journal of Offshore Mechanics and Arctic Engineering*.
13. Bihs H., Kamath A., Alagan Chella M., Aggarwal A., Arntsen Ø.A. (2016), A New Level Set Numerical Wave Tank with Improved Density Interpolation for Complex Wave Hydrodynamics, *Computers & Fluids*, Vol. 140, DOI: 10.1016/j.compfluid.2016.09.012.
12. Bihs H., Kamath A., Alagan Chella M., Arntsen Ø.A. (2016), Breaking wave interaction with tandem cylinders under different impact scenarios, *Journal of Waterway, Port, Coastal, and Ocean Engineering*, Vol. 142 (5), DOI: 10.1061/(ASCE)WW.1943-5460.0000343.
11. Alagan Chella M., Bihs H., Muskulus M. (2016), Solitary Wave Breaking and Breaking Wave Forces on a Vertically Mounted Circular Cylinder over a Sloping Seabed, *Journal of Ocean Engineering and Marine Energy*, DOI: 10.1007/s40722-016-0055-5.
10. Fleit G., Baranya S., R  ther N., Bihs H., Kramer T., Jozsa J (2016), Investigation of the effects of ship induced waves on the littoral zone with field measurements and CFD modeling, *Water*, Vol. 8 (7), DOI: 10.3390/w8070300.
9. Kamath A., Bihs H., Alagan Chella M., Arntsen Ø.A. (2016), Upstream and Downstream Cylinder Influence on the Hydrodynamics of a Four Cylinder Group, Vol. 142 (4), *Journal of Waterway, Port, Coastal, and Ocean Engineering*, DOI: 10.1061/(ASCE)WW.1943-5460.0000339.
8. Alagan Chella M., Bihs H., Myrhaug D., Muskulus M. (2016), Hydrodynamic Characteristics of Spilling and Plunging Breakers over a Sloping Seabed, *Ocean Modelling*, Vol. 103, pp. 53-72, DOI: 10.1016/j.ocemod.2015.11.011.

7. Kamath A., Alagan Chella M., Bihs H., Arntsen Ø.A. (2015), CFD Investigations of Wave Interaction with a Pair of Large Tandem Cylinders, Vol. 108, pp. 738-748, *Ocean Engineering*, DOI: 10.1016/j.oceaneng.2015.08.049.
6. Kamath A., Bihs H., Arntsen Ø.A. (2015), Numerical Investigations of the Hydrodynamics of an Oscillating Water Column, *Ocean Engineering*, Vol. 102, pp. 40-50, DOI: 10.1016/j.oceaneng.2015.04.043.
5. Alagan Chella M., Bihs H., Myrhaug D. (2015), Characteristics and Profile Asymmetry Properties of Waves Breaking Over an Impermeable Submerged Reef, *Coastal Engineering*, Vol. 100, pp. 26-36, DOI: 10.1016/j.coastaleng.2015.03.008
4. Afzal M.S., Bihs H., Kamath A., Arntsen Ø.A. (2015), Three Dimensional Numerical Modeling of Pier Scour under Current and Waves using the Level Set Method, *Journal of Offshore Mechanics and Arctic Engineering*, Vol. 137(3), pp. 032001-032007, DOI: 10.1115/1.4029999.
3. Kamath A., Bihs H., Arntsen Ø.A. (2015) Numerical Modeling of Power Take-off Damping in an Oscillating Water Column Device, *International Journal of Marine Energy*, Vol. 10, pp. 1-16, DOI: 10.1016/j.ijome.2015.01.001
2. Alagan Chella M., Bihs H., Myrhaug D., Muskulus M. (2015), Breaking characteristics and geometric properties of spilling breakers over slopes, *Coastal Engineering*, Vol. 95, pp. 4-19, DOI: 10.1016/j.coastaleng.2014.09.003.
1. Kamath A., Alagan Chella M., Bihs H., Arntsen Ø.A. (2015), CFD Investigations into Wave Forces and Flow around Three Cylinders placed in different configurations, *Engineering Applications of Computational Fluid Mechanics*, Vol. 9, No. 1, pp. 343-354, DOI: 10.1080/19942060.2015.1031318.

8.2 Conference Papers

41. Ahmad N., Bihs H., Kamath A., Arntsen Ø.A. (2016), 3D Numerical Modelling of Pile Scour with Free Surface profile under Wave and Current using Level Set Method in Model REEF3D, *8th International Conference on Local Scour and Erosion, ISCE 2016*, Oxford, UK.
40. Bihs H., Alagan Chella M., Kamath A., Arntsen Ø.A. (2016), Wave-Structure Interaction of Focused Waves with REEF3D, *35th International Conference on Ocean, Offshore and Arctic Engineering, OMAE 2016*, Busan, South-Korea.
39. Grotle E.L., Bihs H., Pedersen E., Æsøy V. (2016), CFD Simulations of Non-Linear Sloshing in a Rotating Rectangular Tank Using the Level Set Method, *35th International Conference on Ocean, Offshore and Arctic Engineering, OMAE 2016*, Busan, South-Korea.
38. Kamath A., Bihs H., Arntsen Ø.A. (2016), Study of Water Impact and Entry of Free Falling Wedge Using CFD Simulations, *35th International Conference on Ocean, Offshore and Arctic Engineering, OMAE 2016*, Busan, South-Korea.

37. Alagan Chella M., Bihs H., Kamath A., Myrhaug D., Arntsen Ø.A. (2016), Numerical Modelling of Breaking Wave Interaction with a Group of Four Vertical Slender Cylinders in Square Arrangements, *35th International Conference on Ocean, Offshore and Arctic Engineering, OMAE 2016*, Busan, South-Korea.
36. Aggarwal A., Alagan Chella M., Kamath A., Bihs H., Arntsen Ø.A. (2016), Numerical Simulation of Irregular Wave Forces on a Horizontal Cylinder, *35th International Conference on Ocean, Offshore and Arctic Engineering, OMAE 2016*, Busan, South-Korea.
35. Alagan Chella M., Ros Collados, X., Bihs H., Myrhaug D., Arntsen Ø.A. (2016), Numerical and experimental investigation of breaking wave interaction with a vertical slender cylinder, *13th Deep Sea Offshore Wind RD Conference, EERA DeepWind'2016*, Trondheim, Norway.
34. Aggarwal A., Alagan Chella M., Kamath A., Bihs H., Arntsen Ø.A. (2016), Irregular Wave Forces on a Large Vertical Circular Cylinder, *13th Deep Sea Offshore Wind RD Conference, EERA DeepWind'2016*, Trondheim, Norway.
33. Kamath A., Bihs H., Alagan Chella M., Arntsen Ø.A. (2015), CFD Simulations to Determine Wave Forces on a Row of Cylinders, APAC 2015 Conference, Chennai, India, *Procedia Engineering*, DOI: 10.1016/j.proeng.2015.08.334.
32. Ahmad N., Bihs H., Kamath A., Arntsen Ø.A. (2015), Three-Dimensional CFD Modeling of Wave Scour around Side-by-Side and Triangular Arrangement of Piles with REEF3D, *Procedia Engineering*, APAC 2015 Conference, Chennai, India, DOI: 10.1016/j.proeng.2015.08.355.
31. Kamath A., Bihs H., Arntsen Ø.A. (2015), CFD Simulations for Fluid Structure Interaction due to Breaking Waves, *8th National Conference on Computational Mechanics, MekIT' 15*, Trondheim, Norway.
30. Ahmad N., Bihs H., Kamath A., Arntsen Ø.A. (2015), Numerical Simulation of Waves and Local Scour Around Vertical Cylinders in a Triangular Arrangement using REEF3D, *8th National Conference on Computational Mechanics, MekIT' 15*, Trondheim, Norway.
29. Aggarwal A., Kamath A., Bihs H., Arntsen Ø.A. (2015), Validation of Irregular Wave Generation in REEF3D, *8th National Conference on Computational Mechanics, MekIT' 15*, Trondheim, Norway.
28. Bihs H., Kamath A., Alagan Chella M., Arntsen Ø.A. (2015) CFD Simulations of Roll Motion of a Floating Ice Block in Waves Using REEF3D, *POAC 2015, International Conference on Port and Ocean Engineering under Arctic Conditions*, Trondheim, Norway.
27. Ahmad N., Bihs H., Kamath A., Arntsen Ø.A. (2015) CFD Modeling of Local Scour around a Pair of Tandem Cylinders under Wave Conditions, *POAC 2015, International Conference on Port and Ocean Engineering under Arctic Conditions*, Trondheim, Norway.
26. Ahmad N., Bihs H., Afzal M.S., Arntsen Ø.A. (2015) Three-Dimensional Numerical Modeling of Local Scour around a Non-Slender Cylinder under Varying Wave Conditions, *36th IAHR World Congress*, Delft, The Netherlands.
25. Alagan Chella M., Bihs H., Myrhaug D. (2015) An Investigation on Plunging Breaking Waves over a Slope with a CFD based Numerical Wave Tank, *36th IAHR World Congress*, Delft, The Netherlands.

24. Kamath A., Bihs H., Arntsen Ø.A. (2015) Three Dimensional CFD Modeling of flow around an OWC wave energy converter, *36th IAHR World Congress*, Delft, The Netherlands.
23. Kamath A., Bihs H., Alagan Chella M., Arntsen Ø.A. (2015), CFD Simulations of Wave Propagation and Shoaling over a Submerged Bar, *Aquatic Procedia*, DOI: 10.1016/j.aqpro.2015.02.042.
22. Kamath A., Bihs H., Arntsen Ø.A. (2014), A CFD based 3D Numerical Wave Tank to Investigate Wave Interaction with Rectangular Cylinders, *ICHE 2014*, Hamburg, Germany.
21. Chella A., Bihs H., Muskulus M. (2014), Numerical Simulation of Breaking Waves on a Plane Slope with a Parallel Level Set Solver, *ICHE 2014*, Hamburg, Germany.
20. Afzal M.S., Bihs H., Arntsen Ø.A. (2014), 3D Numerical Modeling of Contraction Scour under Steady Current using the Level Set Method, *ICHE 2014*, Hamburg, Germany.
19. Kamath A., Bihs H., Arntsen Ø.A. (2014), Comparison of 2D and 3D Simulations of an OWC Device in a Numerical Wave Tank, *ICCE 2014*, Seoul, Korea.
18. Bihs H., Kamath A., Arntsen Ø.A. (2014), CFD Multiphase Simulation of Two-Fluid Sloshing with Free Surface Motion using the Level Set Method, *10th International Conference on CFD in Oil & Gas, Metallurgical and Process Industries*, Trondheim, Norway.
17. Kamath A., Bihs H., Olsen J.E., Arntsen Ø.A. (2014), Effect of Compressibility in CFD Simulation of an Oscillating Water Column Device, *10th International Conference on CFD in Oil & Gas, Metallurgical and Process Industries*, Trondheim, Norway.
16. Kamath A., Bihs H., Arntsen Ø.A. (2014), Analysis of Wave Interaction with Cylinders using a 3D Numerical Wave Tank, *OMAE 2014*, San Francisco, USA.
15. Afzal M.S., Bihs H., Kamath A., Arntsen Ø.A. (2014), Three Dimensional Numerical Modeling of Pier Scour under Current and Waves using the Level Set Method, *OMAE 2014*, San Francisco, USA.
14. Bihs H., Afzal M.S., Kamath A., Arntsen Ø.A. (2013), REEF3D : An Advanced Wave Energy Design Tool for the Simulation of Wave Hydrodynamics and Sediment Transport, *International Workshop on Ocean Wave Energy*, Chennai, India.
13. Kamath A., Bihs H., Arntsen Ø.A. (2013), Investigating OWC Wave Energy Converters Using Two-Dimensional CFD Simulations, *International Workshop on Ocean Wave Energy*, Chennai, India.
12. Bihs H., Kamath A., Arntsen Ø.A. (2013), A 3D Numerical Wave Tank using the Level Set Method for the Calculation of Wave Propagation and Runup, *ISOPE 2013*, Anchorage, USA.
11. Kamath A., Bihs H., Arntsen Ø.A. (2013), Evaluation of Hydrodynamic Efficiency of an Oscillating Water Column Device through CFD Simulation, *ISOPE 2013*, Anchorage, USA.
10. Bihs H., Ong M.C. (2013), Numerical Simulation of Flows past Partially-Submerged Horizontal Circular Cylinders in Free Surface Waves, *OMAE 2013*, Nantes, France.

9. Chella M.A., Bihs H., Kamath A., Muskulus M. (2013), Numerical Modeling of Breaking Waves over a Reef with a Level-Set based Numerical Wave Tank, OMAE 2013, Nantes, France.
8. Kamath A., Bihs H., Arntsen Ø.A. (2013), Calculation of Wave Forces on Cylindrical Piles using a 3D Numerical Wave Tank, OMAE 2013, Nantes, France.
7. Bihs H., Ong M.C., Kamath A., Arntsen Ø.A. (2013), A Level Set Method based Numerical Wave Tank for the Calculation of Wave Forces on Horizontal and Vertical Cylinders, *MekIT 13*, Trondheim, Norway.
6. Kamath A., Bihs H., Arntsen Ø.A. (2013), Application of Porous Media Flow Relation to Simulate Pressure Drop across a Nozzle in a Two Dimensional Numerical Wave Tank, *MekIT 13*, Trondheim, Norway.
5. Bihs H. (2011), A Three Dimensional Multiphase Solver for the Calculation of Propagating Gravity Current Fronts, *7th International Symposium on Stratified Flows 2011*, Rome, Italy.
4. Bihs H. (2011), Three Dimensional Numerical Simulation of Bubble and Droplet Dynamics with a Parallel Particle Level Set Solver, *8th International Conference on CFD in Oil & Gas, Metallurgical and Process Industries*, Trondheim, Norway.
3. Bihs H. (2010), Numerical Investigations of Free Surface Flow in a Channel with a Long Contraction, *International Conference on Fluvial Hydraulics, River Flow 2010*, Braunschweig, Germany.
2. Bihs H. (2009), Three Dimensional Numerical Modeling of Free Surface Flow around an Obstacle with the Level Set Method, *33rd IAHR Congress 2009*, Vancouver, Canada.
1. Bihs H., Olsen N.R.B. (2009), Three Dimensional Numerical Modeling of Complex Free Surface Flow over a Backward Facing Step, *5th National Conference on Computational Mechanics, MekIT' 09*, Trondheim, Norway.

8.3 Reports

14. Paquereau-Gaboreau, T. (2016) Numerical simulation of three-dimensional stratified flows on smooth and rough bottom using an innovative CFD software: REEF3D, *Internship Report, Marine Civil Engineering, NTNU Trondheim*.
13. Schröder, T. (2016) Berechnung der Hydraulik einer stehenden Welle unter Zuhilfenahme von CFD-Modellen, *Master Thesis, Franzius-Institut für Wasserbau, Ästuar- und Küsteningenieurwesen, Leibniz Universität Hannover*.
12. Sasikumar, A. (2016) Numerical Modelling of Wave Hydrodynamics and Wave Interaction with Porous Structures with REEF3D, *Master Thesis, Marine Civil Engineering, NTNU Trondheim*.
11. Amarachaharam, T. (2016) Numerical Modelling of Focussed Wave Hydrodynamics and Focussed Wave-Structure Interaction with REEF3D, *Master Thesis, Marine Civil Engineering, NTNU Trondheim*.

10. Alagan Chella M. (2016) Breaking Wave Characteristics and Breaking Wave Forces on Slender Cylinders, *Doctoral Theses at NTNU Trondheim*, No. 78.
9. Kamath A.M. (2015) Kamath A. (2015) CFD based Investigation of Wave-Structure Interaction and Hydrodynamics of an Oscillating Water Column Device, *Doctoral Theses at NTNU Trondheim*, No. 326.
8. Gabor Fleit (2015), Comprehensive Analysis of Ship Induced Waves by Means of Field and Computational Methods, *Project thesis, Department of Hydraulic and Water Resources Engineering, Budapest University of Technology and Economics*.
7. Ankit Aggarwal (2015), Numerical Modelling of Irregular waves and Wave Forces with REEF3D, *Master thesis, NTNU Trondheim*.
6. Gabor Fleit (2015), Numerical Modeling of Open Channel Flow with REEF3D, *Bachelor thesis, Hydraulic Engineering, NTNU Trondheim*.
5. Aggarwal A. (2014) Numerical Modelling of Wave propagation and Wave Forces using REEF3D, *Project Thesis, Marine Civil Engineering, NTNU Trondheim*.
4. Raducan S. (2014) REEF3D Feasibility Testing, *Internship report, HR Wallingford*.
3. Afzal M.S. (2013) 3D Numerical Modeling of Sediment Transport under Current and Waves, *Master Thesis, Marine Civil Engineering, NTNU Trondheim*.
2. Kamath A.M. (2012) Wave Forces on Structures Using REEF3D, *Master Thesis, Marine Civil Engineering, NTNU Trondheim*.
1. Bihs H. (2011), Three-Dimensional Numerical Modeling of Local Scouring in Open Channel Flow, *Doctoral Theses at NTNU Trondheim*, No. 127.

Bibliography

- [1] S. F. Ashby and R. D. Flagout. A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. *Nuclear Science and Engineering*, 124(1):145–159, 1996.
- [2] S. Beji and J. A. Battjes. Experimental investigation of wave propagation over a bar. *Coastal Engineering*, 19:151–162, 1993.
- [3] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory. *hyper high performance preconditioners - User's Manual*, 2015.
- [4] A. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22:745–762, 1968.
- [5] 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.
- [6] V.E. Henson and U. Meier Yang. BoomerAMG: A parallel algebraic multigrid solver and preconditioner. *Applied Numerical Mathematics*, 41:155–177, 2002.
- [7] G. S. Jiang and D. Peng. Weighted ENO schemes for Hamilton Jacobi equations. *SIAM Journal of Scientific Computing*, 21:2126–2143, 2000.
- [8] G. S. Jiang and C. W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202–228, 1996.
- [9] B. E. Launder and D. B. Spalding. The numerical computation of turbulent flows. *Comput. Methods Appl. Mech. Eng.*, 3:269–289, 1974.
- [10] H. Le and P. Moin. An improvement of fractional step methods for the incompressible navier-stokes equations. *Journal of Computational Physics*, 92:369–379, 1991.
- [11] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.
- [12] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *Journal of Computational Physics*, 155:410–438, 1999.
- [13] M. Peric. Why STAR-CCM+ is the next step for CFD. *dynamics*, 23:3–4, 2004.

-
- [14] H. Schlichting. *Boundary layer theory*. McGraw-Hill Book Company, 1979.
 - [15] C. W. Shu. *Essentially Non-Oscillatory and Weighted Essentially Non-Oscillatory Schemes for Hyperbolic Conservation Laws*. ICASE Report, 1997.
 - [16] C. W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock capturing schemes. *Journal of Computational Physics*, 77:439–471, 1988.
 - [17] D. C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries Inc., La Canada, California., 1994.