MIKE 21 Wave Model FM

Hydrodynamic module

Scientific Documentation

DHI A/S headquarters
Agern Allé 5
DK-2970 Hørsholm
Denmark

+45 4516 9200 Telephone
mike@dhigroup.com
www.mikepoweredbydhi.com

Company Registration No.: DK36466871

CONTENTS

MIKE 21 Wave Model FM
Hydrodynamic Module
Scientific Documentation

1 Introduction

This document presents the scientific background for the MIKE 21 Wave Model FM. The objective is to provide the user with a detailed description of the governing equations, numerical discretization and solution methods.

MIKE 21 Wave Model FM can be applied in the following areas:

- Ports and terminals
  - Wave agitation caused by short and long waves
  - Input to dynamic ship mooring analysis (MIKE 21 MA)
- Coastal areas
  - Non-linear wave transformation
  - Surf and swash zone hydrodynamics
  - Wave breaking and run-up
  - Coastal flooding
- Coastal structures
  - Wave overtopping
  - Wave transmission (and reflection) through porous structures
  - Input to wave load calculation

The model is based on the numerical solution of the two-dimensional Boussinesq-type equations of Madsen and Sørensen (1992). The equations provide excellent accuracy in shoaling as well as in linear dispersion for the ratio of the water depth to the deep water wave length, h/L_0, as large as 0.5. The deep water wave length $L_0=gT^2/(2\pi)$, where T is the wave period and g is the gravity. The spatial discretization of the governing equations in conserved form is performed using a cell-centered finite volume method. The time integration is performed using an explicit Runge-Kutta scheme. The interface convective fluxes are calculated using an approximate Riemann solver. This shock-capturing scheme enables robust and stable simulation of flows involving shocks or discontinuities such as bores and hydraulic jumps. This is essential for modelling of waves in the breaking zone or porous structures.

2 Governing equations

The governing equations are solved in a Cartesian coordinate system.

2.1 Governing equations in a Cartesian coordinate system

2.1.1 Enhanced Boussinesq equations

The two-dimensional enhanced Boussinesq equations are obtained by integrating the Navier–Stokes equations over the water depth and taking into account the effect that vertical accelerations have on the pressure distribution. For more details on the derivation of the two-dimensional enhanced Boussinesq equations see Madsen et al. (1991) and Madsen and Sørensen (1992). The two-dimensional enhanced Boussinesq equations in conservative form can be expressed as

$$\frac{\partial h}{\partial t}+\frac{\partial hu}{\partial x}+\frac{\partial hv}{\partial y}=0 \tag{2.1}$$

$$\frac{\partial hu}{\partial t}+\frac{\partial hu^2}{\partial x}+\frac{\partial huv}{\partial y}= -gh\frac{\partial \eta}{\partial x}-\frac{\tau_{fx}}{\rho}-F_u-F_{vx}+\frac{\partial hT_{xx}}{\partial x}+\frac{\partial hT_{xy}}{\partial y}-\frac{\partial R_{xx}}{\partial x}-\frac{\partial R_{xy}}{\partial y}+\psi_1 \tag{2.2}$$

$$\frac{\partial hv}{\partial t}+\frac{\partial hvu}{\partial x}+\frac{\partial hv^2}{\partial y}= -gh\frac{\partial \eta}{\partial y}-\frac{\tau_{fy}}{\rho}-F_v-F_{vy}+\frac{\partial hT_{xy}}{\partial x}+\frac{\partial hT_{yy}}{\partial y}-\frac{\partial R_{xy}}{\partial x}-\frac{\partial R_{yy}}{\partial y}+\psi_2 \tag{2.3}$$

Here t is the time; x and y are the Cartesian coordinates; $h=\eta+d$ is the total water depth, where η is the surface elevation and d is the still water depth; u and v are the depth averaged velocity components in the x and y direction; g is the gravitational acceleration; ρ is the density of water. (τ_{fx},τ_{fy}) are the x- and y-components of the stresses due to bottom friction (see section 4.2); $\boldsymbol{F}=(F_u,F_v)$ is the flow resistance due to porosity (see section 4.4); $\boldsymbol{F}_v=(F_{vx},F_{vy})$ is the drag force due to vegetation (see section 4.3); R_{xx}, R_{xy} and R_{yy}, are additional terms due to the inclusion of wave breaking (see section 4.5). The lateral stresses, T_{xx}, T_{xy} and T_{yy}, include viscous friction, turbulent friction and differential advection. They are estimated using an eddy viscosity formulation based on the depth averaged velocities

$$T_{xx}=2\nu\frac{\partial u}{\partial x} \qquad T_{xy}=\nu\left(\frac{\partial u}{\partial y}+\frac{\partial v}{\partial x}\right) \qquad T_{yy}=2\nu\frac{\partial u}{\partial y} \tag{2.4}$$

where ν is the eddy viscosity. The Boussinesq terms (ψ_1,ψ_2) are expressed as

$$\psi_1=\left(B+\frac{1}{3}\right)d^2\left(\frac{\partial^3 hu}{\partial x^2 t}+\frac{\partial^3 hv}{\partial xyt}\right)+d\frac{\partial d}{\partial x}\left(\frac{1}{3}\frac{\partial^2 hu}{\partial xt}+\frac{1}{6}\frac{\partial^2 hv}{\partial yt}\right)+\frac{1}{6}d\frac{\partial d}{\partial y}\frac{\partial^2 hv}{\partial xt} + Bgd^3\left(\frac{\partial^3\eta}{\partial x^3}+\frac{\partial^3\eta}{\partial yyx}\right)+Bgd^2\frac{\partial d}{\partial x}\left(2\frac{\partial^2\eta}{\partial x^2}+\frac{\partial^2\eta}{\partial y^2}\right)+Bgd^2\frac{\partial d}{\partial y}\frac{\partial^2\eta}{\partial yx} \tag{2.5}$$

$$\psi_2 = \left(B + \frac{1}{3}\right) d^2 \left(\frac{\partial^3 hv}{\partial y^2 t} + \frac{\partial^3 hu}{\partial yxt}\right) + d\frac{\partial d}{\partial y}\left(\frac{1}{3}\frac{\partial^2 hv}{\partial yt} + \frac{1}{6}\frac{\partial^2 hu}{\partial xt}\right) + \frac{1}{6} d \frac{\partial d}{\partial x}\frac{\partial^2 hu}{\partial yt}$$
$$+ Bgd^3\left(\frac{\partial^3 \eta}{\partial y^3} + \frac{\partial^3 \eta}{\partial xxy}\right) + Bgd^2\frac{\partial d}{\partial y}\left(2\frac{\partial^2 \eta}{\partial y^2} + \frac{\partial^2 \eta}{\partial x^2}\right) + Bgd^2\frac{\partial d}{\partial x}\frac{\partial^2 \eta}{\partial xy} \quad (2.6)$$

To give a conservative formulation, the gravity surface terms are split into two terms (see Chippada (1998), Rogers (2001), Quecedo (2002), Liang or Borthwick (2009))

$$gh\frac{\partial \eta}{\partial x} = \frac{1}{2}g\frac{\partial(h^2 - d^2)}{\partial x} - g\eta\frac{\partial d}{\partial x} = \frac{1}{2}g\frac{\partial(\eta^2 + 2\eta d)}{\partial x} - g\eta\frac{\partial d}{\partial x} \quad (2.7)$$

$$gh\frac{\partial \eta}{\partial y} = \frac{1}{2}g\frac{\partial(h^2 - d^2)}{\partial y} - g\eta\frac{\partial d}{\partial y} = \frac{1}{2}g\frac{\partial(\eta^2 + 2\eta d)}{\partial y} - g\eta\frac{\partial d}{\partial y} \quad (2.8)$$

It is easily seen that if η is constant, the two terms cancel exactly. In the discrete case, this is also true if the two derivatives are calculated using the same scheme.

In matrix form the continuity equation and the momentum equations may be written

$$\frac{\partial h}{\partial t} + \nabla \cdot \boldsymbol{F}^c = 0 \quad (2.9)$$

$$\frac{\partial \boldsymbol{V}}{\partial t} + \nabla \cdot \boldsymbol{F}^m = \boldsymbol{S_0} + \boldsymbol{S_f} + \boldsymbol{S_\eta} + \frac{\partial \boldsymbol{\Psi}}{\partial \mathrm{t}} \quad (2.10)$$

Here $\boldsymbol{F}^c = \left(F_x^c, F_y^c\right)^T = (hu, hv)^T$, $\boldsymbol{V} = (hu, hv)^T$ and $\boldsymbol{F}^m = \boldsymbol{F}^{mc} - \boldsymbol{F}^{md} = (\boldsymbol{F}_x^m, \boldsymbol{F}_y^m)^T$.

The flux components and the source terms can be written

$$\boldsymbol{F}_x^{mc} = \begin{pmatrix} huu + \frac{1}{2}g(\eta^2 + 2\eta d) \\ hvu \end{pmatrix} \quad \boldsymbol{F}_y^{mc} = \begin{pmatrix} huv \\ hvv + \frac{1}{2}g(\eta^2 + 2\eta d) \end{pmatrix} \quad (2.11)$$

$$\boldsymbol{F}_x^{md} = \begin{pmatrix} 2hv\frac{\partial u}{\partial x} - R_{xx} \\ hv\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) - R_{xy} \end{pmatrix} \quad \boldsymbol{F}_y^{md} = \begin{pmatrix} hv\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) - R_{xy} \\ 2hv\frac{\partial v}{\partial y} - R_{yy} \end{pmatrix} \quad (2.12)$$

$$\boldsymbol{S_0} = \begin{pmatrix} g\eta\frac{\partial d}{\partial x} \\ g\eta\frac{\partial d}{\partial y} \end{pmatrix} \quad \boldsymbol{S_f} = \begin{pmatrix} -\frac{\tau_{fx}}{\rho} - F_u - F_{vx} \\ -\frac{\tau_{fy}}{\rho} - F_v - F_{vy} \end{pmatrix} \quad (2.13)$$

$$\boldsymbol{S_\eta} = \begin{pmatrix} Bgd^3\left(\frac{\partial^3 \eta}{\partial x^3} + \frac{\partial^3 \eta}{\partial yyx}\right) + Bgd^2\frac{\partial d}{\partial x}\left(2\frac{\partial^2 \eta}{\partial x^2} + \frac{\partial^2 \eta}{\partial y^2}\right) + Bgd^2\frac{\partial d}{\partial y}\frac{\partial^2 \eta}{\partial yx} \\ Bgd^3\left(\frac{\partial^3 \eta}{\partial y^3} + \frac{\partial^3 \eta}{\partial xxy}\right) + Bgd^2\frac{\partial d}{\partial y}\left(2\frac{\partial^2 \eta}{\partial y^2} + \frac{\partial^2 \eta}{\partial x^2}\right) + Bgd^2\frac{\partial d}{\partial x}\frac{\partial^2 \eta}{\partial xy} \end{pmatrix} \quad (2.14)$$

$$\boldsymbol{\Psi} = \begin{pmatrix} \left(B + \frac{1}{3}\right) d^2\left(\frac{\partial^2 hu}{\partial x^2} + \frac{\partial^2 hv}{\partial xy}\right) + d\frac{\partial d}{\partial x}\left(\frac{1}{3}\frac{\partial hu}{\partial x} + \frac{1}{6}\frac{\partial hv}{\partial y}\right) + \frac{1}{6}d\frac{\partial d}{\partial y}\frac{\partial hv}{\partial x} \\ \left(B + \frac{1}{3}\right) d^2\left(\frac{\partial^2 hv}{\partial y^2} + \frac{\partial^2 hu}{\partial yx}\right) + d\frac{\partial d}{\partial y}\left(\frac{1}{3}\frac{\partial hv}{\partial y} + \frac{1}{6}\frac{\partial hu}{\partial x}\right) + \frac{1}{6}d\frac{\partial d}{\partial x}\frac{\partial hu}{\partial y} \end{pmatrix} \quad (2.15)$$

3 Numerical method

The discretization in solution domain is performed using a cell-centered finite volume method (CCFV). The spatial domain is discretized by subdivision of the continuum into non-overlapping control volumes and by evaluating the field equations in integral form on these cells.

3.1 Mesh and discretization scheme

3.1.1 Mesh

The computational mesh is based on the unstructured meshes approach, which gives the maximum degree of flexibility. Control of node distribution allows for optimal usage of nodes and adaptation of mesh resolution to the relevant physical scales. The use of unstructured meshes also makes it possible to handle problems characterized by computational domains with complex boundaries.

The elements in the mesh can be triangles, quadrilateral elements or a combination of these (see Figure 3.1).

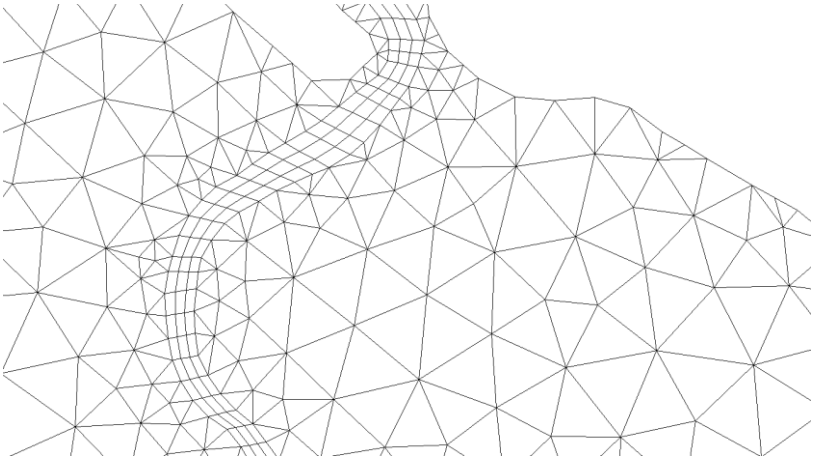

Figure 3.1 Unstructured mesh with both triangular and quadrilateral elements.

3.1.2 Discretization scheme

The discrete solution for the water depth, h, for the velocity components, u and v, are defined at the centroid of the elements in the mesh.

3.2 Finite volume method

The matrix form of the governing equations presented in Chapter 2 can be written as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{S} \tag{3.1}$$

Integrating Eq. 3.1 over the *i*th cell and using Gauss's theorem to rewrite the flux integral gives

$$\int_{A_i} \frac{\partial \boldsymbol{U}}{\partial t} d\Omega + \int_{\Gamma_i} (\boldsymbol{F}(\boldsymbol{U}) \cdot \boldsymbol{n})\, ds = \int_{A_i} \boldsymbol{S}(\boldsymbol{U}) d\Omega \tag{3.2}$$

where A is the area of the *i*th cell, Ω is the integration variable defined on A_i, Γ_i is the boundary of the *i*th cell and s is the integration variable along the boundary. $\boldsymbol{n} = (n_x, n_y)^T$ is the unit outward normal vector along the boundary. Evaluating the area integrals by a one-point quadrature rule, the quadrature point being the centroid of the cell, and evaluating the boundary integral using a mid-point quadrature rule, Eq. 3.2 can be written

$$\frac{\partial \boldsymbol{U}_i}{\partial t} + \frac{1}{A_i} \sum_{j}^{NF} \boldsymbol{F} \cdot \boldsymbol{n}_{ij} \Delta s_{ij} = \boldsymbol{S}_i \tag{3.3}$$

Here $\boldsymbol{U}_i$ and $\boldsymbol{S}_i$, respectively, are average values of $\boldsymbol{U}$ and $\boldsymbol{S}$ over the *i*th cell and stored at the cell centre. NF is the number of faces of the cell and the face ij is common to the cells associated with $\boldsymbol{U}_i$ and $\boldsymbol{U}_j$. Δs_{ij} is the length of the face ij, and $\boldsymbol{n}_{ij}$ is the restriction of $\boldsymbol{n}$ to the face ij.

3.3 Numerical solution of the flow equations

3.3.1 Space discretization of the enhanced Boussinesq equations

The space discretization is performed using the finite volume method as described in Section 3.2. The normal convective flux $\boldsymbol{F}_n(\boldsymbol{U}_L, \boldsymbol{U}_R) = \boldsymbol{F}(\boldsymbol{U}_L, \boldsymbol{U}_R) \cdot \boldsymbol{n}_{ij}$ across a face ij is determined using an approximate Riemann solver. The Riemann solver uses the variable $\boldsymbol{U} = (h, hu, hv)^T$ to the left and right of the face. The diffusive flux at the cell interfaces is approximated by a central scheme. The higher-order Boussinesq terms are approximated by successive application of the first-order derivative approximations used in the linear gradient-reconstruction technique, which is described in the following.

Reconstruction of face values

The variables, $\boldsymbol{U}_L$ and $\boldsymbol{U}_R$, to the left and right of a face are reconstructed from the cell values, $\boldsymbol{U}_i$ and $\boldsymbol{U}_j$, in two steps.

In the first step, the variables $\boldsymbol{U}_l$ and $\boldsymbol{U}_r$ are determined from element values. For a first order scheme, $\boldsymbol{U}_l = \boldsymbol{U}_i$ and $\boldsymbol{U}_r = \boldsymbol{U}_j$. Second-order spatial accuracy is achieved by employing a linear gradient-reconstruction technique for the primitive variables η, u and v. The face value at the vertical faces for a variable q in cell i is obtained by

$$q_l = q_i + \nabla q_i \cdot \boldsymbol{r}_{if}\,, \qquad q_r = q_j + \nabla q_j \cdot \boldsymbol{r}_{jf} \tag{3.4}$$

where $\boldsymbol{r}_{if}$ is the distance vector from the cell centre to the face and ∇q_i is the gradient vector. For estimation of the gradient vector, the Green-Gauss gradient approach is utilized. Here, the procedure proposed by Jawahar and Kamath (2000) is used. This procedure is based on a wide computational stencil to improve accuracy also for meshes with poor connectivity. The vertex (node) value is computed using the pseudo-Laplacian procedure proposed by Holmes and Connell (1989).

For a second-order scheme, the water depth d is also reconstructed at each side of the face using Eq. (3.4) and then the total water depths at the left and right of the face are defined as

$$h_l = \max(0, \eta_l + d_l), \qquad h_r = \max(0, \eta_r + d_r) \tag{3.5}$$

In the second step, the variables $\boldsymbol{U}_L$ and $\boldsymbol{U}_R$ are determined from $\boldsymbol{U}_l$ and $\boldsymbol{U}_r$. The water depth is assigned the same value at both sides of the face, $d_f = d_L = d_R$. Depending on the total water depth, two different techniques are used. As default, the average value of water depth is used

$$d_f = \frac{1}{2}(d_l + d_r) \tag{3.6}$$

and the total water depths are defined by

$$h_L = \max(\eta_l + d_f, 0), \qquad h_R = \max(\eta_r + d_f, 0). \tag{3.7}$$

If the total water height on either side is smaller than the difference in water depth, that is, if $h_l < |d_r - d_l|$ or $h_r < |d_r - d_l|$, the water depth is instead defined as in Chen & Noelle (2017) by

$$d_f = -\min(-\min(d_l, d_r), \min(\eta_l, \eta_r)) \tag{3.8}$$

and the total water depths are defined by

$$h_L = \min(\eta_l + d_f, h_l), \qquad h_R = \min(\eta_r + d_f, h_r). \tag{3.9}$$

Finally, the fluxes at each side of the face are determined from the velocities u_l, v_l, u_r, v_r and the total water depths h_L, h_R.

Riemann solver

The normal convective flux $\boldsymbol{F}_n(\boldsymbol{U})$ at the faces can be written

$$\boldsymbol{F}_n(\boldsymbol{U}) = \begin{pmatrix} hu_\perp \\ huu_\perp + \frac{1}{2}g(\eta^2 + 2\eta d)n_x \\ hvu_\perp + \frac{1}{2}g(\eta^2 + 2\eta d)n_y \end{pmatrix} \tag{3.10}$$

where $\boldsymbol{U} = (h, hu, hv)^T$ is the solution vector, and $u_\perp = un_x + vn_y$ is the velocity perpendicular to the cell face. This flux is reconstructed at cell-interfaces using the HLLC scheme introduced by Toro et al. (1994) for solving the Euler equations. The shock-capturing scheme enables robust and stable simulation of flows involving shocks or discontinuities such as bores and hydraulic jumps. The interface flux is computed as follows (see Toro (2001))

$$\boldsymbol{F}(\boldsymbol{U}_L, \boldsymbol{U}_R) \cdot \boldsymbol{n} = \begin{cases} \boldsymbol{F}_L & if\ S_L \geq 0 \\ \boldsymbol{F}_{*L} & if\ S_L < 0 \leq S_* \\ \boldsymbol{F}_{*R} & if\ S_* < 0 \leq S_R \\ \boldsymbol{F}_R & if\ S_R \leq 0 \end{cases} \tag{3.11}$$

where $\boldsymbol{F}_L = \boldsymbol{F}_n(\boldsymbol{U}_L)$ and $\boldsymbol{F}_R = \boldsymbol{F}_n(\boldsymbol{U}_R)$ are calculated from Eq. (3.10), and the middle region fluxes, $\boldsymbol{F}_{*L}$ and $\boldsymbol{F}_{*R}$ are given by

$$\boldsymbol{F}_{*L} = \begin{pmatrix} e_1 \\ e_2 n_x - u_{\|L} e_1 n_y \\ e_2 n_y + u_{\|L} e_1 n_x \end{pmatrix} \tag{3.12}$$

$$\boldsymbol{F}_{*R} = \begin{pmatrix} e_1 \\ e_2 n_x - u_{\parallel R} e_1 n_y \\ e_2 n_y + u_{\parallel R} e_1 n_x \end{pmatrix} \tag{3.13}$$

Here $u_{\parallel} = -un_y + vn_x$ is the velocity tangential to the cell face, and (e_1, e_2) is the component of the normal flux which is calculated using the HLL solver proposed by Harten et al. (1983)

$$\boldsymbol{E} = \frac{S_R \widehat{\boldsymbol{E}}_L - S_L \widehat{\boldsymbol{E}}_R + f_{HLLC} S_L S_R (\widehat{\boldsymbol{U}}_R - \widehat{\boldsymbol{U}}_L)}{S_R - S_L} \tag{3.14}$$

Here $\widehat{\boldsymbol{U}} = (h, hu_{\perp})^T$ and $\widehat{\boldsymbol{E}} = \left(hu_{\perp}, hu_{\perp}u_{\perp} + \frac{1}{2}g(\eta^2 + 2\eta d)\right)^T$. To be able to scale the damping introduced by the HLLC solver a scaling factor f_{HLLC} has been introduced, where the factor must be in the interval [0,1]. The scaling factor, $f_{HLLC} = 1$, corresponds to the standard HLLC solver.

An appropriate method for approximating the wave speeds is essential for the efficiency of the HLLC solver. Different approximations can be found in the literature, e.g. Fraccarollo and Toro (1994). Here the approach used by Song et al. (2011) is used

$$S_L = \begin{cases} u_{\perp R} - 2\sqrt{gh_R} & h_L = 0 \\ min(u_{\perp L} - \sqrt{gh_L}, u_{\perp *} - \sqrt{gh_*}) & h_L > 0 \end{cases} \tag{3.15}$$

and

$$S_R = \begin{cases} u_{\perp L} + 2\sqrt{gh_L} & h_R = 0 \\ max(u_{\perp R} + \sqrt{gh_R}, u_{\perp *} + \sqrt{gh_*}) & h_R > 0 \end{cases} \tag{3.16}$$

where the Roe-averaged quantities

$$u_{\perp *} = \frac{u_{\perp L}\sqrt{h_L} + u_{\perp R}\sqrt{h_R}}{\sqrt{h_L} + \sqrt{h_R}} \tag{3.17}$$

$$h_* = \frac{1}{2}(h_L + h_R) \tag{3.18}$$

The wave speed S_* is given by the

$$S_* = \frac{S_L h_R (u_{\perp R} - S_R) - S_R h_L (u_{\perp L} - S_L)}{h_R (u_{\perp R} - S_R) - h_L (u_{\perp L} - S_L)} \tag{3.19}$$

3.3.2 Time integration of the enhanced Boussinesq equations

The time integration of the enhanced Boussinesq equations is performed using an explicit scheme. Here a two-stage explicit second-order Runge-Kutta scheme (the midpoint method) are applied, and in each stage a fractional step approach is used. In the first step the water depth and the estimated depth integrated velocities is calculated using the shallow water equations and in the second step the depth integrated velocities is corrected taking into account the dispersion terms.

Stage 1:

$$\frac{h^{n+1/2}-h^n}{\Delta t/2} = -\left(\frac{\partial F_x^c}{\partial x}+\frac{\partial F_y^c}{\partial y}\right)^n \tag{3.20}$$

$$\frac{\boldsymbol{V}^*-\boldsymbol{V}^n}{\Delta t/2} = -\left(\frac{\partial \boldsymbol{F}_x^m}{\partial x}+\frac{\partial \boldsymbol{F}_y^m}{\partial y}\right)^n + \boldsymbol{S}_0^n + \boldsymbol{S}_f^* \tag{3.21}$$

$$\frac{\boldsymbol{V}^{n+1/2}-\boldsymbol{V}^*}{\Delta t/2} = \boldsymbol{S}_\eta^\alpha + \frac{\boldsymbol{\Psi}^{n+1/2}-\boldsymbol{\Psi}^n}{\Delta t/2} \tag{3.22}$$

Stage 2:

$$\frac{h^{n+1}-h^n}{\Delta t} = -\left(\frac{\partial F_x^c}{\partial x}+\frac{\partial F_y^c}{\partial y}\right)^{n+1/2} \tag{3.23}$$

$$\frac{\boldsymbol{V}^{**}-\boldsymbol{V}^n}{\Delta t} = -\left(\frac{\partial \boldsymbol{F}_x^m}{\partial x}+\frac{\partial \boldsymbol{F}_y^m}{\partial y}\right)^{n+1/2} + \boldsymbol{S}_0^{n+1/2} + \boldsymbol{S}_f^{**} \tag{3.24}$$

$$\frac{\boldsymbol{V}^{n+1}-\boldsymbol{V}^{**}}{\Delta t} = \boldsymbol{S}_\eta^\beta + \frac{\boldsymbol{\Psi}^{n+1}-\boldsymbol{\Psi}^n}{\Delta t} \tag{3.25}$$

where $\boldsymbol{S}_\eta^\alpha$ is evaluated using the averaged surface elevation $\frac{1}{2}(\eta^{n+1/2}+\eta^n)$, and $\boldsymbol{S}_\eta^\beta$ is evaluated similarly as $\frac{1}{2}(\eta^{n+1}+\eta^n)$. The special treatment of the friction source term is discussed in section 3.3.3. In the last step of each stage, i.e. Eq. (3.22) and (3.25), a linear system of equations has to be solved as, e.g., $\boldsymbol{\Psi}^{n+1}=\boldsymbol{\Psi}(\boldsymbol{V}^{n+1})$. The resulting sparse linear systems are solved using a preconditioned iterative solver from the PETSc library, Balay (2017). More specifically, the iterative solver is a flexible Generalized Minimal Residual method (FGMRES). See Chapter 0 for further details on single- and multi-subdomain simulations.

Due to the explicit scheme, the time step interval, Δt, is restricted by the Courant-Friedrichs-Lewy (CFL) condition

$$C = \Delta t \frac{\left(\sqrt{gh}+|u|\right)+\left(\sqrt{gh}+|v|\right)}{\Delta l} \leq C_{max} \tag{3.26}$$

where C is the Courant number and Δl is a characteristic length. C_{max} is the maximum Courant number and must be less than or equal to 1. A variable time step interval is used in the time integration of the enhanced Boussinesq equations and determined so that the Courant number is less than a maximum Courant number in all computational nodes. The characteristic length for a quadrilateral element, is determined as the area of the element divided by the longest edge length of the element. For a triangular element, the characteristic length is two times the area divided by the longest edge length.

3.3.3 Friction source term discretization

The contribution to the source term due to bed resistance and flow resistance can be written (see section 4.2, 4.3 and 4.4)

$$\boldsymbol{S}_f = -\begin{pmatrix} \frac{\tau_{fx}}{\rho} + F_u + F_{vx} \\ \frac{\tau_{fy}}{\rho} + F_v + F_{vx} \end{pmatrix} \quad (3.27)$$

A fully explicit treatment of the friction terms can cause instability for very shallow water. Hence, this term is treated fully implicit. For each element at each stage in the time integration procedure the water depth is updated and then the two momentum equations are solved using a Newton-Raphson method. The stopping criteria is that the 2-norm of the residual vector is less than 10^{-10}.

3.3.4 Flooding and drying

The approach for treatment of the moving boundaries (flooding and drying fronts) problem is based on the work by Zhao et al. (1994) and Sleigh et al. (1998). When the depths are small the problem is reformulated, and only when the depths are very small the elements/cells are removed from the calculation. The reformulation is made by setting the momentum fluxes to zero and only taking the mass fluxes into consideration.

The depth in each element is monitored and the elements are classified as dry, partially dry or wet. Also, the element faces are monitored to identify flooded element faces.

- An element face is defined as flooded if the water depth at one side of a face is less than a tolerance depth, h_{dry}, and the water depth at the other side of the face is larger than a tolerance depth, h_{wet}.
- An element is dry if the water depth is less than a tolerance depth, h_{dry}, and none of the element faces are flooded faces. The element is removed from the calculation.
- An element is partially dry if the water depth is larger than h_{dry} and less than a tolerance depth, h_{wet}, or when the depth is less than h_{dry}, and one of the element faces is a flooded face. The momentum fluxes are set to zero, and only the mass fluxes are calculated.
- An element is wet if the water depth is bigger than h_{wet}. Both the mass flux and the momentum flux are calculated.

A non-physical flow across the face will be introduced for a flooded face when the surface elevation in the wet element on one side of the face is lower than the bed level in the partially wet element on the other side. To overcome this problem the face will be treated as a closed boundary (Section 3.3.7).

In case the water depth becomes negative, the water depth is set to zero, and the water is subtracted from the adjacent elements to maintain mass balance. When this occur the water depth at the adjacent elements may become negative. Therefore, an iterative correction of the water depth is applied (max. 100 iterations). Normally only one or a few correction steps are needed.

3.3.5 Sponge layer

Sponge (or absorbing) layers can be used as efficient numerical wave absorbers in wave simulations. These could be set up along model boundaries to provide radiation boundary conditions, which absorb wave energy propagating out of the model area.

The implemented method is based on the sponge layer technique introduced by Larsen and Dancy (1983). In the sponge layer the calculated surface elevation, η, and the velocities u, and v, are corrected at every time step as

$$\eta = \frac{\eta - \eta^{ref}}{c} + \eta^{ref} \qquad u = \frac{u}{c} \qquad v = \frac{v}{c} \tag{3.28}$$

where c is the sponge coefficient, and η^{ref} is the reference level.

To minimize reflections, the values of the sponge layer coefficient, c, should be close to unity along the front edge of the sponge layer and should increase smoothly towards the closed/land boundary. When selecting the sponge layer coefficient, c, the following formula has been found to work well

$$c = a^{r^{s/\Delta s}}, \quad 0 \le s \le w \tag{3.29}$$

where w is the width of the sponge layer, and a and r are assigned constant values. s is the distance from the closed boundary, and Δs is the characteristic size of the elements in the sponge layer area. Depending on ratio $w/\Delta s$ you may use the values listed in Table 3.1.

Table 3.1 Recommended values for sponge layer coefficients

w/Δs	a	r
10	5	0.5
20	7	0.7
50	10	0.85
100	10	0.92
200	10	0.95

3.3.6 Internal wave generation

The relaxation zone technique is applied for wave generation and absorption. Here a relaxation function is applied to introduce the analytical solution for the incoming waves smoothly into the calculation domain. The analytical solution is the target solution and contains values from the chosen wave theory for the surface elevation, the velocities and the pressure. The relaxation zone is defined as the area to the right of the polyline when positioned at the starting point and looking forward along the line (see Figure 3.2). The target and the computed solution are weighted in the relaxation zone after each step in the time integration

$$\theta = \alpha\theta_{Computed} + (1 - \alpha)\theta_{Target} \tag{3.30}$$

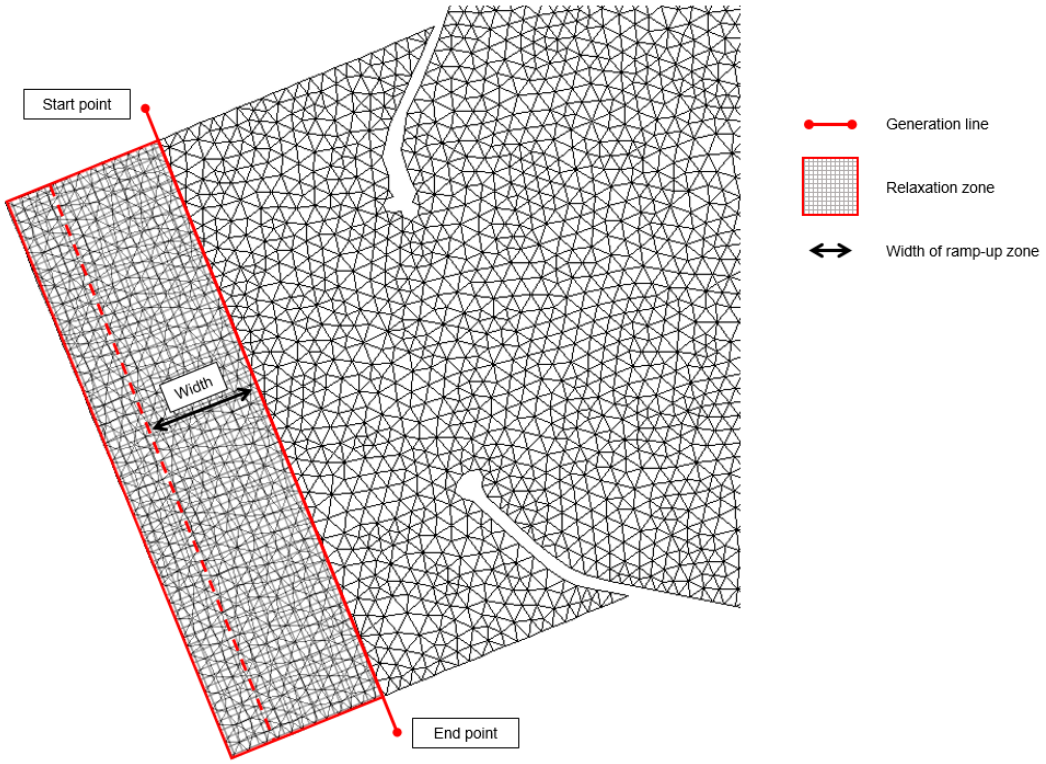

Figure 3.2 The relaxation zone is the area to the right of the generation line when looking forward along the line from the starting point. The width of the ramp-up zone is specified by the width parameter.

where θ represents the surface elevation, velocity component and non-hydrostatic pressure. For the surface elevation and velocities the ramp up factor, α, is given as

$$\begin{aligned} \alpha &= 1 - \frac{exp(s^f)-1}{\exp(1)-1} && 0 \le s \le 1 \\ \alpha &= 0 && s > 1 \end{aligned} \tag{3.31}$$

Here, s is the distance from the polyline divided by the width of the ramp up zone, and f is the ramp up factor. The value $f = 3.5$ is applied. For the non-hydrostatic pressure, q, the target value is applied as a Dirichlet condition for $s > 1$. Hence, here the ramp up factor is given by

$$\begin{aligned} \alpha &= 1 && 0 \le s \le 1 \\ \alpha &= 0 && s > 1 \end{aligned} \tag{3.32}$$

For unidirectional regular waves Stokes theory (up to 5th order), Cnoidal theory and Boussinesq theory (up to 3rd order) can be applied. For irregular waves the single summation method is applied. Here a single direction is assigned to each discrete frequency. A range of standard formulations for the frequency spectrum and the directional distribution are applied.

3.3.7 Boundary conditions

At the lateral closed (solid) boundaries a free-slip boundary condition is imposed for the velocities: The normal flux is zero, and the tangential stress is set to zero. The normal flux at a closed boundary is therefore given as

$$\boldsymbol{F}_n(\boldsymbol{U}) = \begin{pmatrix} 0 \\ \frac{1}{2}g(\eta^2 + 2\eta d)n_x \\ \frac{1}{2}g(\eta^2 + 2\eta d)n_y \end{pmatrix} \tag{3.33}$$

Furthermore, the boundary conditions for higher-order surface elevation terms are defined as homogeneous gradients normal to the closed boundaries.

3.4 Time stepping procedure

The solution is determined at a sequence of discrete times

$$t^k = t^0 + k\Delta t_{overall} \qquad k = 0, 1, 2, 3 \ldots \tag{3.33}$$

where $\Delta t_{overall}$ is the overall time step interval. The time steps for the hydrodynamic calculations are dynamic.

At the actual time t in the interval $t^{k-1} < t \leq t^k$ the new time step interval is determined using the following procedure

$$\Delta t^* = C_{max} min\left(\frac{\Delta l}{\left(\sqrt{gh_i} + |u_i|\right) + \left(\sqrt{gh_i} + |v_i|\right)}\right) \tag{3.34}$$

$$\Delta t^{**} = min\left(max(\Delta t^*, \Delta t_{min}), \Delta t_{max}\right) \tag{3.35}$$

$$\Delta t = \frac{t^k - t}{int\left(\frac{t^k - t}{\Delta t^{**}}\right) + 1} \tag{3.34}$$

Here Δt_{min} and Δt_{max} are the minimum and maximum time steps, respectively, and int is the whole number of $(t^k - t)/\Delta t^{**}$. This procedure secures that the time steps for the hydrodynamic calculations are synchronized at the overall discrete time steps.

4 Physics

4.1 Eddy viscosity

In some applications a constant eddy viscosity can be used for the horizontal eddy viscosity. Alternatively, Smagorinsky (1963) proposed to express sub-grid scale transports by an effective eddy viscosity related to a characteristic length scale. The subgrid scale eddy viscosity is given by

$$\nu = c_s^2 l^2 \sqrt{2\left(S_{xx}S_{xx} + 2S_{xy}S_{xy} + S_{yy}S_{yy}\right)} \tag{4.1}$$

where c_s is a constant, l is a characteristic length and the deformation rate is given by

$$S_{xx} = \frac{\partial u}{\partial x} \quad S_{xy} = \frac{1}{2}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \quad S_{yy} = \frac{\partial v}{\partial y} \tag{4.2}$$

For more details on this formulation, the reader is referred to Lilly (1966), Leonard (1974), Aupoix (1984), and Horiuti (1987).

4.2 Bed resistance

The stress due to friction at the bed, $\boldsymbol{\tau}_f = (\tau_{fx}, \tau_{fy})$ is determined by a quadratic friction law

$$\frac{\boldsymbol{\tau}_f}{\rho} = c_{fb}\boldsymbol{u}|\boldsymbol{u}| \tag{4.3}$$

where c_{fb} is the drag coefficient and $\boldsymbol{u} = (u, v)$ is the depth-averaged flow velocity. The drag coefficient is determined from the Chezy number, C, or the Manning number, M

$$c_{fb} = \frac{g}{C^2} \tag{4.4}$$

$$c_{fb} = \frac{g}{(Mh^{1/6})^2} \tag{4.5}$$

4.3 Vegetation

The vegetation structure is modelled as rigid or flexible stems with stem diameter, d_s, or as flexible blades (leaves) with blade width, w_b, and blade thickness, t_b. The height of the vegetation is h_v.

The effect of the vegetation on the flow characteristics is modelled by inclusion of the following drag force in the depth integrated momentum equations

$$\boldsymbol{F}_v = \frac{1}{2} C_D h_v^* b_v \boldsymbol{N}_v \boldsymbol{u}_v |\boldsymbol{u}_v| \tag{4.6}$$

where C_D is the drag coefficient, b_v is the plant size, N_v is the vegetation density and $\boldsymbol{u}_v$ is the apparent velocity vector in the vegetation region. h_v^* is given as $h_v^* = min(h_v, h)$,, where h is the water depth. The plant size is either the stem diameter or the blade width.

The vegetation density is the number of plants per unit area. Stone and Chen (2002) proposed the following expression for the apparent velocity

$$\boldsymbol{u}_v = \left(\frac{h_v}{h}\right)^p \boldsymbol{u} \tag{4.7}$$

where $\boldsymbol{u}$ is the flow velocity vector and p=1/2.

For rigid stems a layered approach can be used to take into account the vertical variation of the vegetation. The drag coefficient, $C_{D,i}$, the stem diameter, $d_{s,i}$, the vegetation height, $h_{v,i}$, and the vegetation density, $N_{v,i}$, are then specified for each vertical layer, i. The vegetation height is the distance from the bed to the top of the vegetation layer. The dissipation term due to vegetation is then determined as

$$\boldsymbol{F}_v = \left(\sum_{i=1}^{nlayers} \frac{1}{2} C_{D,i} d_{s,i} (h^*_{v,i} - h^*_{v,i-1}) N_{v,i} \right) \boldsymbol{u}_v |\boldsymbol{u}_v| \tag{4.8}$$

where $h_{v,0} = 0$.

The reduction of the drag due to flexibility of the vegetation is taken into account using the approach by Luhar and Nepf (2011, 2013). They suggested the use of a deflected height, h_d, and an effective length, l_e. The effective length is defined as the length of a rigid vertical plant that generates the same drag as the total length of a flexible plant. The deflected height and the effective length are given by

$$\frac{h_d}{h_v} = 1 - \frac{1 - Ca^{-1/4}}{1 + Ca^{-3/5}(4 + B^{3/5}) + Ca^{-2}(8 + B^2)} \tag{4.9}$$

$$\frac{l_e}{h_v} = 1 - \frac{1 - 0.9Ca^{-1/3}}{1 + Ca^{-3/2}(8 + B^{3/2})} \tag{4.10}$$

where Ca is the Cauchy number and B is the buoyancy parameter

$$Ca = \frac{\rho A |\boldsymbol{u}_v|^2}{EI/h_v^2} \qquad B = \frac{(\rho - \rho_v) g V_p}{EI/h_v^2} \tag{4.11}$$

Here g is the gravitational acceleration, ρ is the density of water, ρ_v is the density of the plant, E is the elastic modulus for the plant, A is the frontal area, V_p is the volume of the plant element and I is the second moment of the area. For a circular stem

$$A = d_s h_v \qquad V_p = d_s^2 h_v / 4 \qquad I = \pi d_s^4 / 64 \tag{4.12}$$

and for a blade

$$A = w_b h_v \qquad V_p = w_b t_b h_v \qquad I = w_b t_b^3 / 12 \tag{4.13}$$

The flexibility is taken into account by using h_d instead of h_v in Eq. (4.6) and (4.7) and introducing a factor l_e/h_d in Eq. (4.6).

4.4 Porosity

For wave simulations, the governing equations have been modified to include porosity and the effects of non-Darcy flow through porous media. In this way, it is possible to model partial reflection, absorption and transmission of wave energy at porous structures such as rubble mound breakwaters.

The main effects of porosity are introduced by additional laminar and turbulent friction terms for describing losses due to flow through a porous structure. In most practical cases the pore sizes are relatively large (typically 0.1m to 1.0m), and the turbulent losses will dominate. The laminar loss term has also been included to allow the simulation of small scale physical model tests.

The flow resistance components $\boldsymbol{F} = (\boldsymbol{F}_u, \boldsymbol{F}_v)^T$ inside the porous structure are described by the linear and non-linear resistance forces expressed as

$$\boldsymbol{F} = ah\boldsymbol{u} + b|\boldsymbol{u}|h\boldsymbol{u} \tag{4.14}$$

where a and b are resistance coefficients accounting for the laminar and turbulent friction loss, respectively, $\boldsymbol{u} = (u, v)$ is the filter velocity vector, and the magnitude of the filter velocity is defined by $|\boldsymbol{u}| = \sqrt{u^2 + v^2}$. a and b are determined by following the approach by Engelund (1954)

$$a = \alpha \frac{(1-n)^2}{n^3} \frac{\vartheta}{D_{50}^2} \tag{4.15}$$

$$b = \beta \frac{(1-n)}{n^3} \frac{1}{D_{50}} \tag{4.16}$$

Where n is the porosity, α and β are user specified coefficients, ϑ is the kinematic viscosity and D_{50} is the grain diameter of the porous materials. KC is the Keulegan-Carpenter number defined as

$$KC = \frac{u_m T}{n D_{50}} \tag{4.17}$$

where u_m is the maximum oscillating velocity, and T is the period of the oscillation. u_m is approximated by the magnitude of the filter velocity.

In the momentum equations the time derivative terms are multiplied by a factor $(1 + C_m)$ where C_m is the added mass coefficient to take transient interaction between grains and water into account. van Gent (1995) gave C_m as

$$c_m = \gamma \frac{1-n}{n} \tag{4.18}$$

where γ is an empirical coefficient, which takes the value 0.34.

4.5 Wave breaking

The momentum terms that account for the excess momentum originating from the non-uniform velocity distribution due to the presence of a roller (see Figure 4.1) are defined as

$$R_{xx} = \frac{\delta}{1-\frac{\delta}{h}}(c_x - u)^2 \tag{4.19}$$

$$R_{xy} = \frac{\delta}{1-\frac{\delta}{h}}(c_x - u)(c_y - v) \tag{4.20}$$

$$R_{yy} = \frac{\delta}{1-\frac{\delta}{h}}(c_y - v)^2 \tag{4.21}$$

where $\delta = \delta(x, y, t)$ is the thickness of the surface roller and (c_x, c_y) are components of the roller celerity. A detailed description of these quantities is given in Madsen et al. (1997) and Sørensen et al. (2004).

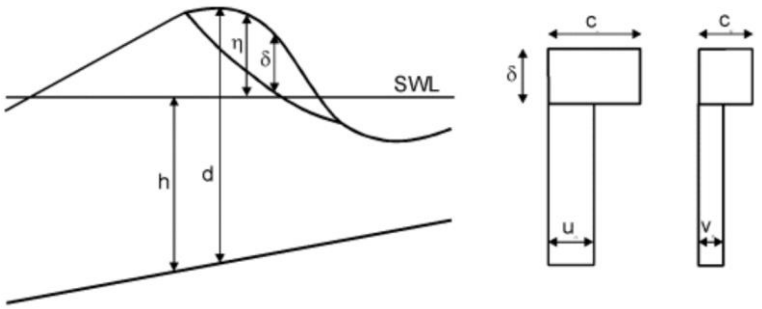

Figure 4.1 Surface roller concept: cross section of a breaking wave and assumed vertical profile of the horizontal particle velocity components.

The roller celerity (c_x, c_y) is an essential parameter in surface roller model used in MIKE 21 Wave Model FM. A similar formulation as the one presented in Sørensen et al. (2004) is used and is based on the approximations

$$(c_x, c_y) = (c\cos\theta, c\sin\theta) \tag{4.22}$$

where θ is the roller direction, $c = f_v\sqrt{gh}$ is the approximated magnitude of the wave celerity and f_v is the roller celerity factor. Using a roller celerity factor $f_v = 1.0$, we obtain the celerity determined by shallow water theory. This is often a rather good approximation just outside the surf zone, while $f_v = 1.3$ is more appropriate inside the surf zone according to the discussion in Madsen et al. (1997). The transition from $f_v = 1.0$ to $f_v = 1.3$ is modelled with an exponentially decaying time variation similar to the time variation of the slope of the breaking angle, which can be expressed analytically as

$$\tan\phi(t) = \tan\phi_0 + (\tan\phi_B - \tan\phi_0)\exp\left(-\ln(2)\frac{t - t_B}{t_{1/2}}\right) \tag{4.23}$$

where ϕ_B is the initial breaking angle, ϕ_0 is the final breaking angle, t_B is the time instant of initialized wave breaking, and $t_{1/2}$ is the time scale for the development of the roller.

The roller direction can be defined either as a predefined direction according to an expected mean direction of the roller, or computed interactively by

$$(c_x, c_y) = -sign(c, \partial\eta/\partial t)\frac{(\partial\eta/\partial x\,, \partial\eta/\partial y)}{|\nabla\eta|} \tag{4.24}$$

where $|\nabla\eta|$ is the magnitude of the surface elevation gradient and $sign(c, \partial\eta/\partial t)$ returns c with the sign of the surface elevation time derivative.

5 Parallelization

The MIKE 21 Wave Model FM is parallelized for shared-memory multiprocessor/multicore computers using OpenMP. This parallelization is performed by adding compiler directives to the code. To improve performance and to be able to perform simulations on large massively parallel distributed-memory computers and clusters, MIKE 21 Wave Model FM has also been parallelized using domain decomposition concept and Message Passing Interface (MPI). Given the number of processor cores allocated to a simulation, the computational mesh is partitioned into subdomains, and the workload associated with each domain is distributed between the allocated cores. The data exchange between domains is performed by message passing using the Intel MPI Library, which has multi fabric message passing capabilities. It allows the use of mixed communication between the domains. Thus, domains will exchange data via the fastest communication interface – in ranked order: shared memory, InfiniBand, Ethernet, etc.. The implementation uses a hybrid approach (OpenMP and MPI).

5.1 The domain decomposition

The domain partitioning is performed using the METIS graph partitioning library (Karypis and Kumar, (1998, 1999)). The computational mesh is converted into a graph, and then METIS uses a multi-level graph partitioning scheme to split the graph into subgraphs, representing the partitioned subdomains, which are distributed among the allocated cores. METIS computes a balanced partitioning that minimizes the connectivity of the subdomains and the difference in the number of elements in all subdomains.

The chosen numerical scheme for the discretization in the spatial domain requires an overlapping domain decomposition. It is based on the halo-layer (“ghost”-cells) approach, where each subdomain contains elements from connected subdomains. This overlap is needed, because calculations require values from the connecting elements. Thus, calculations of some elements at the border between subdomains require values from the connected subdomains.

5.2 Data exchange

The data exchange between processes is based on the aforementioned halo-layer (“ghost”-cells) approach with overlapping elements. The extension of the halo-layer area depends on the numerical scheme used for the discretization in the spatial domain and which variables are chosen to be exchanged between subdomains. Here a two-element wide halo-layer is applied. The data exchanges are performed via asynchronous communication when possible, and synchronous communications are used in different parts of the system to ensure correct execution. The MIKE 21 Wave Model FM uses a dynamic time step in the time integration scheme. To ensure that the calculations are performed with the same time step in all subdomains, time step information is exchanged between processes and thereby synchronizing the processes of each time step. Several special features require additional data exchange. These special interest points cause synchronization of two or more subdomains during the data exchange. The case of input and output data exchange is mentioned in the next subsection. Finally, information is exchanged between subdomains in connection with error handling. When the system encounters an error in the model, the error is distributed to the other processes when the time step is finished and the simulation is stopped.

5.3 Input and output

The input and output (I/O) is handled using a parallel I/O approach. The master process reads the global mesh information, performs the partitioning of the mesh and distributes the information about the individual subdomains to the slave processes. Each process then reads the additional input specifications using the generic specification file. The input data (porosity maps, sponge layer maps, etc.) are read by each process using the global data files. Since the individual processes perform I/O locally, the simulation data files must be accessible by each process. This access could be through a network-attached storage system or locally on each computer. The output data files from the simulations are written to private files for each subdomain. At the end of the simulation, the data files are merged to obtain data files containing global information.

6 References

/1/ Aupoix, B. (1984), Eddy Viscosity Subgrid Scale Models for HomogeneousTurbulence, in Macroscopic Modelling of Turbulent Flow, Lecture Notes in Physics, Proc. Sophie-Antipolis, France.

/2/ Balay, S. K. (2017), *PETSc User Manual*, Tech. Rep. ANL-95/11-Revision 3.8, Mathematics and Computer Science Division., Argonne National Lab.

/3/ Chen, G., & Noelle, S. (2017). *A new hydrostatic reconstruction scheme based on subcell reconstructions*. SIAM Journal on Numerical Analysis, 55(2), 758-784.

/4/ Chippada, S., Dawson, C.N., Martinez, M.L. and Wheeler, M.F. (1998), *A Godunov-type finite volume method for the system of Shallow Water Equations*, Computational Methods in Applied Mechanics and Engineering, 151, 105-129.

/5/ Engelund, F., (1954). On the laminar and turbulent flows of ground water through homogeneous sand. Technical Report.Danish Academy of Technical Sciences.

/6/ Fraccarollo I., Toro E.F. (1994), *Experimental and numerical assessment of the shallow water model for two-dimensional dam-break type problems*, Journal of Hydraulic Research 33, 951-979.

/7/ Harten A., Lax P.D., Van Leer B. (1983), *On upstream differencing and Godunov-type schemes for hyperbolic conservation-laws*, SIAM Rev 25(1), 54-74.

/8/ Holmes, D. G. and Connell, S. D. (1989), *Solution of the 2D Navier-Stokes on unstructured adaptive grids*, AIAA Pap. 89-1932 in Proc. AIAA 9th CFD Conference.

/9/ Horiuti, K. (1987), Comparison of Conservative and Rotational Forms in Large Eddy Simulation of Turbulent Channel Flow, Journal of Computational Physics, 71, pp 343-370.

/10/ Jawahar P. and Kamath H. (2000), *A high-resolution procedure for Euler and Navier-Stokes computations on unstructured grids*, Journal of Computational Physics, 164, 165-203.

/11/ Karypis G., Kumar .V. (1998), *METIS: family of multilevel partitioning algorithms*, Available from: http://glaros.dtc.umn.edu/gkhome/views/metis

/12/ Karypis G., V. Kumar (1999), *A Fast and Highly Quality Multilevel Scheme for Partitioning Irregular Graphs*, SIAM Journal on Scientific Computing, Vol. 20, No. 1, 1999, pp. 359—392.

/13/ Larsen, J., & Dancy, H. (1983), *Open Boundaries in Short-wave Simulations - A New Approach*, Coastal Engineering, 7, 285-297.

/14/ Leonard, A. (1974), Energy Cascades in Large-Eddy Simulations of Turbulent Fluid Flows, Advances in Geophysics, 18, pp 237-247.

/15/ Liang Q., Borthwick A.G.L, (2009), *Adaptive quadtree simulation of shallow flows with wet-dry fronts over complex topography*, Computers and Fluids 38, 221-234.

/16/ Lilly, D.K. (1966), On the Application of the Eddy Viscosity Concept in the Inertial Subrange of Turbulence, NCAR Manuscript No. 123, National Center for Atmospheric Research, Boulder, Colorado.

/17/ Luhar M, Nepf H. M. (2011), Flow-induced reconfiguration of buoyant and flexible aquatic vegetation. Limnol Oceanogr 56(6), 2003–17.

/18/ Luhar M. Nepf H.M (2013), From the blade scale to the reach scale: A characterization of aquatic vegetative drag, Advances in Water Resources 51, 305–316.

/19/ Madsen, P. A., Murray, R., and Sørensen, O. R. (1991), A new form of the Boussinesq equations with improved linear dispersion characteristics. Coastal Engineering, 15, 4, 371-388.

/20/ Madsen, P. A., and Sørensen, O. R. (1992), A new form of the Boussinesq equations with improved linear dispersion characteristics. Part 2. A slowly-varying bathymetry. Coastal Engineering, 18, 3-4, 183-204.

/21/ Madsen, P. A., Sørensen, O. R., and Schäffer, H. A. (1997), Surf zone dynamics simulated by a Boussinesq type model. Part I. Model description and cross-shore motion of regular waves. Coastal Engineering, 32, 4, 255-287.

/22/ Quecedo, M. and Pastor, M. (2002), *A reappraisal of Taylor-Galerkin algorithm for drying-wetting areas in shallow water computations*, International Journal for Numerical Methods in Fluids, 38, 515-531.

/23/ Rogers, B., Fujihara, M. and Borthwick, A.G.L. (2001), *Adaptive Q-tree Godunov-type scheme for shallow water equations*, International Journal for Numerical Methods in Fluids, 35, 247-280.

/24/ Sleigh, P.A., Gaskell, P.H., Bersins, M. and Wright, N.G. (1998), *An unstructured finite-volume algorithm for predicting flow in rivers and estuaries*, Computers & Fluids, Vol. 27, No. 4, 479-508.

/25/ Smagorinsky J. (1963), *General Circulation Experiment with the Primitive Equations*, Monthly Weather Review, 91, No. 3, 99-164.

/26/ Song, L. Zhou J., Guo J., Zou Q., and Liu Y. (2011), *A robust well-balanced finite volume model for shallow water flows with wetting and drying over irregular terrain*, Advances in Water Resources, vol. 34, no. 7, 915–932.

/27/ Stone B.M. Shen H.T. (2002), Hydraulic resistance of flow in channels with cylindrical roughness, Journal of Hydraulic Engineering, 128(5), 500-506.

/28/ Sørensen, O. R., Schäffer, H. A., and Sørensen, L. S. (2004), Boussinesq-type modelling using an unstructured finite element technique. Coastal Engineering, 50, 4, 181-198.

/29/ Toro, E.F. (2001), *Shock-capturing methods for free-surface shallow flows*, Chichester, John Wiley & Sons.

/30/ Toro, E.F., Spruce, M., Speares, W. (1994). Restoration of the contact surface in the HLL-Riemann solver. Shock Waves 4, 25–34.

/31/ Zhao, D.H., Shen, H.W., Tabios, G.Q., Tan, W.Y. and Lai, J.S. (1994), *Finite volume 2-dimensional unsteady-flow model for river basins*, Journal of Hydraulic Engineering, ASCE, 1994, 120, No. 7, 863-833.