Chapter 2

The Morfo70 Model

2.1 Introduction

Numerical modelling is a powerful tool to understand the mechanisms and the role of the different processes involved in coastal morphodynamics. Shoreline and nearshore morphodynamics are the result of the combination and the coupling of a large amount of physical processes related with nearshore hydrodynamics and sediment transport. Nowadays, some of these processes are not entirely understood and because of that the existing numerical models use a combination of fundamental governing equations, empirical parametrizations and behaviour based expressions. Models have been developed since 30 year ago and we can split in different groups according with its complexity and scope:

- **Fully 3D Models (depth-resolving models):** In these type of models, the 3D hydrodynamics are fully described using VARANS (Volume-Averaged Reynolds-Averaged Navier-Stokes) equations and the sediment transport using semi-empirical formulations. (Jacobsen 2014, Ondina 2018). There is a good agreement between hydrodynamics and laboratory experiments (Higuera,Jacobsen), and they include most processes involved in shoaling , surf and swash zones. However, computational cost, together with the lack of knowledge and measurements in sediment dynamics at small scales, prevent the use of these models for studying long term and medium-large scale morphodynamics.

- **Quasi-3D Models (depth-averaged and phase-resolving models):** In this case, a depth-averaged (and simplified) version of Reynolds equations with non-hydrostatic terms are used. These formulations are Boussinesq-type equations or Non-Hydrostatic and Non-Lineal Shallow Water Equations (NLSWE). These equations are coupled with sediment transport formulations and bed updating. Examples of these type of codes are Delf3D () and X-Beach[1] (Roelvink 2008). These kind of models are significantly computational cheaper than fully 3D ones. They can describe longshore and crosshore processes and swash morphodynamics (Kelly and Dodd 2008, 2010). Nevertheless, they present two disadvantages for being applied in medium-large scale studies. First, the computational cost is still enough expensive to be a tool for long-term simulations. Second, since they use a combination of empirical parameters and simplified Naviers Stokes equations, in order to describe complex processes at small scales like breaking or intra-web velocities, this approach requires laboratory experiments to validate and calibrate. This data is difficult to obtain and it takes away reliability in order to use these models for general purposes.

[1] X-Beach has implemented wave resolving and non wave resolving modes

- **2DH Models (depth-averaged and phase-averaged models):** In this group the NLSWE equations are time-averaged over a wave period T. Therefore, short waves are no included in hydrodynamics. The contribution of short-waves is parametrized, and wave transformation is modelled separately. The main advantage of these kind of models is they become faster and more robust than wave-resolving ones. For this reason they present a good equilibrium between computational cost and realibity in simulating the surf-zone morphodynamics. Codes like Morfo50 (Caballeria 2000) and Morfo 55 (Garnier 2006) have been used successfully to study the mechanisms behind surf-zone patterns (...). Other codes like X-Beach are extensively used in coastal engineering. These models have presented several limitations. First of all, the swash-zone morphodynacmis are not described, and consequently the shoreline evolution is well simulated. Secondly, these models have two present two limitations to simulate crosshore processes: a) the lack three-dimensionality in hydrodynamics; b) wave asymmetry and skewness are not described. Finally, as a consequence of calculating wave transformation outside hydrodynamics, wave-current interaction is not accurately implemented. In recent years, new 2DH models have been developed including crosshore morphodynamics using empirical formulations for the wave asymmetry and skewness. For instance, X-Beach and MorfoDyn (Dubarbier 2014) have implemented this feature. Even thought it is an evolution of classical 2DH models, they do not perform properly long term morphodynamics near the shoreline.

- **One-Line models:** In these kind of models, the bi-dimensional surf-zone is collapsed into a *wave breaking line*. Total sediment transport fluxes alongshore are calculated from the wave conditions at breaking line using empirical formulations (CERC 1984). With this approach, shoreline evolution can be predicted from offshore wave conditions. Several codes of these type, as UNIBEST and GENESIS, have been largely used in coastal engineering. One special case of one-line model is Q2DMorfo (Falques 2008, Van Der Berg 2012 and Arriaga 2018). This model expands the one-line approach. First, it calculates the total sediment transport alongshore and expand it over the entire plan-form. Second, it uses a behaviour-based expression to estimate the crosshore transport and the morphological evolution in the swash-zone. Q2DMorfo has been successfully applied to study different problems related with large scale shoreline dynamics such as Shoreline Sand Waves formation and evolution (Van Der Berg, 2014) or mega-nourishment diffusion (Arriaga 2017).

- **Behaviour-based models:** In this case, the morphological response to wave forcing is described using a simple behaviour-based expression as if it was a firs-order reaction. In the last decade models with this approach has been presented to describe the global shoreline response (Yates 2009), embayed beaches shoreline rotation (Turki 2015) or crosshore/longshore shoreline evolution (Robinet 2018). The simplicity of this type of models allows an easy calibration using field data and because of that are a good tool to forecast the long term beach evolution. The main disadvantage is that its use is restricted to simulate few morphological features of a beach.

In the context of this thesis we are interested in study surf-zone morphodynamics patterns, long term shoreline evolution and the coupling of both dynamics. In order

to reach these objectives the numerical model Morfo70 has been developed. According with the previous classification, Morfo70 is a 2DH-type model with 3D capabilities. Physical processes included: wave transformation, surface rollers, crosshore-longshore sediment transport, flooding, shoreline evolution, tide and storm-surge.

2.2 General Characteristics

2.2.1 Physical system, assumptions and dynamical unknowns

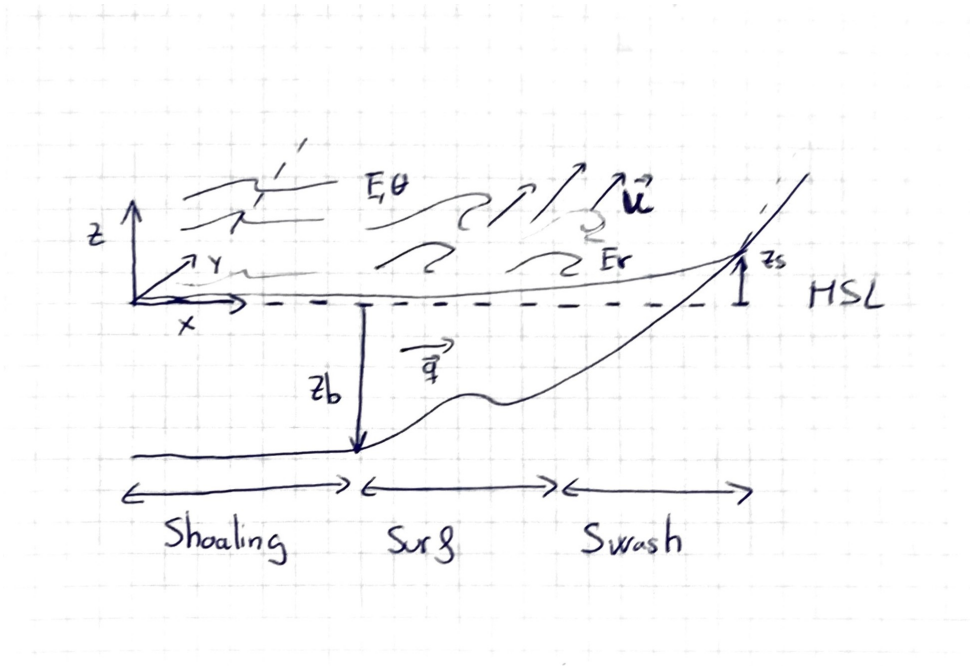

FIGURE 2.1: Physical System

The physical system simulated by Morfo70 it can be showed in 2.1. The origin of our coordinates system is set at offshore mean sea level (MSL). The instantaneous morphodynamic state of the system is described by the following variables:

- Three-dimensional velocity field: $\tilde{u}_i(x, y, z, t^*)$
- Sea level over the MSL: $\tilde{z}_s(x, y, t^*)$
- Volumetric three-dimensional sediment flux: $\tilde{q}_i(x, y, z, t^*)$
- Bottom depth from the MSL: $\tilde{z}_b(x, y, t^*)$

Where t^* is the instantaneous time and $(x, y, z) = (x_1, x_2, x_3)$ are the cross-shore, long-shore and vertical coordinates. The index i ($i = 1, 2, 3$), corresponds to the projection of a vectorial variable on the x_i axis.

As we mentioned in the previous section, we are interested in describe the long-term and large scale morphodynamical evolution. For that purpose, several simplifications are necessary which consist in the introduction of averaged versions of the above variables (Pedolsky 1987,Mei, 1989, Ribas 2004):

The fastest phenomena such as turbulence and wave orbital motion are filtered. Thus, in order to describe the system the instantaneous magnitudes can be replaced

for a version averaged over a time period T:

$$f(t) = \langle \tilde{f}(t^*) \rangle = \frac{1}{T} \int_{t-T/2}^{t+T/2} \tilde{f}(t^*) dt^* \tag{2.1}$$

Therefore, for the 2D variables, we define $z_s(x,y,t) = \langle \tilde{z_s}(x,y,t^*) \rangle$ and $z_b(x,y,t) = \langle \tilde{z_b}(x,y,t^*) \rangle$. This variables are easier to calculate than the instantaneous ones, however filtering intra-wave motion removes the main forcing of the system. In order to handle this inconvenience the wave transformation must be described. Using linear theory and monochromatic waves (Figure 2.2), the instantaneous free surface elevation can be characterize as:

$$\tilde{z_s}(x,y,t^*) = z_s + \frac{H}{2} e^{i\Phi} \tag{2.2}$$

where $H(x,y,t)$ is the wave height ($H = 2A$, A is the wave amplitude) and $\tilde{\Phi}(x,y,t^*)$ is the instantaneous wave phase ($\Phi(x,y,t) = \langle \tilde{\Phi}(x.y,t^*) \rangle$).

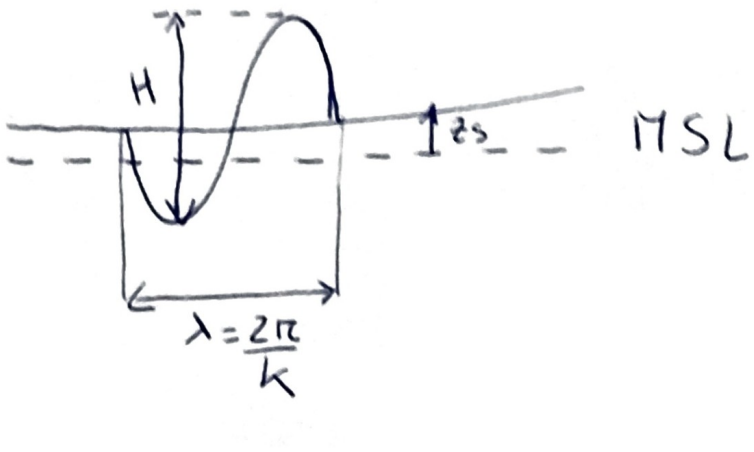

FIGURE 2.2: Averaged and instantaneous free surface

Instead of the use the wave height, H, is convenient to use the wave energy density E. Both variables are related as follows:

$$E(x,y,t) = \frac{1}{8} \rho g H^2 \tag{2.3}$$

where ρ is the water density and g is the acceleration due to gravity. The variable, H, represents a regular wave train (same amplitude for all waves), which only is possible for controlled conditions (laboratory). Real nearshore waves are irregular and wave heights are randomly distributed. In this case the wave energy density is related with the root-mean-square average of the wave height, $H_{rms}(x,y,t)$, according this equation (Longuet-Higgins 1952):

$$E(x,y,t) = \frac{1}{8} \rho g H_{rms}^2 \tag{2.4}$$

The phase, Φ is associated with more intuitive wave parameters as wave number,

$k = 2\pi/\lambda$, wave propagations direction, θ, and wave frequency, $\sigma = 2\pi/T$, for the following equations:

$$\nabla\Phi = \vec{k} = (k\cos\theta, k\sin\theta) \tag{2.5}$$

$$\frac{\partial\Phi}{\partial t} = \sigma \tag{2.6}$$

Furthermore of the wave transformation characterization, the description of the surface rollers is needed. The variable $E_r(x,y,t)$ represents the rollers energy density.

In addition to the average over the wave period, the shallow water theory is used which consider the horizontal scales much larger than the vertical scales. Taking into account these assumptions, we introduce the phase-averaged and depth-averaged version of thee three-dimensional instantaneous velocity:

$$u_i(x,y,t) = \left\langle \frac{1}{\tilde{D}} \int_{-\tilde{z}_b}^{\tilde{z}_s} \tilde{u}_i(x,y,z,t^*)dz \right\rangle \tag{2.7}$$

where $\tilde{D}(x,y,t^*) = \tilde{z}_s + \tilde{z}_b$ is the instantaneous water depth ($D = z_s + z_b$) and $i = 1,2$ since we are no taking account the vertical projection of the velocity. Using u_i instead of $\tilde{u}_i$ simplifies the calculation of the velocity field but the lost of the three-dimensionality and intra-web description take out the processes involved in the cross-shore sediment transport. In order to get a more realistic value of the velocity, estimations of the depth distribution of the horizontal velocities and the intra-web motion are required:

$$\tilde{u}_i(x,y,z,t^*) = u_i(x,y,t) + u_{d,i}(x,y,z,t) + \tilde{u}_{w,i}(x,y,z,t^*) \quad i = 1,2 \tag{2.8}$$

where $u_{d,i}$ represents the variation of the velocity with depth and $\tilde{u}_{w,i} = \frac{k_i}{k}$ is the instantaneous wave orbital motion.

For the characterization of the sediment fluxes, the shallow water assumption is applied and we define the phase-averaged and depth-integrated sediment flux:

$$q_i(x,y,t) = \left\langle \int_{-\tilde{z}_b}^{\tilde{z}_s} \tilde{q}_i(x,y,t^*)dz \right\rangle \quad i = 1,2 \tag{2.9}$$

Hereafter, both vector and i-projection notation for vectorial magnitudes will be used. For instance, for averaged sediment fluxes and velocities:

$$\vec{u} = (u_1, u_2) = (u, v)$$
$$\vec{q} = (q_1, q_2) = (q_x, q_y)$$

2.2.2 Model schema

In Figure 2.3 we can see the a sketch of the physical processes included in Morfo70 model. For the sake of simplification, it only contains crosshore dimension (x), but as we said before, Morfo70 also include the longshore dimension (y). We can analyse separately the evolution of waves, velocities, sea level and sediment transport as following:

- **Waves:** waves propagate from offshore ($E_0, \vec{k_0}$) to onshore. Depth changes induces shoaling and refraction. These processes are also affected by the currents. Inside the surf zone, the shape of the wave change rapidly and breaks.

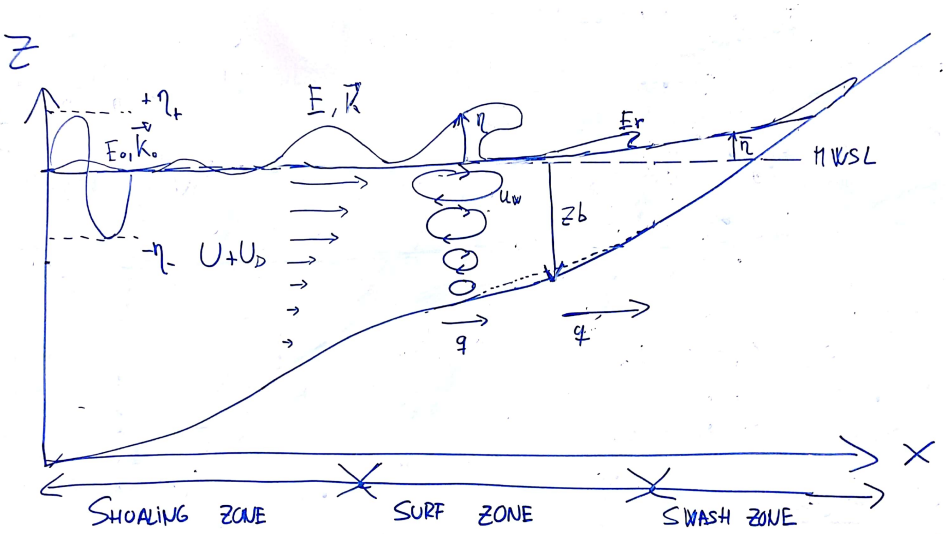

FIGURE 2.3: Simplified sketch of the physical processes related with beach morphodynamics

In this process, there is a transfer from a fraction of wave breaking energy dissipation to the generation of surface rollers.

- **Velocities:** the instantaneous velocity is the sum of two components: orbital intra-web motion, $\vec{u}_w$, and phase-averaged velocity, $\vec{U} = \vec{u} + \vec{u}_d$. The last term represents the currents and compensates long-waves, short-waves and rollers momentum transport.

- **Water level (o free surface):** there are three contributions: a) the short-wave periodic water level oscillation elevation. b) the phase-averaged elevation or set-up, z_s. This set-up is the accumulation or lowering of water due to a velocity gradient. c) Free surface perturbation due to long waves such as tides or storm surge. Offshore changes in the free surface, such as tides or storm surges, $z_{s,t}$. The near-shore propagation of these perturbations affects waves, currents and therefore the set-up.

- **Sediment transport:** waves and currents produce shear stresses at the bottom. These forces cause : a) transport the sediment near the bed layer; b) pick up the sediment and suspend it in the water column, where it will be advected. Both terms contribute to the sediment fluxes, $\vec{q}$. For mass conservation, gradients in sediment fluxes cause accretion/erosion and therefore changes in bottom elevation, z_b.

In the above analysis, swash-zone morphodynamics was not consider since Morfo70 does not include the physical processes in that region. Nevertheless, we will adopt a similar approach than Q2DMorfo. Thus, we will use a behaviour-based expression to describe the sediment transport in that zone. Lately, in the section about sediment transport module, we will detail this methodology.

Morfo70 is a temporal evolution model and for each time step solves separately waves, hydrodynamics and sediment transport. In the Figure 2.4 we see a time step execution:

(a) The wave transformation module, propagates waves from offshore to onshore and calculates surface rollers. With this data, radiation stresses and intra-web velocities computed.

(b) The hydrodynamics module, updates the velocities field, free surface elevation and water depth, using waves and rollers radiation stresses as a forcing. In this module, depth distribution of horizontal currents are estimated.

(c) The sediment transport module, uses the velocity and wave field to calculate the depth-integrated horizontal sediment fluxes. Inside this module,the bottom elevation, z_b is updated.

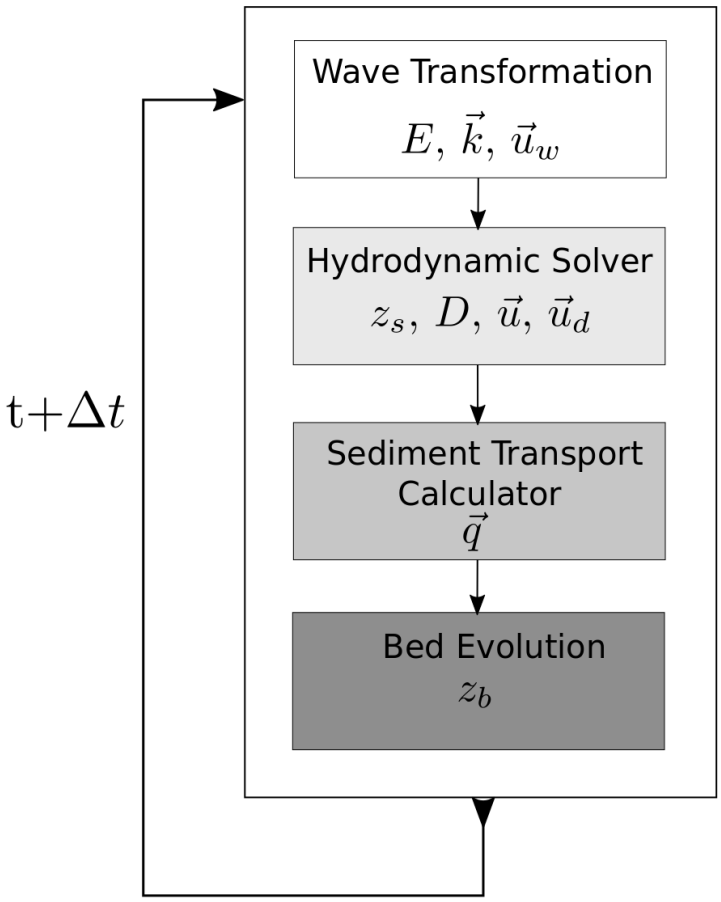

FIGURE 2.4: Scheme of a time step execution of the modules of Morfo70 model

2.2.3 Spatial Domain

Morfo70 is a 2DH model, so the differential equations are solve in horizontal bi-dimensional domain. We define the origin of the coordinate system at offshore. The **x-axis** is the cross-shore direction and **y-axis** the longshore one.

Grid

The numerical scheme used is a finite difference method. Hence, in order to discretise the spatial domain we use a rectangular grid as we can see in Figure 2.5. The

cell x-size is variable, while the y-size is constant. The number of cells of he grid is $n_x \times n_y$ and the size of the domain is $L_x \times L_y$

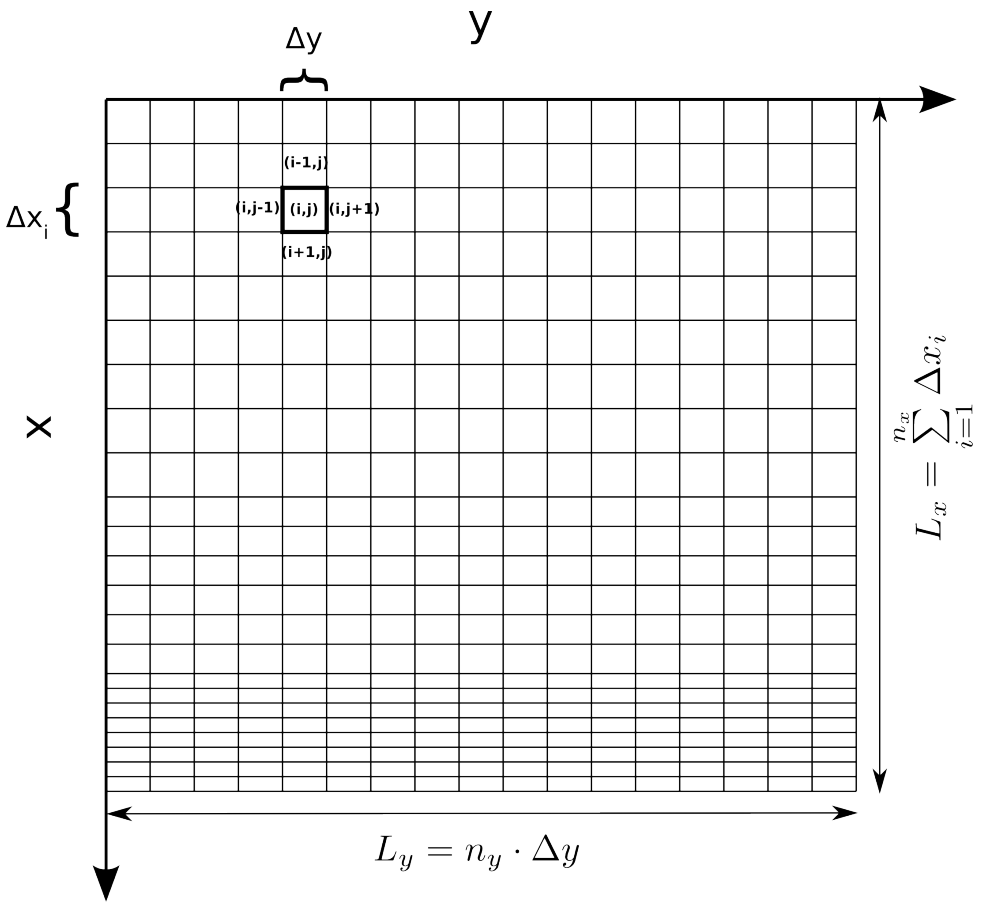

FIGURE 2.5: Morfo70 Grid

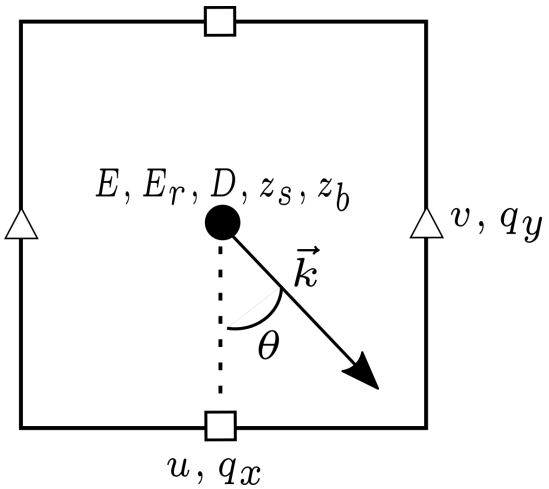

FIGURE 2.6: Morfo70 Staggered Cell

For improving model stability, we use a staggered grid. In Figure 2.6 we can see in which location of the grid-cell are defined the model variables:

- In the cell center positions, *c-nodes*, are defined wave and rollers energy and direction, E, E_r, θ; bottom elevation, set-up and water depth, z_b, z_s, D.

- In the middle of horizontal faces, *u-nodes*, are defined cross-shore velocities and sediment flux, $u = u_1$, $q_x = q_1$.
- In the center of horizontal faces, *v-nodes*, are defined long-shore velocities and sediment flux, $v = u_2$, $q_y = q_2$.

Spatial derivatives

In finite difference methods, derivatives are discretized. In Morfo70 we use a second order scheme for the spatial first derivatives and a first order scheme for the second derivatives. The expressions used in the code to approximate derivatives depends on two factors. First, the position within the cell where we want to estimate the derivative. Second, the position where is the dynamic variable defined. According with figure 2.7, the following expressions are implemented:

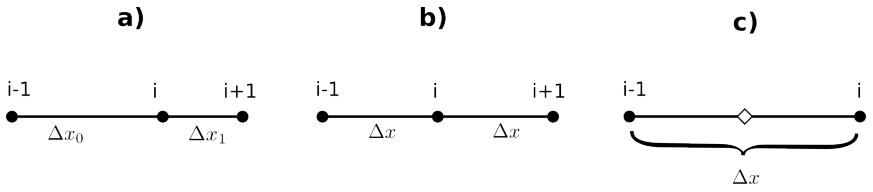

FIGURE 2.7: Spatial finite difference cases

(a) *Derivatives of node-variable at node-position with variable cell length:*

$$\frac{\partial f}{\partial x} \approx [d x^-] f^{i-1} + [d x] f + [d x^+] f^{i+1} + \mathcal{O}(\Delta x^2)$$

$$\begin{aligned}
[d x] &= \frac{\Delta x_1 - \Delta x_0}{\Delta x_0 \Delta x_1} \\
[d x^+] &= \frac{\Delta x_0}{(\Delta x_0 + \Delta x_1)\Delta x_1} \\
[d x^-] &= -\frac{\Delta x_1}{(\Delta x_0 + \Delta x_1)\Delta x_0}
\end{aligned}$$

$$\frac{\partial^2 f}{\partial x^2} \approx [d^2 x^-] f^{i-1} + [d^2 x] f + [d^2 x^+] f^{i+1} + \mathcal{O}(\Delta x)$$

$$\begin{aligned}
[d^2 x] &= -\frac{2}{\Delta x_0 \Delta x_1} \\
[d^2 x^+] &= \frac{2}{(\Delta x_0 + \Delta x_1)\Delta x_1} \\
[d^2 x^-] &= \frac{2}{(\Delta x_0 + \Delta x_1)\Delta x_0}
\end{aligned}$$

(b) *Derivatives of node-variable at node-position with constant cell length:*

$$\frac{\partial f}{\partial x} \approx \frac{f^{i+1} - f^{i-1}}{2\Delta x} + \mathcal{O}(\Delta x^2)$$

$$\frac{\partial^2 f}{\partial x^2} \approx \frac{f^{i+1} - 2f_i + f^{i-1}}{\Delta x^2} + \mathcal{O}(\Delta x)$$

(c) *First derivative*[1] *of node-variable at face-position:*

$$\frac{\partial f}{\partial x} \approx \frac{f^i - f^{i-1}}{\Delta x} + \mathcal{O}(\Delta x^2)$$

Note that expressions a) and b) are the same in the case of constant cell length, $\Delta x_0 = \Delta x_1$. For the y-direction we have the same approximations since Δy is constant.

2.2.4 Temporal Domain

Temporal numerical scheme

The wave driver is solved steadily while hydrodynamics and bed evolution are solved using multi-step methods. These schemes use a combination of second order explicit and implicit integration. Given the differential equation $\frac{\partial f}{\partial t} = g(f,t)$, the implemented methods in the code are the following:

(a) *Explicit:*

$$f^{t+\Delta t} = f^t + \frac{\Delta t}{2}\left(3g(f^t,t) - g(f^{t-\Delta t}, t-\Delta t)\right) \tag{2.10}$$

(b) *Implicit:*

$$f^{t+\Delta t} = f^t + \frac{\Delta t}{2}\left(g(f^{t+\Delta t}, t+\Delta t) + g(f^t,t)\right) \tag{2.11}$$

Time step

The time step is estimated dynamically, according with the CFL condition. For a grid cell, the time is given by:

$$\Delta t = C_t \frac{\min(\Delta x, \Delta y)}{\sqrt{u^2+v^2} + c_g}$$

where C_t and c_g are the courant number (for our implicit numerical scheme a vale between $5-10$ can be used) and the group velocity respectively. The selected time step is the maximum value for the entire grid.

2.2.5 Boundary conditions

Grid boundaries

Although, for each module of Morfo70 there are different offshore and onshore boundary conditions, the lateral boundary treatment is the same. Since we are interested in study rhythmic nearshore patterns periodic lateral conditions are implemented. Thus, for each dynamic variable of the system $f(x,y)$ the following relations must be satisfied :

$$f(x,y) = f(x, y+L_y)$$
$$\frac{\partial f(x,y)}{\partial y} = \frac{\partial f(x, y+L_y)}{\partial y}$$

where L_y is the longshore length of the domain

[1] In this case, the second derivative is not needed in Morfo70

Dry region

Morfo70 computes hydrodynamics, sediment fluxes and bed evolution of the submerged region of the beach. Hence, the domain of the model is essentially the wet region. Due to complex coast geometry and shoreline movement the description of the entire physical system including the swash zone, would require a no-steady mesh and complicated boundary conditions. In order to simplified the computation, a thin layer of water is imposed at dry cells at the beginning of the simulation. The default value of the initial water depth of those cells is $D_{dry} \sim 0.01$ m and a grid element is considered wet if its water depth is greater than D_{dry} (figure 2.8).

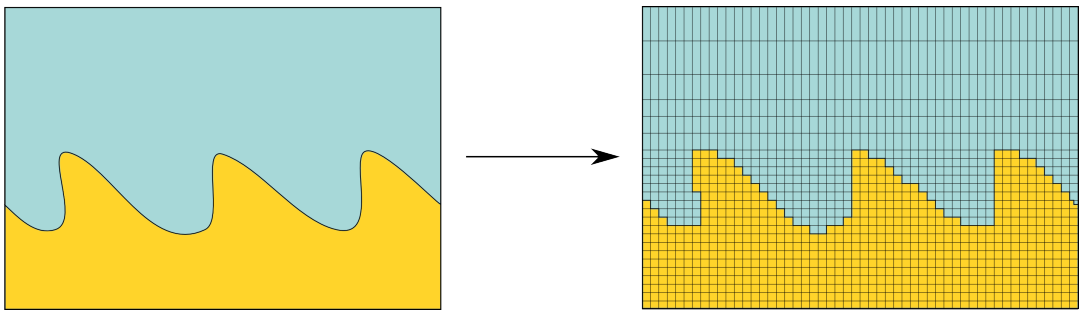

FIGURE 2.8: Representation of a dry-wet domain according to Morfo70

Using this approach all governing equation of the model can be solved for the entire domain, however each module requires a special treatment for the dry cells in order to avoid unnatural physical results.

- In the wave driver, the dry cells are computed in the same way as the wet ones. This ensure the continuity of the dynamical unknowns. Dissipation due to wave breaking and bottom friction makes that in these cells the wave energy is almost negligible.
- For the hydrodynamic solver, a dry-wet algorithm is implemented to prevent water flow from dry cells.
- In Sediment transport formulations a continuous behaviour-based parametrization allows grain movement in both submerged and emerged part of the swash region. Beyond this zone, the sediment flux in dry domain is constrained to negligible.

2.3 Wave Driver and Rollers Model

The description of the wave transformation under uneven bottoms is a complex problem which involves irregular boundary conditions and a wide range of coupled non-linear physical processes as shoaling, refraction, diffraction, wave breaking, bottom friction and wave-wind interaction. In the last decades several numerical models have been developed using different approaches for solving this problem. It is not the aim of this thesis to make a detailed review of the state of the art of this models, however is useful to show a synthesis of the different approaches taken and its advantages and limitations:

- **Wave Spectral models:** The wave spectrum energy density $E = E(x, y, \sigma, \theta)$, is solved using the action balance equation (e.g., Mei , 1983; Komen et al., 1994), which reads as following in absence of ambient currents:

$$\frac{\partial N}{\partial t} + \boldsymbol{\nabla} \cdot (\vec{c_g} N) + \frac{\partial c_\sigma N}{\partial \sigma} + \frac{\partial c_\theta N}{\partial \theta} = \frac{S_t}{\sigma} \tag{2.12}$$

  where $N = E/\sigma$ is the wave action, $\vec{c_g}$ is the wave group celerity vector and S_t represents the total source and dissipation terms. Numerical models like *SWAN* (Holthuijsen, L.H et al., 1993) solves this equation. In SWAN model, the term S_t contains wave-wind generation and energy dissipation due to wave breaking, bottom friction and white capping. This code has been widely used successfully to determine the near-shore wave transformation. The use of the wave spectrum and the wave-wind generation gives a realistic description of the wave field. The main disadvantages of this kind of of model are the computational cost (is four-dimensional problem) and the lack of wave diffraction [2].

- **Wave refraction-diffraction models:** The wave field is computed using the Mild-slope equation (Berkhoff, 1972):

$$\boldsymbol{\nabla} \cdot \left(c_g c \boldsymbol{\nabla} \eta\right) + k^2 c_g c \eta = 0 \tag{2.13}$$

  where η is the complex-valued amplitude of the free surface elevation (i.e. for linear theory $\eta = \frac{H}{2} e^{i\Phi}$) and $c = \sigma/k$ is the wave celerity. The above expression describes the monochromatic waves propagation over a bathimetry taking account the combined diffraction/refraction effects. It is an hyperbolic equation, therefore in order to solve it wave description at boundaries are need. For computing the wave transformation in beaches the parabolic approximation of 2.13 is wide used. It gives two advantages: a) The onshore boundary condition is no needed which eases the implementation of wave breaking; b) the numerical resolution is simplified. The main limitation of using the parabolic approximation is that the wave incident angle must be less than 60^o respect the bathimetry. The most known numerical model which solves the parabolic approximation of the Mild-slope equation is the REF/DIFF code (Kirby 1983b).

- **Monochromatic and non-diffraction wave propagation models:** Wave action balance equation (eq. 2.12) for monochromatic waves or Mild-slope equation (eq. 2.13) without diffraction lead to the following expression:

$$\frac{\partial E}{\partial t} + \boldsymbol{\nabla} \cdot (\vec{c_g} E) = S_t \tag{2.14}$$

  The computational resolution is faster and more robust than with the other kind of models. Since Morfo70 is designed to study sandy beaches morphodynamic patterns, we are not interested in the interaction with coastal structures such as dykes o breakwaters. Therefore, wave diffraction can be neglected. For this reason this approach is suitable to be implemented as the wave driver of Morfo70 as it will be detailed in the next section.

[2] SWAN no solves diffraction but gives an estimation

2.3.1 Governing Equations

Wave energy Balance

The wave energy density conservation including wave-current interaction for irregular waves is given by (Mei 1989):

$$\frac{\partial E}{\partial t} + \boldsymbol{\nabla} \cdot \left((\vec{u} + \vec{c}_g)E\right) + S_{ij}^{w} \frac{\partial u_i}{\partial x_j} = -\left(D_w + D_f\right) \quad , \; i,j = 1,2 \tag{2.15}$$

where D_w and D_f are the wave energy dissipation rates due to breaking and bottom friction respectively and S_{ij}^{w} are the wave radiation stresses. The wave energy density is related to the root mean squared wave height by:

$$E = \frac{1}{8}\rho g H_{rms}^2 \tag{2.16}$$

Wave Number irrationality

The wave propagation direction is given by the vectorial wave number $\vec{k}$. The relation between the wave number and the wave phase as $\vec{k} = \boldsymbol{\nabla}\Phi$ leads to the irrationality of the wave number:

$$\frac{\partial k_x}{\partial y} - \frac{\partial k_y}{\partial x} = 0 \tag{2.17}$$

Dispersion relation

From the linear wave theory, valid in the case of slow-varying current and water depth, we obtain the dispersion relation (Mei, 1989):

$$\sigma^2 = gk \tanh(kD) \tag{2.18}$$

where σ is the intrinsic frequency, that is, the frequency in a frame moving with the current $\vec{u}$. The relation between the intrinsic frequency σ and the absolute frequency $w = 2\pi/T$ is given by:

$$w = \sigma + \vec{u} \cdot \vec{k} \tag{2.19}$$

Surface rollers energy balance

The roller energy balance is given for the following equation (Svedsen 1984):

$$\frac{\partial (2E_r)}{\partial t} + \boldsymbol{\nabla} \cdot (2(\vec{u} + \vec{c})E_r) + S_{ij}^{r} \frac{\partial u_i}{\partial x_j} = -D_r + a_R D_w \quad , \; i,j = 1,2 \tag{2.20}$$

where E_r is the roller energy density, $\vec{c} = \frac{sigma}{k^2}\vec{k}$ is the wave celerity vector, S_{ij}^{r} are the roller radiation stresses, D_r is roller energy dissipation rate (transferred to turbulence) and a_R is the fraction of broken wave energy dissipation that is transferred into the surface roller.

2.3.2 Parametrizations

Group celerity

For linear wave theory the group celerity is defined by $c_g = \frac{\partial \sigma}{\partial k}$, thus:

$$c_g = \frac{c}{2}\left(1 + \frac{2kD}{\sinh(2kD)}\right) \tag{2.21}$$

Wave orbital velocity

For several Morfo70 modules is necessary to compute the module of the instantaneous wave orbital velocity $\tilde{u}_w$. Depending on the required wave description, two different parametrizations are used for calculating the magnitude of the motion:

(a) *Linear wave theory (Airy theory):*

$$\tilde{u}_w(x, y, z, t^*) = u_w \cos \Phi \tag{2.22}$$

where Φ is the wave phase and $u_w(x, y, z)$ is the amplitude of the orbital motion:

$$u_w(x, y, z, t) = \frac{H}{2}\frac{\sigma \cosh(k(z + z_b)}{\sinh(kD)} \tag{2.23}$$

for irregular waves, the equivalent expression is the root mean squared wave orbital velocity:

$$u_w(x, y, z, t) = \frac{H_{rms}}{2}\frac{\sigma \cosh(k(z + z_b))}{\sinh(kD)} \tag{2.24}$$

(b) *Assymetric and skewed wave orbital velocity:*

A nearshore realistic parametrization comes from Abreu (2010) and Ruessink (2012):

$$\tilde{u}_w(x, y, z, t^*) = u_w\sqrt{1 - r^2}\frac{\sin(\sigma t^*) + \frac{r \sin \phi}{\sqrt{1-r^2}}}{1 - r\cos(\sigma t^* + \phi)} \tag{2.25}$$

where u_w can be calculate with linear theory (2.24 and 2.23), ϕ represents the wave form and r is the non linearity. Ruessink (2012) shows in detail the methodology to estimate these parameters from the wave skewness S_u and the wave asymmetry A_u:

$$\phi = -\frac{\pi}{2} - \tan^{-1}(A_u / S_u) \tag{2.26}$$

$$r = \frac{2b}{(1 + b^2)} \tag{2.27}$$

where the parameter b is given by:

$$b^2 = \frac{2(S_u^2 + A_u^2)}{9 + 2(S_u^2 + A_u^2)} \tag{2.28}$$

The skewness and the asymmetry can be computed from the Ursell number (Ruessink 2012):

$$s_u^2 + A_u^2 = \left(p_1 + \frac{p_2 - p_1}{1 + e^{\frac{p_3 - \log(U_r)}{p4}}} \right)^2 \tag{2.29}$$

$$\tan^{-1}\left(A_u / S_u\right) = \frac{\pi}{2}\left(\tanh\left(p_5 / U_r^{p_6}\right) - 1\right) \tag{2.30}$$

where $p_1 = 0$, $p_2 = 0.857$, $p_3 = -0.471$, $p_4 = 0.297$, $p_5 = 0.815$ and $p_6 = 0.672$ are fitting parameters. The Ursell number depends on the wave characteristics and the water depth:

$$U_r = \frac{H_{rms}\lambda^2}{D^3} \tag{2.31}$$

where $\lambda = 2\pi/k$ is the wave length.

Depth-induced breaking energy dissipation rate

Different parametrizations are implemented:

(a) *Thornton and Guza 1983:*

$$D_w = \frac{3B^3 E\sigma H_{rms}^3}{4\sqrt{\pi}\gamma_b^2 D^3}\left(1 - \left(1 + \left(\frac{H_{rms}}{\gamma_b D}\right)^2\right)^{-2.5}\right) \tag{2.32}$$

where $B^3 \sim 1.0$ and $\gamma_b = 0.4 \sim 0.5$ is the expected saturated value of H_{rms}/D

(b) *Church and Thornton 1993:*

$$D_w = \frac{3B^3 E\sigma H_{rms}}{4\sqrt{\pi}D}\left(1 - \left(1 + \left(\frac{H_{rms}}{\gamma_b D}\right)^2\right)^{-2.5}\right)\left(1 + \tanh\left(8\left(\frac{H_{rms}}{\gamma_b D} - 1\right)\right)\right) \tag{2.33}$$

where $B^3 \sim 2.0$

(c) *van Leeuwen et al., 2006 for regular waves:*

$$D_w = \frac{B^3 E\sigma H}{\pi D}\left(\frac{H}{\gamma_b D}\right)^m \left(1 - \exp\left(-\left(\frac{H}{\gamma_b D}\right)^n\right)\right) \tag{2.34}$$

where $\gamma_b = 0.7 \sim 0.8$, $m = 20 \sim 50$ and $n = 20 \sim 30$

Bottom friction energy dissipation rate

According to Horikawa (1988), the dissipation due to bottom friction, D_f , is:

$$D_f = \frac{4}{3\pi}\rho c_d \frac{\pi^3 H_{rms}^3}{T^3}\frac{1}{\sinh^3 kD} \tag{2.35}$$

where c_d is the hydrodynamical drag coefficient (default value: $c_d = 0.01$)

Wave radiaton stresses

$$S_{ij}^w = E\left(\frac{c_g}{c}\frac{k_i k_j}{k^2} + \left(\frac{c_g}{c} - \frac{1}{2}\right)\delta_{ij}\right) \quad , \; i,j = 1,2 \tag{2.36}$$

Roller dissipation rate

The parameterization of the roller energy dissipation rate comes from Svedsen (1984)

$$D_r = \frac{2gE_r \sin\beta}{c} \tag{2.37}$$

where $\beta \sim 0.05$ charaterices the slope of the wave front.

Roller radiaton stresses

The radiation stresses due to roller energy propagation come

$$S^r_{ij} = 2E_r \frac{k_i k_j}{k^2} \quad , \; i,j = 1,2 \tag{2.38}$$

2.3.3 Boundary conditions

As we said before, all Morfo70 modules have implemented periodic lateral boundary conditions. However, offshore/onshore boundary description is specific for each one. Since wave and rollers governing equations (eq. 2.15 and 2.20) are first-order differential equations, they can be integrated along **x**-direction only defining offshore condition. For waves, energy (or wave height) and direction values must be imposed:

$$E\left(x'_{off}, y\right) = E_0 \tag{2.39}$$

$$\theta\left(x'_{off}, y\right) = \theta_0 \tag{2.40}$$

For rollers, energy is set equal to zero at offshore:

$$E_r\left(x'_{off}, y\right) = 0 \tag{2.41}$$

where $x'_{off} = \frac{\Delta x_1}{2}$ is the offshore x-position of the offshore wave nodes as is showed in figure 2.9, where Δx_1 is the x-length of the first-row grid cells.

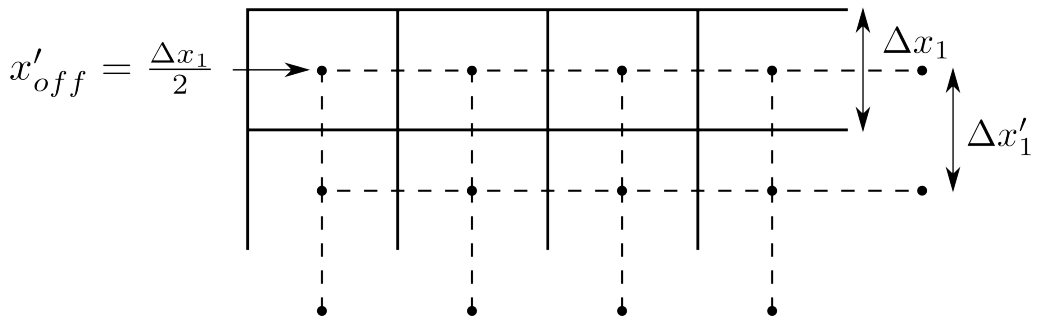

FIGURE 2.9: Offshore row of the auxiliary wave grid (dash-line) over Morfo70 grid (solid line)

The "′" symbol over a grid-variable means that this magnitude is related to the wave grid. In section 2.3.4 the use of this symbol is deprecated for the sake of simplifying the notation, since all variables are centred in wave-nodes.

2.3.4 Numerical integration

Wave and rollers field is solved integrating the governing equations over the wave grid (fig 2.9). Morfo70 assumes that waves propagate from offshore to onshore without reflection. Therefore, the code solves the corresponding expressions for each grid-row using the prior-rows data and results. The algorithm gets the incident wave to a node as a input (E_{in}, θ_{in}) and obtains the propagated wave (E, θ), such as is showed in figure 2.10. The incident wave to node (i, j) is estimated from the

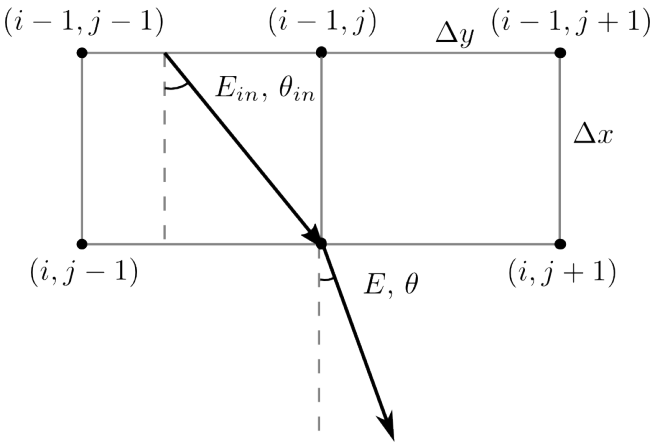

FIGURE 2.10: Wave transformation in a node

wave information from nodes $(i-1, j+1)$, $(i-1, j)$ and $(i-1, j-1)$. For this reason, the wave obliquity is limited by the ratio between Δy and Δx according to the following expression (fig. 2.11**b**):

$$\tan\theta_{max} = \frac{\Delta y}{\Delta x} \tag{2.42}$$

In order to compute simulations with higher obliquity than the constrained by the cell shape, the code creates an auxiliary row where waves can be propagated (fig. 2.11**a**). The auxiliary cell x-length Δx^* is calculated by:

$$\Delta x^* = \frac{\max(|\tan\theta_{i-1}|)}{\Delta y} \tag{2.43}$$

where θ_{i-1} is the wave angle in the upper row nodes.

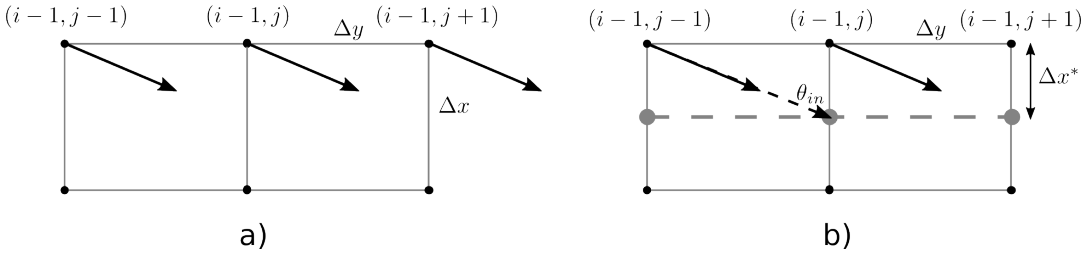

FIGURE 2.11: High angle wave incidence: a) The Δy-Δx ratio is no enough for incident waves to reach the target node; b) An intermediate grid-row (dash-line) is created

The wave driver is executed row by row. For each row the algorithm tests the

wave obliquity. If the ratio $\Delta y/\Delta x$ is not large enough, an intermediate row is created in the position given by 2.43. Water depth and velocity field is linearly interpolated to the new nodes. Once the obliquity is checked, the following steps are performed in every current-row node:

1. The wave number k is computed from equations 2.18 and 2.19.
2. The incident wave (θ_{in}, E_{in}, $E_{r,in}$) is estimated from nodes $(i-1, j-1)$ and $(i-1, j+1)$.
3. Using k and 2.17 the wave direction θ is calculated.
4. Wave energy density is solved steadily (i.e. $\frac{\partial}{\partial t} = 0$) from equation 2.15.
5. Rollers energy density is also solved steadily from equation 2.20.

Wave number

Equations 2.18 and 2.19 are combined to calculate the wave number:

$$(w - k(\vec{u} \cdot \vec{s}))^2 = gk \tanh(kD) \tag{2.44}$$

where $\vec{s}$ is the wave propagation unitary vector. Thus, the term $\vec{u} \cdot \vec{s}$ represents the projection of the currents in the wave direction. For improving the computation, non-dimensionalization of expression 2.44 is performed introducing the following dimensionless quantities:

$$\mu = kD$$
$$\nu = \frac{w^2 D}{g}$$
$$\alpha = \frac{\vec{u} \cdot \vec{s}}{\sqrt{gD}}$$

Hence, the dimensionless dispersion relation with Doppler effect is given by:

$$\nu = \left(\sqrt{\mu \tanh \mu} + \alpha\mu\right)^2 \tag{2.45}$$

In order to solve the above equation, a similar approach to the work of Ondina (to be subimtted) is taken. The method estimates explicitly an accurate initial guess of μ and follows performing an implicit refinement. For the inital explicit approximation, Ondina proposes the following expression based on Eckart (1952):

$$\mu_0 = \frac{\nu}{\sqrt{1 - \exp(-\nu f(\nu))}} \tag{2.46}$$

$$f(\nu) = 1 + \frac{a\nu + b\nu^2 + \nu^3}{c + d\nu + e\nu^2 + \nu^3}$$

where $a = 14.979441698850072$, $b = 1.1491762457673198$, $c = 89.70572188644893$, $d = -4.073909516735278$ and $e = 5.028261191874598$. Equation 2.46 does not take into account Doppler effect. In this thesis an alternative expression which includes the effect of currents is proposed:

$$\mu_0 = \frac{\nu}{\sqrt{1 - \exp(-\nu f(\nu))} + 2\alpha\sqrt{\nu}} \tag{2.47}$$

The above formula was achieved neglecting higher terms than $\mathcal{O}(\alpha)$. This assumption works since $\sqrt{gD} > 3$ ms^{-1} for depths greater than 1 m and consequently $|\alpha| < 1$.

Once μ_0 is estimated, a Newton-Rapshon method is used for the implicit refinement:

$$\mu_{n+1} = \mu_n - \frac{g(\mu_n)}{g'(\mu_n)} \tag{2.48}$$

where:

$$g(\mu_n) = \mu_n \tanh \mu_n - \left(\sqrt{\nu} - \alpha\mu_n\right)^2$$
$$g'(\mu_n) = \tanh \mu_n + \mu_n(1 - \tanh^2 \mu_n) + 2\alpha\left(\sqrt{\nu} - \alpha\mu_n\right)$$

According with Ondina, this method using 2.47 as initial value, requires a maximum of three iterations to converge up to a machine precision.

Incident wave estimation

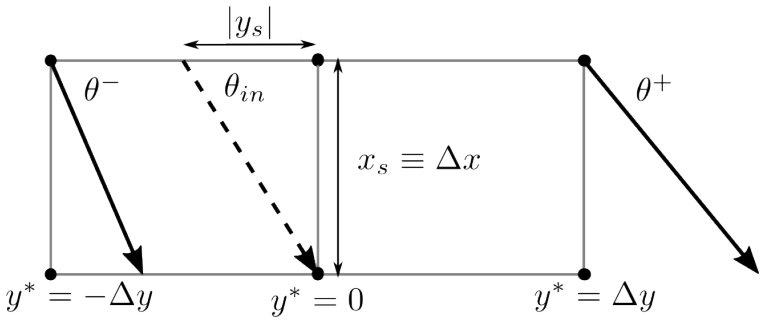

FIGURE 2.12: Incident wave (dash-line) estimation from lateral nodes

Figure 2.12 shows how the angle of the wave propagated to node (i, j) is estimated from the values of the wave angle at nodes $(i-1, j-1)$, $(i-1, j+1)$. From this figure and defining Θ as Θ=tan θ, the following relations of the incident wave ray are obtained:

$$\Theta_{in} = \frac{-y_s}{\Delta x} \tag{2.49}$$

$$\Delta s = \sqrt{\Delta x^2 + y_s^2} \tag{2.50}$$

where Δs is the length of the wave ray. Assuming that Θ varies linearly in the domain $-\Delta y \leq y^* \leq \Delta y$:

$$\Theta_{in} = \left(\frac{\Theta^+ + \Theta^-}{2}\right) + \left(\frac{\Theta^+ - \Theta^-}{2\Delta y}\right) y_s \tag{2.51}$$

and combining with 2.49, the incident wave can be explicitly described:

$$\Theta_{in} = \frac{\Theta^+ + \Theta^-}{2 + (\Theta^+ - \Theta^-)\frac{\Delta x}{\Delta y}} \tag{2.52}$$

$$\theta_{in} = \tan^{-1} \Theta_{in} \tag{2.53}$$

Once y_s is computed by $y_s = -\Theta_{in}\Delta x$, wave and rollers energy density is calculated by linear interpolation in the same way showed in 2.51.

Wave direction

The change of wave direction along propagation is governed by wave number irrationality (eq. 2.17). From this equation, we can derive an easier to compute expression which describes the angle variation along the wave ray. Given $k_x = k\cos\theta$ and $k_y = k\sin\theta$, equation 2.17 can be written as:

$$\frac{\partial k\sin\theta}{\partial y} = \frac{\partial k\cos\theta}{\partial x}$$

Performing derivatives:

$$\frac{\partial k}{\partial x}\sin\theta + k\cos\theta\frac{\partial\theta}{\partial x} = \frac{\partial k}{\partial y}\cos\theta - k\sin\theta\frac{\partial\theta}{\partial y}$$

Arranging terms:

$$\boldsymbol{\nabla}\theta\cdot\vec{s} = \frac{\boldsymbol{\nabla}k}{k}\cdot\vec{s}_\perp$$

where $\vec{s} = (\cos\theta, \sin\theta))$ is the wave propagation unitary vector and $\vec{s}_\perp = (-\sin\theta, \cos\theta)$ is the normal vector. Using the mathematical properties $\boldsymbol{\nabla}()\cdot\vec{s} = \frac{d}{d\vec{s}}()$ and $\frac{1}{f}\frac{df}{dx} = \frac{d\log f}{dx}$ we get:

$$\frac{d\theta}{d\vec{s}} = \frac{d\mathcal{K}}{d\vec{s}_\perp} \tag{2.54}$$

where $\mathcal{K} = \log k$. This formula describes how the wave propagation direction varies according transversal bottom gradients.

In order to get the angle in the target node, the above equation is discretized along the incident wave ray:

$$\frac{\theta - \theta_{in}}{\Delta s} = \left(\frac{d\mathcal{K}}{d\vec{s}_\perp}\right)_c \tag{2.55}$$

where the sub-index "c" represents the center of the wave ray. The transverse derivative can be expressed as:

$$\left(\frac{d\mathcal{K}}{d\vec{s}_\perp}\right)_c = (\boldsymbol{\nabla}\mathcal{K})_c\cdot(-\sin\theta_c, \cos\theta_c) \tag{2.56}$$

The trigonometric functions at ray-center are approximated by:

$$\sin\theta_c \approx \frac{\sin\theta + \sin\theta_{in}}{2}$$
$$\cos\theta_c \approx \frac{\cos\theta + \cos\theta_{in}}{2}$$

Since is assumed than the wave ray is not refracted more than 15° in a row, first order Taylor polynomial approximations of $\cos\theta$ and $\sin\theta$ can be used:

$$\sin\theta_c \approx \sin\theta_{in} + \cos\theta_{in}\frac{(\theta - \theta_{in})}{2}$$
$$\cos\theta_c \approx \cos\theta_{in} - \sin\theta_{in}\frac{(\theta - \theta_{in})}{2}$$

Introducing these approximations in equation 2.56:

$$\left(\frac{d\mathcal{K}}{d\vec{s}_\perp}\right)_c \approx (\nabla\mathcal{K})_c \cdot \left(\vec{s}_{\perp,in} - \vec{s}_{in}\frac{(\theta-\theta_{in})}{2}\right) \tag{2.57}$$

where $\vec{s}_{in} = (\cos\theta_{in}, \sin\theta_{in})$ and $\vec{s}_{\perp,in} = (-\sin\theta_{in}, \cos\theta_{in})$.

Combining 2.54 with 2.57 the wave angle at node (i,j) can be computed explicitly by:

$$\theta = \theta_{in} + \frac{(\nabla\mathcal{K})_c \cdot \vec{s}_{\perp,in}}{\frac{1}{\Delta s} + \frac{1}{2}(\nabla\mathcal{K})_c \cdot \vec{s}_{in}} \tag{2.58}$$

The term $(\nabla\mathcal{K})_c$ is computed using the following expressions:

$$\begin{aligned}\left(\frac{\partial\mathcal{K}}{\partial x}\right)_c &\approx \frac{1}{\Delta x}\left(\mathcal{K} - \mathcal{K}^{i-1}\right)\\ \left(\frac{\partial\mathcal{K}}{\partial y}\right)_c &\approx \frac{1}{4\Delta y}\left(\mathcal{K}^{j+1} - \mathcal{K}^{j-1} + \mathcal{K}^{i-1,j+1} - \mathcal{K}^{i-1,j-1}\right)\end{aligned} \tag{2.59}$$

where super index (i,j) are neglected for simplifying the notation, as is drawn in figure 2.13

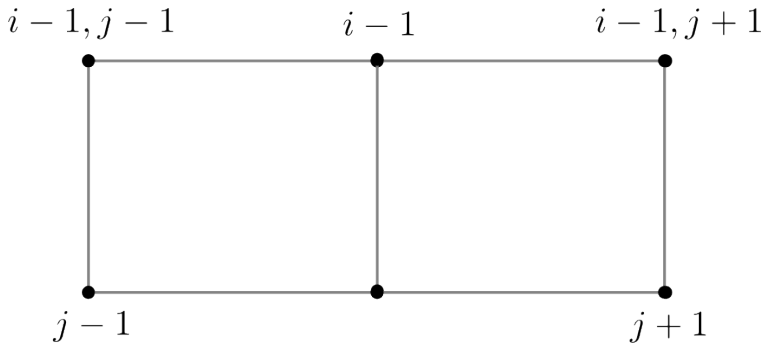

FIGURE 2.13: Sketch of simplified notation of node position

The use of 2.58 gives a fast, accurate and robust method for obtaining the wave angle. The algorithm is stable as long as the wave number is real and positive. This condition is guaranteed since in Morfo70 the water depth in the entire domain is > 0 and consequently $k > 0$.

In order to avoid excessive obliquity and propagation to offshore,the code constraints output wave angles to a $\pm\theta_{max}$ value defined by the user (usually $\theta_{max} \sim 85°$).

Wave Energy

The wave energy is computed from equation 2.15 in stationary mode, it means $\frac{\partial E}{\partial t} = 0$. Using this approach the model is more stable and there is no a significant loss of performance. Thus, the wave energy balance is given by:

$$\nabla \cdot \left(\left(\vec{u} + \vec{c_g}\right) E\right) + S^w_{ij}\frac{\partial u_i}{\partial x_j} = -\left(D_w + D_f\right) \qquad i,j = 1,2$$

Dividing all terms by E and performing the divergence of the energy flux, the above equation can be written as:

$$(\vec{u}+\vec{c}_g)\cdot\boldsymbol{\nabla}\mathcal{E}+\boldsymbol{\nabla}\cdot(\vec{u}+\vec{c}_g)+s^w_{ij}\frac{\partial u_i}{\partial x_j}=-\left(d_w+d_f\right)\qquad i,j=1,2 \tag{2.60}$$

where $\mathcal{E}=\log E$, $s^w_{ij}=S^w_{ij}/E$, $d_w=D_w/E$ and $d_f=D_f/E$. The use of $\mathcal{E}$ instead of E avoid the occurrence of negative values of the wave energy. Arranging terms of 2.60 the expression computed by Morfo70 is obtained:

$$\frac{\partial\mathcal{E}}{\partial x}=\mathcal{F} \tag{2.61}$$

where $\mathcal{F}$ is given by:

$$\mathcal{F}=-\frac{1}{u+c_{g,x}}\left(\boldsymbol{\nabla}\cdot(\vec{u}+\vec{c}_g)+s^w_{ij}\frac{\partial u_i}{\partial x_j}+d_w+d_f+(v+c_{g,y})\frac{\partial\mathcal{E}}{\partial y}\right)\qquad i,j=1,2 \tag{2.62}$$

Equation 2.61 is integrated in order to obtain the wave energy in node $(i+1,j)$ from information in rows (i,j) and $(i-1,j)$. This expression is discretized using the differential operators presented is section 2.2.3 and following the introduced notation in figure 2.14:

$$\frac{\partial\mathcal{E}}{\partial x}\approx[\not{d}x^+]\mathcal{E}^{i+1}+[\not{d}x]\mathcal{E}+[\not{d}x^-]\mathcal{E}^{i-1}$$

Since $\mathcal{F}$ in node (i,j) can be determined by straightforward direct calculation, wave energy in node $(i+1,j)$ is computed explicitly according the expression:

$$\mathcal{E}^{i+1}=\frac{1}{[\not{d}x^+]}\left(\mathcal{F}-[\not{d}x]\mathcal{E}-[\not{d}x^-]\mathcal{E}^{i-1}\right) \tag{2.63}$$

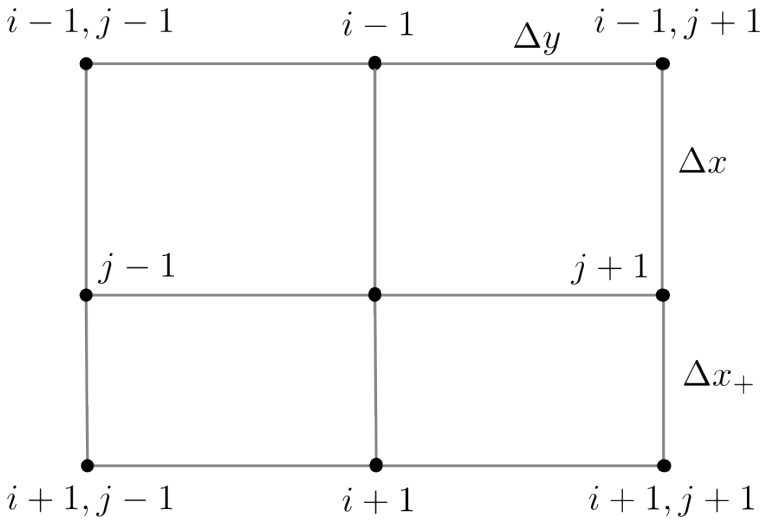

FIGURE 2.14: Sketch of simplified notation of node positions for computing $\mathcal{E}^{i+1}$

This methodology is fast and robust. Nevertheless, it presents two disadvantages: First, it only can be used from the third row (first row is set by offshore condition). Second, even-odd artefacts can appear and produce numerical noise. In order to face both issues, wave ray energy equation is computed. Solving this expression

requires iterative calculation, however, wave energy in row i can be obtained from information in rows i and $i-1$. In our case, wave ray energy computation provides a solution for the second row and from the third row is used as a corrector of 2.63 result. Wave-ray version of 2.60 can be derived from in a similar way was done for the wave direction integration:

$$c_g \frac{d\mathcal{E}}{d\vec{s}} + u\frac{\partial \mathcal{E}}{\partial x} + v\frac{\partial \mathcal{E}}{\partial y} + \boldsymbol{\nabla}\cdot(\vec{u}+\vec{c}_g) + s^w_{ij}\frac{\partial u_i}{\partial x_j} + d_{wf} = 0 \qquad i,j = 1,2 \tag{2.64}$$

where $d_{wf} = d_w + d_f$. Discretising the above equation:

$$c_{g,c}\frac{\mathcal{E}-\mathcal{E}_{in}}{\Delta s} + u_c\frac{\mathcal{E}-\mathcal{E}^{i-1}}{\Delta x} + v_c\left(\frac{\partial \mathcal{E}}{\partial y}\right)_c + \boldsymbol{\nabla}\cdot(\vec{u}+\vec{c}_g)_c + \left(s^w_{ij}\frac{\partial u_i}{\partial x_j}\right)_c + \left(d_{wf}\right)_c = 0 \tag{2.65}$$

where sub-index "c" represents the center of the wave ray. Variables are computed by interpolation. For instance, for the dissipation term:

$$\left(d_{wf}\right)_c = \frac{d_{wf,in} + d_{wf}}{2}$$

The y-derivative of $\mathcal{E}$ is approximated by:

$$\begin{aligned}
\left(\frac{\partial \mathcal{E}}{\partial y}\right)_c &\approx \frac{\mathcal{E}^{i-1} - \mathcal{E}^{i-1,j-1}}{\Delta y} && \text{for} \quad v > 0 \\
\left(\frac{\partial \mathcal{E}}{\partial y}\right)_c &\approx \frac{\mathcal{E}^{i-1,j+1} - \mathcal{E}^{i-1}}{\Delta y} && \text{for} \quad v < 0 \\
\left(\frac{\partial \mathcal{E}}{\partial y}\right)_c &\approx \frac{\mathcal{E}^{i-1,j+1} - \mathcal{E}^{i-1,j-1}}{2\Delta y} && \text{for} \quad v = 0
\end{aligned}$$

The other center-derivatives are obtained in the same way used in wave direction section 2.59.

Equation 2.65 can no longer be solved explicitly since d_{wf} depends no linearly on wave energy. A iterative Newton-Raphson is used:

$$\mathcal{E}^{n+1} = \mathcal{E}^n - \frac{\mathcal{G}(\mathcal{E}^n)}{\mathcal{G}'(\mathcal{E}^n)} \tag{2.66}$$

where:

$$\mathcal{G} = c_{g,c}\frac{\mathcal{E}-\mathcal{E}_{in}}{\Delta s} + u_c\frac{\mathcal{E}-\mathcal{E}^{i-1}}{\Delta x} + v_c\left(\frac{\partial \mathcal{E}}{\partial y}\right)_c + \boldsymbol{\nabla}\cdot(\vec{u}+\vec{c}_g)_c + \left(s^w_{ij}\frac{\partial u_i}{\partial x_j}\right)_c + \left(d_{wf}\right)_c$$

$$\mathcal{G}' = \frac{c_{g,c}}{\Delta s} + \frac{u_c}{\Delta x} + \frac{1}{2}\frac{d\left(d_{wf}\right)}{d\mathcal{E}}$$

For computing wave energy in the second grid-row, equation 2.71 is solved iteratively until the solution converges. The initial guess is the offshore condition E_0. For the correction step of $\mathcal{E}$ in the remaining rows ($i > 2$) the initial guess used is the result of 2.63, and only **one iteration** of 2.71 is executed.

Wave Rollers

The surface rollers are calculated from equation 2.20 in the same way as wave energy: the term $\frac{\partial E_r}{\partial t}$ is removed, the variable $\mathcal{E}_r = \log E_r$ is defined. The integration is performed by a predictor-corrector method from the third grid-row and a iterative calculation in the second one.

The equation solved in the predictor step is given by:

$$\frac{\partial \mathcal{E}_r}{\partial x} = \mathcal{F}_r \tag{2.67}$$

where $\mathcal{F}_r$:

$$\mathcal{F}_r = -\frac{1}{2(u+c_x)}\left(2\boldsymbol{\nabla}\cdot(\vec{u}+\vec{c}) + s^r_{ij}\frac{\partial u_i}{\partial x_j} + d_r - \alpha_R\frac{D_w}{\exp(\mathcal{E}_r)} + 2(v+c_y)\frac{\partial \mathcal{E}_r}{\partial y}\right) \qquad i,j = 1,2 \tag{2.68}$$

For the second row and for the corrector step of the other rows, the surface rollers ray equation is computed. This expression is given by:

$$2c\frac{d\mathcal{E}_r}{d\vec{s}} + 2u\frac{\partial \mathcal{E}}{\partial x} + 2v\frac{\partial \mathcal{E}}{\partial y} + 2\boldsymbol{\nabla}\cdot(\vec{u}+\vec{c}) + s^r_{ij}\frac{\partial u_i}{\partial x_j} + d_r - \frac{D_w}{\exp(\mathcal{E}_r)} = 0 \tag{2.69}$$

After discretization:

$$\begin{aligned} 0 =& 2c_c\frac{\mathcal{E}_r - \mathcal{E}_{r,in}}{\Delta s} + 2u_c\frac{\mathcal{E}_r - \mathcal{E}_r^{i-1}}{\Delta x} + 2v_c\left(\frac{\partial \mathcal{E}}{\partial y}\right)_c + 2\left(\boldsymbol{\nabla}\cdot(\vec{u}+\vec{c})\right)_c \\ &+ \left(s^r_{ij}\frac{\partial u_i}{\partial x_j}\right)_c + d_{r,c} - \frac{1}{2}D_{w,c}\left(\exp(-\mathcal{E}_r) + \exp(-\mathcal{E}_{r,in})\right) \end{aligned} \tag{2.70}$$

As was done for the wave energy, Newton-Raphson method is implemented to solve the above equation. For the second row is computed until converges using $\mathcal{E}_r = -10$ as the initial guess. From the third row, it is executed one single iteration in order to correct the predictor step. The iterative method can be expressed by:

$$\mathcal{E}_r^{n+1} = \mathcal{E}_r^n - \frac{\mathcal{G}_r(\mathcal{E}_r^n)}{\mathcal{G}'_r(\mathcal{E}_r^n)} \tag{2.71}$$

where:

$$\begin{aligned} \mathcal{G}_r =& 2c_c\frac{\mathcal{E}_r - \mathcal{E}_{r,in}}{\Delta s} + 2u_c\frac{\mathcal{E}_r - \mathcal{E}_r^{i-1}}{\Delta x} + 2v_c\left(\frac{\partial \mathcal{E}}{\partial y}\right)_c + 2\left(\boldsymbol{\nabla}\cdot(\vec{u}+\vec{c})\right)_c \\ &+ \left(s^r_{ij}\frac{\partial u_i}{\partial x_j}\right)_c + d_{r,c} - \frac{1}{2}D_{w,c}\left(\exp(-\mathcal{E}_r) + \exp(-\mathcal{E}_{r,in})\right) \\ \mathcal{G}'_r =& \frac{2c_c}{\Delta s} + \frac{2u_c}{\Delta x} + \frac{1}{2}D_{w,c}\exp(-\mathcal{E}_r) \end{aligned}$$

2.4 Hydrodynamic solver

2.4.1 Governing Equations

Instantaneous velocity decomposition

As we showed in the section related to the dynamical unknowns, the instantaneous velocity is split in depth-averaged and phase-averaged velocity, the depth-distribution over the mean horizontal velocity and the intra-web orbital motion:

$$\vec{\tilde{u}}(x,y,z,t^*) = \vec{u}(x,y,t) + \vec{u}_d(x,y,z,t) + \vec{\tilde{u}}_w(x,y,z,t^*) \tag{2.72}$$

Since $\vec{u} = \frac{1}{D}\left\langle \int_{-\tilde{z}_b}^{\tilde{z}_s} \vec{\tilde{u}} dz \right\rangle$ the following relation is obtained:

$$\int_{-z_b}^{z_s} \vec{u}_d dz = -\left\langle \int_{-\tilde{z}_b}^{\tilde{z}_s} \vec{\tilde{u}}_w dz \right\rangle \tag{2.73}$$

The right part of the equation represents the net short-wave-carried water volume flux over a wave period, that is the stokes drift. Consequently, the first term is the induced current by mass conservation, it is called undertow. Thus, $u_d(x,y,z,t)$ represents the depth-distribution of the depth-averaged velocity and the undertow.

Water mass conservation

The water mass conservation equation reads (Mei, 1989):

$$\frac{\partial D}{\partial t} + \frac{\partial Q_i}{\partial x_i} = 0 \quad , \; i = 1,2 \tag{2.74}$$

where $Q_i = Du_i$ is the i-projection of the discharge or volumetric flow rate, D is the total mean depth ($D = z_s + z_b$) as said in section 2.2.1. Assuming steady bottom topography in a hydrodynamic time step:

$$\frac{\partial z_s}{\partial t} + \frac{\partial Q_i}{\partial x_i} = 0 \quad , \; i = 1,2 \tag{2.75}$$

Water momentum balance

The phase and depth averaged momentum conservation equations are (Mei, 1989):

$$\frac{\partial Q_i}{\partial t} + \frac{\partial Q_i u_j}{\partial x_j} + gD\frac{\partial z_s}{\partial x_i} + \frac{1}{\rho}\left(S_{ij}^w + S_{ij}^r - S_{ij}^t\right) + \frac{\tau_{b,i}}{\rho} = 0 \quad , \; i,j = 1,2 \tag{2.76}$$

where S_{ij}^t are the turbulent Reynolds stresses, $\tau_{b,i}$ is the bed shear stress.

2.4.2 Parametrizations

Depth distribution of horizontal currents

A first order estimation of $\vec{u}_d(x,y,z,t)$ is given by Potrevu and Svedsen (1999) and Zhan (2003):

$$u_{d,i} = a_{1,i}(z+z_b)^2 + a_{2,i}(z+z_b) + a_{3,i} \tag{2.77}$$

$$a_{1,i} = \frac{1}{2\nu_t \rho D}\left(\frac{\partial S_{ij}^w}{\partial x_j} + \tau_{b,i}\right) - \frac{f_{w,i}}{2\nu_t} \tag{2.78}$$

$$a_{2,i} = \frac{\tau_{b,i}}{\rho \nu_t} \tag{2.79}$$

$$a_{3,i} = -\left(a_{1,i}\frac{D^2}{3} + a_{2,i}\frac{D}{2} + \frac{Q_{w,i}}{D}\right) \tag{2.80}$$

$$f_{w,i} = \frac{\partial \left\langle u_{w,i}u_{w,j}\right\rangle}{\partial x_j} + \frac{\partial \left\langle u_{w,i}w_w\right\rangle}{\partial z} \tag{2.81}$$

where $i,j = 1,2$, w_w is the z-projection of the wave orbital velocity, $Q_{w,i}$ is the Stokes drift, $f_{w,i}$ represents the vertically local contribution to the radiation stress and ν_t is the turbulent horizontal diffusivity. Realize that $z+z_b$ is equal to zero at bottom ($z=-z_b$) and equal to z_b at the MSL ($z=0$).

The stokes drift and the term $f_{w,i}$ can be estimated using linear wave theory (see appendix A):

$$Q_{w,i} = \frac{E}{\rho\sigma}\left(1 + \frac{k^2}{4}\frac{E}{\rho g}\right)k_i \tag{2.82}$$

$$f_{w,i} = \frac{\partial}{\partial x_j}\left(u_w^2 \frac{k_i k_j}{k^2}\right) \tag{2.83}$$

Wave Asymmetry and Skewness

In the above section, the three-dimensionality description of horizontal currents is estimated using linear wave theory, therefore the wave asymmetry and skewness are neglected. Since it is known that those wave characteristics are relevant in chrosshore processes, for the wave orbital motion component in equation 2.72 the expression 2.25 is used.

Shear stresses

For the bead shear stresses a 2DH extension of Feddersen (2000) is used:

$$\tau_{b,i} = \rho c_D \frac{u_w(z_0)}{\sqrt{2}}\left(\alpha_F^2 + 2\frac{|\vec{u}|^2}{u_w^2(z_0)}\right)^{1/2} u_i \tag{2.84}$$

where $u_w(z_0)$ is the root mean squared wave orbital velocity at the boundary layer edge, $z_0 \sim 0.001 \sim m$ is the bed roughness (ripple size, if they exist), α_F is constant and the default value used is $\alpha_F = 1.16$ and c_D is the bed drag coefficient. For the bed drag coefficient the following formulations have been implemented in the code:

(a) *Constant drag coefficient:*

$$c_D = c_d \tag{2.85}$$

where $c_d \sim 0.001 - 0.01$ and the default value in the code is $c_d = 0.01$

(b) *Depth varying logaritmic profile, Soulsby (1997):*

$$c_D = \left(\frac{0.4}{\ln(D/z_0) - 1} \right)^2 \tag{2.86}$$

(c) *The Manning-Strickler law:*

$$c_D = 0.015 \left(\frac{k_a}{D} \right)^{1/3} \tag{2.87}$$

where $k_a = 30 z_0$ is the apparent bed roughness.

Momentum turbulent mixing

The turbulent Reynolds stresses can be parametrized by:

$$S_{ij}^t = \rho \nu_t D \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad , \; i, j = 1, 2 \tag{2.88}$$

For the horizontal turbulent diffusivity the formulations have been implemented in the code are the following:

(a) *Constant diffusivity:*

$$\nu_t = \nu_{t,0} \tag{2.89}$$

where the default value for $\nu_{t,0}$ is $\nu_{t,0} = 0.5 \text{ m}^2/\text{s}$

(b) *Depth varying diffusivity related with wave energy dissipation:*

$$\nu_t = \nu_{t,0} + M \left(\frac{D_w}{\rho} \right) D \tag{2.90}$$

(c) *Wave height varying diffusivity related with wave energy dissipation:*

$$\nu_t = \nu_{t,0} + M \left(\frac{D_w}{\rho} \right) H_{rms} \tag{2.91}$$

where $M \sim 1$ is a parameter that characterizes the turbulence. $\nu_{t,0} \sim 0.001 - 0.0001$ represents the background diffusivity and its value is smaller than using the first parametrization.

2.4.3 Boundary conditions

Offshore boundary condition

There are two different offshore boundary conditions for the hydrodynamical model:

(a) *Offshore sponge*:

$$\frac{\partial \vec{u}}{\partial x} = \frac{\vec{u}}{L_s} \quad , \; x < x'_{off} \tag{2.92}$$

where L_s is the sponge characteristic decay length ($L_s \sim 5\Delta x$), and x'_{off} is the offshore location in the model. In Morfo70 offshore limit is defined by the first wave row , this is $x'_{off} = \Delta x/2$

(b) *Offshore long waves*. Long wave forcing as tide or storm surge is implemented in Morfo70, imposing the free surface evolution at offshore, $\left(\frac{\partial z_s}{\partial t}\right)_{x=x'_{off}} = \frac{\partial z_{s,0}}{\partial t}$.

Combining this condition with equation 2.75:

$$\frac{\partial z_{s,0}}{\partial t} = -\left(\frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y}\right)_{x=x'_{off}} \tag{2.93}$$

As it will show in the numerical implementation section, this condition the defines chrosshore flow discharge at offshore given the free surface evolution at that location, since is assumed that the long wave propagates in **x**-direction. For $Q_y(x < x'_{off}, y)$ a sponge condition is established.

For the rest of variables which need be determined for $x < x'_{off}$, such as radiation stresses, water depth or viscosity continuity is applied:

$$\left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x'_{off}} = 0 \tag{2.94}$$

Onshore boundary condition

For the dry-wet interface a flooding condition is implemented. At the beginning of the simulation, there is thin layer of water ($D_{dry} \sim 0.01$ m) in the entire domain. For each time step the flow discharge between two grid cells is allowed , if al least one of the following scenarios are met:

(a) Both cells were wet in the previous time step. A cell is considered *wet* it its water depth is greater than D_{dry}.

(b) At the previous time step, one of the grid cells was dry, the other one was wet and the hydrostatic gradient is able to flood the cell (z_s of the wet cell is higher than the dry one).

Otherwise the discharge is imposed to be equal to 0. At the end of the domain ($x = L_x$) flow obeys the non-slip condition $\vec{Q}(L_x, y) = 0$.

2.4.4 Numerical integration

Numerical Schema

Applying the trapezoidal rule (2.11) in mass and momentum conservation equations:

$$z_s^{t+\Delta t} = z_s^t - \frac{\Delta t}{2}\left[\left(\frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y}\right)^{t+\Delta t} - \left(\frac{\partial z_s}{\partial t}\right)^t\right] \tag{2.95}$$

$$Q_x^{t+\Delta t} = Q_x^t - \frac{\Delta t}{2}\left(\frac{\partial Q_x u_i}{\partial x_i} + gD\frac{\partial z_s}{\partial x} + \frac{\tau_{b,x}}{\rho} + \frac{\partial}{\partial x_i}\left(S_{xi}^w + S_{xi}^r - S_{xi}^t\right)\right)^{t+\Delta t} + \frac{\Delta t}{2}\left(\frac{\partial Q_x}{\partial t}\right)^t \quad i = 1,2 \tag{2.96}$$

$$Q_y^{t+\Delta t} = Q_y^t - \frac{\Delta t}{2}\left(\frac{\partial Q_y u_i}{\partial x_i} + gD\frac{\partial z_s}{\partial y} + \frac{\tau_{b,y}}{\rho} + \frac{\partial}{\partial x_i}\left(S_{yi}^w + S_{yi}^r - S_{yi}^t\right)\right)^{t+\Delta t} + \frac{\Delta t}{2}\left(\frac{\partial Q_y}{\partial t}\right)^t \quad i = 1,2 \tag{2.97}$$

The hydrostatic term $\left(\frac{\partial z_s}{\partial x}, \frac{\partial z_s}{\partial y}\right)^{t+\Delta t}$ in momentum equations can be derivated from 2.95:

$$\left(\frac{\partial z_s}{\partial x}\right)^{t+\Delta t} = \left(\frac{\partial z_s}{\partial x}\right)^t - \frac{\Delta t}{2}\left[\left(\frac{\partial^2 Q_x}{\partial x^2} + \frac{\partial^2 Q_y}{\partial x \partial y}\right)^{t+\Delta t} - \frac{\partial}{\partial x}\left(\frac{\partial z_s}{\partial t}\right)^t\right] \tag{2.98}$$

$$\left(\frac{\partial z_s}{\partial y}\right)^{t+\Delta t} = \left(\frac{\partial z_s}{\partial y}\right)^t - \frac{\Delta t}{2}\left[\left(\frac{\partial^2 Q_x}{\partial x \partial y} + \frac{\partial^2 Q_y}{\partial y^2}\right)^{t+\Delta t} - \frac{\partial}{\partial y}\left(\frac{\partial z_s}{\partial t}\right)^t\right] \tag{2.99}$$

Combining above equations, the number of variables of the system to solve has been reduced from z_s, Q_x, Q_y to Q_x and Q_y. non-linear-terms are approximated using explicit-predicted quantities, i.e. $\left(D\frac{\partial z_s}{\partial x}\right)^{t+\Delta t} \approx D^e\left(\frac{\partial z_s}{\partial x}\right)^{t+\Delta t}$:

$$\begin{aligned} z_s^e &= z_s^t + \frac{\Delta t}{2}\left(3\left(\frac{\partial z_s}{\partial t}\right)^t - \left(\frac{\partial z_s}{\partial t}\right)^{t-\Delta t}\right) \\ Q_x^e &= Q_x^t + \frac{\Delta t}{2}\left(3\left(\frac{\partial Q_x}{\partial t}\right)^t - \left(\frac{\partial Q_x}{\partial t}\right)^{t-\Delta t}\right) \\ Q_y^e &= Q_x^t + \frac{\Delta t}{2}\left(3\left(\frac{\partial Q_y}{\partial t}\right)^t - \left(\frac{\partial Q_y}{\partial t}\right)^{t-\Delta t}\right) \\ D^e &= z_b + z_s^e \end{aligned} \tag{2.100}$$

After discretization, the following linear system is achieved:

$$\mathbf{A} \cdot \mathbf{Q} = \mathbf{B} \tag{2.101}$$

In Morfo70 grid there are $n_x + 1$ u-nodes and $ny + 1$ v-nodes. Thus, taking account domain boundaries, $\mathbf{Q}$ is a vector of length $2 \times (n_x - 1) \times n_y$ such as:

$$\mathbf{Q} = [Q_x(1,1), Q_x(2,1), \ldots, Q_x(n_x - 1, n_y), Q_y(1,1), Q_y(2,1), \ldots, Q_y(n_x - 1, n_y)]^T$$

Therefore, in order to calculate each time step is required to solve a $(2 \times (n_x - 1) \times n_y)^2$ system, which is not computational feasible even in small domains.

In order to reduce the computational cost, a multi-step and semi-implicit method is implemented. This numerical scheme is based of the work of Leendertse (1970). The original methodology is a two-step time-staggered method: In the first step the x-momentum equation is integrated along x-direction implicitly computing y-derivatives and non-linear terms explicitly (Figure 2.15a). In the second step the

y-momentum equation is integrated implicitly along y-direction with explicit non-linear terms and x-derivatives (Figure 2.15b). Using this methodology the size of the problem changes from solving a $2 \times (n_x - 1) \times n_y$ system to solve n_y times a $(n_x - 1)$ system in the first step and n_x times a (n_y) system in the second.

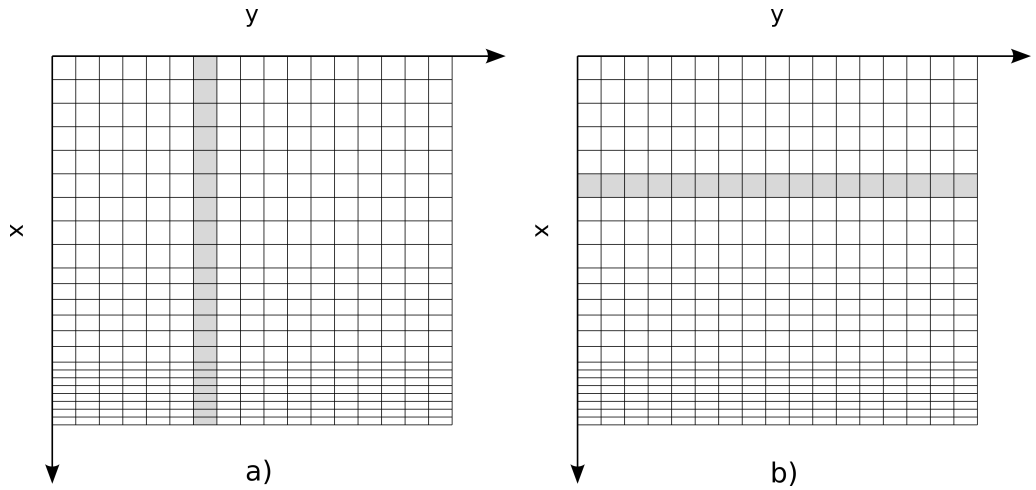

FIGURE 2.15: Numerical integration along x-direction (a), and y-direction (b)

The method presented in this thesis is a variation of Leendertse scheme, where predictor-projector sub-steps for each main step are introduced and non-staggered temporal domain is used. Morfo70 scheme provides smoother results near the shoreline than Leendertse, which result in more stability and performance under bottom changes in the dry-wet region. The algorithm is divided in seven stages: explicit predictor, Q_x predictor, Q_x projector, water depth predictor, Q_y predictor, Q_y projector and water depth projector.

For simplify the text, the following nomenclature in used for the description of the schema:

- The superscript "e" means a explicit predicted variable.
- The superscript "$*$" means a predictor unknown or a non-linear function of predictor unknowns.
- A variable without superscript means that corresponds to time $t + \Delta t$
- A variable or function from the previous time step has the superscript "t"

The integration stages are:

1) Explicit estimation: Variables Q_x^e, Q_y^e, z_s^e and D^e are estimated from expressions 2.100. These variables will be used to compute non-linear terms in the further steps.

2) x-momentum predictor: Q_x is integrated from x-momentum equation along x-axis:

$$Q_x^* = Q_x^t - \frac{\Delta t}{2}\left(\frac{\partial Q_x^* u^e}{\partial x} + \frac{\partial Q_x^t v^e}{\partial y} + gD^e\frac{\partial z_s^*}{\partial x} + \frac{\tau_{b,x}^*}{\rho} + \frac{\partial}{\partial x_i}(S_{xi}^w + S_{xi}^r - S_{xi}^{t*})\right) + \frac{\Delta t}{2}\left(\frac{\partial Q_x}{\partial t}\right)^t \quad i = 1,2 \tag{2.102}$$

where:

$$\frac{\partial z_s^*}{\partial x} = \frac{\partial z_s^t}{\partial x} - \frac{\Delta t}{2}\left[\frac{\partial^2 Q_x^*}{\partial x^2} + \frac{\partial^2 Q_y^e}{\partial x \partial y} - \frac{\partial}{\partial x}\left(\frac{\partial z_s}{\partial t}\right)^t\right]$$
$$\tau_{b,x}^* = \rho c_D \frac{u_w(z_0)}{\sqrt{2}}\left(\alpha_F^2 + 2\frac{(|\vec{u}^e|^2)}{u_w^2(z_0)}\right)^{1/2}\frac{Q_x^*}{D^e}$$
$$\frac{\partial S_{xi}^{t*}}{\partial x_i} = \frac{\partial}{\partial x}\left(2\rho\nu_t D^e \frac{\partial}{\partial x}\left(\frac{Q_x^*}{D^e}\right)\right) + \frac{\partial}{\partial y}\left(\rho\nu_t D^e\left(\frac{\partial u^e}{\partial y} + \frac{\partial v^e}{\partial x}\right)\right)$$

Viscosity and drag coefficient are computed using the water depth in the previous time step:

$$\nu_t = \nu_t(D^t)$$
$$c_D = c_D(D^t)$$

3) x-momentum projector: Q_x is integrated from x-momentum equation along y-axis:

$$Q_x = Q_x^t - \frac{\Delta t}{2}\left(\frac{\partial Q_x^* u^e}{\partial x} + \frac{\partial Q_x v^e}{\partial y} + gD^e\frac{\partial z_s^*}{\partial x} + \frac{\tau_{b,x}}{\rho} + \frac{\partial}{\partial x_i}(S_{xi}^w + S_{xi}^r - S_{xi}^t)\right) + \frac{\Delta t}{2}\left(\frac{\partial Q_x}{\partial t}\right)^t \quad i = 1,2 \tag{2.103}$$

where:

$$\tau_{b,x} = \rho c_D \frac{u_w(z_0)}{\sqrt{2}}\left(\alpha_F^2 + 2\frac{|\vec{u}^e|^2}{u_w^2(z_0)}\right)^{1/2}\frac{Q_x}{D^e}$$
$$\frac{\partial S_{xi}^t}{\partial x_i} = \frac{\partial}{\partial x}\left(2\rho\nu_t D^e \frac{\partial}{\partial x}\left(\frac{Q_x^*}{D^e}\right)\right) + \frac{\partial}{\partial y}\left(\rho\nu_t D^e\left(\frac{\partial}{\partial y}\left(\frac{Q_x}{D^e}\right) + \frac{\partial v^e}{\partial x}\right)\right)$$

4) Water depth predictor: In this step the free surface and water depth are updated:

$$z_s^* = z_s^t - \frac{\Delta t}{2}\left[\frac{\partial Q_x^*}{\partial x} + \frac{\partial Q_y^t}{\partial y} - \left(\frac{\partial z_s}{\partial t}\right)^t\right]$$
$$D^* = z_b + z_s^*$$
$$u = Q_x/D^*$$

5) y-momentum predictor: Q_y is integrated from y-momentum equation along y-axis:

$$Q_y^* = Q_y^t - \frac{\Delta t}{2}\left(\frac{\partial Q_y^e u}{\partial x} + \frac{\partial Q_y^* v^e}{\partial y} + gD^*\frac{\partial z_s^*}{\partial y} + \frac{\tau_{b,y}^*}{\rho} + \frac{\partial}{\partial x_i}(S_{yi}^w + S_{yi}^r - S_{yi}^{t*})\right) + \frac{\Delta t}{2}\left(\frac{\partial Q_y}{\partial t}\right)^t \quad i = 1,2 \tag{2.104}$$

where:

$$\frac{\partial z_s^*}{\partial y} = \frac{\partial z_s^t}{\partial x} - \frac{\Delta t}{2}\left[\frac{\partial^2 Q_y^e}{\partial x \partial y} + \frac{\partial^2 Q_y^*}{\partial y^2} - \frac{\partial}{\partial y}\left(\frac{\partial z_s}{\partial t}\right)^t\right]$$
$$\tau_{b,y}^* = \rho c_D^* \frac{u_w(z_0)}{\sqrt{2}}\left(\alpha_F^2 + 2\frac{u^2 + (v^e)^2}{u_w^2(z_0)}\right)^{1/2}\frac{Q_y^*}{D^*}$$
$$\frac{\partial S_{yi}^{t*}}{\partial x_i} = \frac{\partial}{\partial y}\left(2\rho\nu_t^* D^*\frac{\partial}{\partial y}\left(\frac{Q_y^*}{D^*}\right)\right) + \frac{\partial}{\partial x}\left(\rho\nu_t^* D^*\left(\frac{\partial u}{\partial y} + \frac{\partial v^e}{\partial x}\right)\right)$$

for viscosity and drag:

$$\nu_t^* = \nu_t(D^*)$$
$$c_D^* = c_D(D^*)$$

6) **y-momentum projector:** Q_y is integrated from y-momentum equation along x-axis:

$$Q_y = Q_y^t - \frac{\Delta t}{2}\left(\frac{\partial Q_y u}{\partial x} + \frac{\partial Q_y^* v^e}{\partial y} + gD^*\frac{\partial z_s^*}{\partial y} + \frac{\tau_{b,y}}{\rho} + \frac{\partial}{\partial x_i}(S_{yi}^w + S_{yi}^r - S_{yi}^t)\right)$$
$$+ \frac{\Delta t}{2}\left(\frac{\partial Q_y}{\partial t}\right)^t \quad i = 1,2 \tag{2.105}$$

where:

$$\tau_{b,y} = \rho c_D^* \frac{u_w(z_0)}{\sqrt{2}}\left(\alpha_F^2 + 2\frac{u^2 + (v^e)^2}{u_w^2(z_0)}\right)^{1/2}\frac{Q_y}{D^*}$$
$$\frac{\partial S_{yi}^t}{\partial x_i} = \frac{\partial}{\partial y}\left(2\rho\nu_t^* D^*\frac{\partial}{\partial y}\left(\frac{Q_y^*}{D^*}\right)\right) + \frac{\partial}{\partial x}\left(\rho\nu_t^* D^*\left(\frac{\partial u}{\partial y} + \frac{\partial}{\partial x}\left(\frac{Q_y}{D^*}\right)\right)\right)$$

7) **Water depth projector:** Final values of z_s, D and v are computed:

$$z_s = z_s^t - \frac{\Delta t}{2}\left[\left(\frac{\partial Q_x}{\partial x} + \frac{\partial Q_y}{\partial y}\right) - \left(\frac{\partial z_s}{\partial t}\right)^t\right]$$
$$D = z_b + z_s$$
$$v = Q_y / D$$

In order to avoid even-odd artefacts, for each time step the execution order of Q_x (**2** and **3**) and Q_y (**5** and**6**) stages is switched.

Linear system and boundary conditions

Computation of stages **1**, **4** and **7** are straightforward, while calculation of Q_x and Q_y requires solving linear systems. In this section the resolution of these problems is sketched. The detailed description of linear system coefficients is out of scope of the main aim of this thesis due to the complexity of equations 2.102, 2.103, 2.104 and 2.105. However, in Sediment transport governing equation are simpler and discretization is full-described. Since both modules share numerical schema, we consider that the methodology of obtaining the system coefficients is explained.

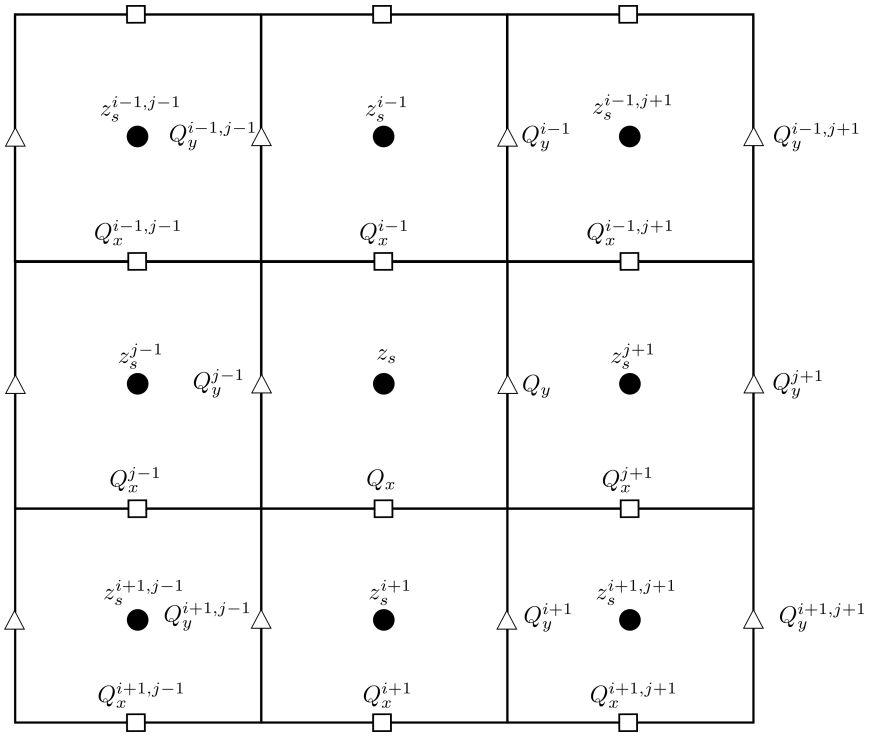

FIGURE 2.16: Sub-domain required to discretizate predictor/projector equations of Qx and Q_y in node (i, j). Super-index of column j and row i are neglected, i.e. $i+1$, j=$i+1$

- Q_x **calculation**:

  According with notation introduced in figure 2.16, for the predictor step discretization of equation 2.102 in node i, j (figure) leads to the following linear system:

$$\mathcal{L}_x^* Q_x^{*,i-1} + d_x^* Q_x^* + u_x^* Q_x^{*,i+1} + = b_x^* \quad 1 \leq i < n_x \tag{2.106}$$

  In a grid-column j there are $n_x + 1$ u-nodes (so $n_x + 1$ unknowns) and $n_x - 1$ equations as 2.106. From offshore and onshore boundary conditions the two additional relations required are obtained. From offshore B.C:

  (a) *Offshore sponge:*

$$Q_x^{*,0} = \frac{Q_x^{*,1}}{D^{t,0}} \exp\left(-\frac{\Delta x^1}{L_s}\right) \tag{2.107}$$

  (b) *Offshore long wave (tide):*

$$Q_x^{*,0} = Q_x^{*,1} + \Delta x^1 \frac{\partial z_{s,0}}{\partial t} \tag{2.108}$$

  From onshore B.C:

$$Q_x^{*,n_x} = 0 \tag{2.109}$$

  In order to impose no-flow in dry nodes, system coefficients in $\mathbf{i_{dry}} = \{\}$ index set must be adequate to force $Q_x^{*,i} = 0 \quad , \; i \in \mathbf{i_{dry}}$. The indexes of $\mathbf{i_{dry}}$ are

determined by conditions described in section 2.4.3:

$$i \in \mathbf{i_{dry}} \quad \text{if} \begin{cases} D^i \leq D_{dry} & \text{AND} \quad D^{i+1} \leq D_{dry} \\ & \text{OR} \\ D^i \leq D_{dry} & \text{AND} \quad z_s^{i+1} \leq z_s^i \\ & \text{OR} \\ D^{i+1} \leq D_{dry} & \text{AND} \quad z_s^i \leq z_s^{i+1} \end{cases} \tag{2.110}$$

After applying boundary and dry-conditions, the following tridiagonal system is obtained:

$$\begin{bmatrix} d^{*,1} & u^{*,1} & & & 0 \\ l^{*,2} & d^{*,2} & u^{*,1} & & \\ & l^{*,3} & d^{*,3} & \ddots & \\ & & \ddots & \ddots & u^{*,n_x-2} \\ 0 & & & l^{*,n_x-1} & d^{*,n_x-1} \end{bmatrix} \begin{bmatrix} Q_x^{*,1} \\ Q_x^{*,2} \\ Q_x^{*,3} \\ \vdots \\ Q_x^{*,n_x-1} \end{bmatrix} = \begin{bmatrix} b_x^{*,1} \\ b_x^{*,2} \\ b_x^{*,3} \\ \vdots \\ b_x^{*,n_x-1} \end{bmatrix} \tag{2.111}$$

$$\mathbf{A}_x^* \cdot \mathbf{Q}_x^* = \mathbf{B}_x^*$$

The above system can be solved using a simplified form of Gaussian elimination, known as Thomas algorithm (see Appendix **??**), which is stable when matrix **A** is diagonal dominant o symmetric positive definite (Niyogi 2006). In our case, we can ensure this condition if the time step is at most of the same order than the cell size ($\frac{1}{\Delta t} \geq \frac{1}{\Delta x}$). This algorithm can solve the system in $\mathcal{O}(n)$ operations instead of $\mathcal{O}(n^3)$ required by Gaussian elimination.

For the projector step, equation 2.103 is discretized along x-direction. For each grid-row i such $1 \leq i \leq n_x$ ($i = n_x$ is constraint by the no-slip condition) , n_y linear equations are obtained:

$$l_x Q_x^{j-1} + d_x Q_x + u_x Q_x^{j+1} + = b_x \quad 1 \leq j \leq n_y \tag{2.112}$$

In the same, the periodic lateral B.C. give us the missing equations needed:

$$Q_x^0 = Q_x^{n_y}$$
$$Q_x^{n_y+1} = Q_x^1$$

For the dry u-nodes, coefficients of the system are corrected to constrain $Q_x^i = 0$ in $\mathbf{j_{dry}}$ positions. The dry-index set is build collecting in each column the

indexes of the $\mathbf{i_{dry}}$ obtained before. Writing the system in matrix notation:

$$\begin{bmatrix} d_1 & u_1 & & & l_1 \\ l_2 & d_2 & u_2 & & \\ & l_3 & d_3 & \ddots & \\ & & \ddots & \ddots & u_{n_y-1} \\ u_{n_y} & & & l_{n_y} & d_{n_y} \end{bmatrix} \begin{bmatrix} Q_x \\ Q_x \\ Q_x \\ \vdots \\ Q_{x.n_y} \end{bmatrix} = \begin{bmatrix} Q_x^1 \\ Q_x^2 \\ Q_x^3 \\ \vdots \\ Q_x^{n_y} \end{bmatrix} = \begin{bmatrix} b_x^1 \\ b_x^2 \\ b_x^3 \\ \vdots \\ b_x^{n_y} \end{bmatrix} \tag{2.113}$$

$$\mathbf{A}_x \cdot \mathbf{Q}_x = \mathbf{B}_x$$

In this case, periodic lateral boundary conditions leads to a periodic tridiagonal matrix. This type of systems are usually solved using iterative methods based in Thomas algorithm. This approach is not as accurate and stable as the tridiagonal algorithm is. However, El-Mikkawy (2005) presented a robust, fast and accurate method to solve this type of systems (see Appendix **??**). The algorithm complexity is $\mathcal{O}(n)$ and the condition for stability is the non singularity of matrix $\mathbf{A}$, which is met by our problem.

- Q_y **calculation**:

  The resolution of Q_y is analogous to explained about Q_x. In the predictor step Q_y^* is integrated along y-direction: $n_x - 1$ periodic tridiagonal systems $(n_y \times n_y)$ are computed. Dry v-nodes set is build in the same as was explained in Q_x section and periodic lateral boundary conditions are imposed:

$$Q_y^{*,0} = Q_y^{*,n_y}$$
$$Q_y^{*,n_y+1} = Q_y^{*,1}$$

  For the last grid-row ($i = n_x$), Q_y^* is estimated from the non-slip condition:

$$Q_y^*(x = L_x) = 0$$

  Therefore, the value of last-row v-nodes is inferred by quadratic interpolation:

$$\begin{aligned} Q_y^{*,j,n_x} &= Q_y^* \left(x = L_x - \frac{\Delta x^{n_x}}{2} \right) \\ &= \frac{1}{15} \left(10 Q_y^{*,j,n_x-1} - 3 Q_y^{*,j,n_x-2} \right) \end{aligned}$$

  For the projector step Q_y is determined integrating along x-direction: n_y tridiagonal systems $(n_x \times n_x)$ are solved. Offshore sponge and onshore no-slip condition are imposed:

$$Q_y^0 = \frac{Q_y^1}{D^{*,0}} \exp\left(-\frac{\Delta x^1}{L_s} \right) \tag{2.114}$$
$$Q_y^{n_x+1} = 0 \tag{2.115}$$

2.5 Sediment transport and bed evolution

2.5.1 Governing equations

The bed evolution can be described by the sediment volume conservation:

$$\frac{\partial z_b}{\partial t} - \frac{1}{1-p}\nabla \cdot \vec{q} = 0 \tag{2.116}$$

where p (~ 0.4) is the seabed porosity. It is more appropriate to use the bottom perturbation $h(x,y,t)$ over the initial alongshore uniform profile $z_{b,0}(x)$ in stead of using the bed elevation $z_b(x,y,t)$ (equation 2.117 and figure 2.17).

$$z_b(x,y,t) = z_{b,0}(x) - h(x,y,t) \tag{2.117}$$

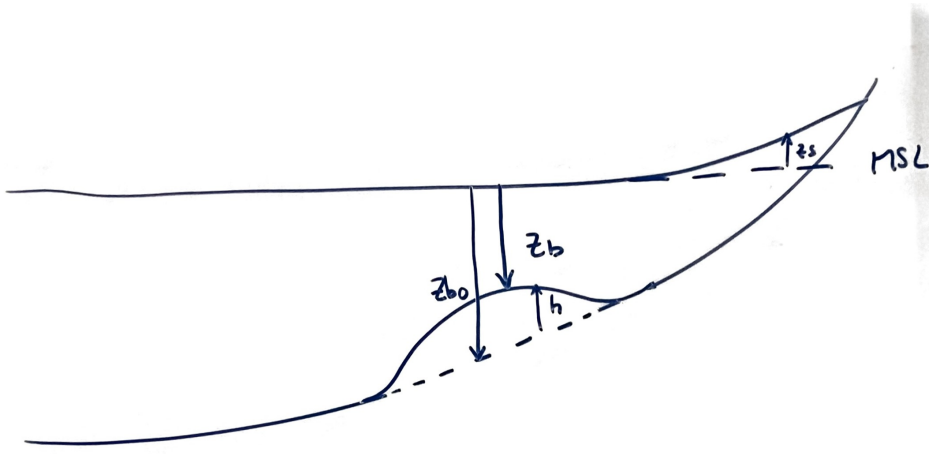

FIGURE 2.17: Initial alongshore uniform profile and bottom perturbation

Since $\frac{\partial z_{b,0}}{\partial t} = 0$ and combining 2.117 with 2.116, we obtain:

$$\frac{\partial h}{\partial t} + \frac{1}{1-p}\nabla \cdot \vec{q} = 0 \tag{2.118}$$

2.5.2 Sediment transport formulations

Convective formulation without cross-shore transport

The total sediment flux ($\vec{q} = (q_x, q_y)$) is split in advective and diffusive contributions.

$$\vec{q} = \vec{q}_{adv} + \vec{q}_{dif} \tag{2.119}$$

- **Advection:** The advective term describes the sand grains transport due to bulk motion. Thus, the depth-integrated advected volumetric sediment flux can be characterized by:

$$\vec{q}_{adv} = \alpha \vec{u} \tag{2.120}$$

where α is the *potential wave stirring* and represents the depth-integrated volumetric sediment concentration. In order to estimate α, different formulations are implemented in Morfo70:

(a) *Constant Wave Stirring (CWS)*
Based on the work of Garnier (2006), the chosen value for constant wave stirring is:

$$\alpha \sim 0.001 \text{ m} \tag{2.121}$$

(b) *Stirring according Soulsby and Van Rijn (SVR)*
This parametrization is based on the Soulsby and Van Rijn formula (Soulsby, 1997), and reads:

$$\begin{aligned} \alpha &= A_S \left[u_{str} - u_{crit}\right]^{2.4} && \text{if } u_{str} > u_{crit} \\ \alpha &= 0 && \text{otherwise} \end{aligned} \tag{2.122}$$

where the parameter $A_S = A_{SS} + A_{SB}$ represents the dependence of the sediment properties and water depth on suspended load transport (A_{SS}) and bedload transport (A_{SB}) (Soulsby, 1997):

$$\begin{aligned} A_{SS} &= \frac{0.012 d_{50} D_*^{-0.6}}{\left[(s-1) g d_{50}\right]^{1.2}} \\ A_{SB} &= \frac{0.005 D (d_{50}/D)^{1.2}}{\left[(s-1) g d_{50}\right]^{1.2}} \end{aligned} \tag{2.123}$$

where d_{50} is the median grain size ($d_{50} \sim 0.25$ mm), s is the relative density of density ($s \sim 2.65$), and D_* is the dimensionless grain size:

$$D_* = \left[\frac{g(s-1)}{\nu^2}\right]^{1/3} d_{50} \tag{2.124}$$

and ν is the kinematic viscosity of water ($\nu = 1.3 \times 10^{-6}$ m^{2}s^{-2})
The stirring velocity u_{str}, depends on the currents, the wave orbital velocity at boundary layer edge and the bottom drag (Soulsby 1997):

$$u_{str} = \left(u^2 + \frac{0.018}{c_D} u_w^2(z_0)\right)^{1/2} \tag{2.125}$$

For the bed drag coefficient c_D, Soulsby (1997) propose to use the logarithmic depth varying formula (2.86) with a roughness length value $z_0 = 0.006$ m. Realize that the morphological drag coefficient can differ from the hydrodynamical one as is suggested by Souslby (1997)
The threshold velocity for sediment transport u_{crit} depends on sediment characteristics and water depth (Soulsby, 1997)

$$u_{crit} = \Lambda \log_{10} \frac{D}{d_{50}} \tag{2.126}$$

where Λ is:

$$\Lambda = \begin{cases} 0.19 (d_{50})^{0.1} & \text{for} \quad 0.00001 \text{ m} \leq d_{50} \leq 0.00005 \text{ m} \\ 8.5 (d_{50})^{0.6} & \text{for} \quad 0.00005 \text{ m} \leq d_{50} \leq 0.002 \text{ m} \end{cases} \tag{2.127}$$

(c) *Stirring according Reniers*

In this formulation the contribution of the turbulent eddies created after the roller dissipation is added to the sedimenten stirring:

$$u_{str} = \left(u^2 + \frac{0.018}{c_D} u^2_{w,rms}(z_0) + n_{bor} u^2_{bor} \right)^{1/2} \tag{2.128}$$

Following the arguments Roelvink et al. (1989) and Stive et al. (1994), the parameter u_{bor} is modelled in the following way:

$$u^2_{bor} = \left(e^{(D/H_{rms})} - 1 \right)^{-m_{bor}} \left(\frac{D_r}{\rho} \right)^{2/3} \tag{2.129}$$

where m_{bor} and n_{bor} are parameters.

- **Diffusion:** The diffusive term represents the net sediment movement driven by the slope. In this case there are two contributions to the diffusive sediment transport:

$$\vec{q}_{dif} = \vec{q}_{dslp} + \vec{q}_{swash} \tag{2.130}$$

  – *Gravitational downslope transport*

  This contribution describes the sediment flux from a region of higher bottom elevation to a region of lower bottom elevation. Based on the Souslby Van Rijn formula (Sousby 1997) and the work of Bagnold (1963), the bed-slope transport is described by:

$$\vec{q}_{dslp} = -\alpha \gamma_{dslp} \frac{u_{dslp}}{|\tan \phi_c - |\nabla z_b||} \nabla h \tag{2.131}$$

  where γ_{dslp} is a coefficient , u_{dslp} represents the velocity which forces the random movement of the grains that induces the diffusion and ϕ_c is the friction angle $\tan \phi_c = 0.63$. In this formulation the crosshore processes are excluded. For this reason the slope which drives the transport is ∇h in stead of ∇z_b, since a equilibrium between the gravitational downslope transport and the chossore transport is assumed for the inital profile $z_{b,0}$. Several expressions for u_{dslp} are implemented in the code:

  (a) *Constant velocity*

$$u_{dslp} = u_0 \tag{2.132}$$

  where $u_0 \sim 1$.

  (b) *Currents*

$$u_{dslp} = |\vec{u}| \tag{2.133}$$

  (c) *Wave orbital velocity at boundary layer*

$$u_{dslp} = u_w(z_0) \tag{2.134}$$

  (d) *Shear stress velocity*

$$u_{dslp} = \left(u^2 + \frac{0.018}{c_D} u^2_w(z_0) \right)^{1/2} \tag{2.135}$$

  – *Swash zone transport* The default value of γ_{dslp} depends on the u_{dslp} formula. For (a) and (b) $\gamma_{dslp} \sim 1$, for (b) and (c) $\gamma_{dslp} \sim 0.5$.

The Swash transport is not a diffusive transport in the sense that is not a sediment flux due to grains net movement from high concentration to lower concentration. Nevertheless, it is a diffusive transport according with the definition given above: it is modelled using the bottom slope as driver. In absence of run-up description the swash transport model follows a behaviour-based formula. This expression forces sediment movement in swash region in order to tend to a equilibrium slope.

Therefore, for this parametrization the swash transport is modelled as follows:

$$\vec{q}_{swash} = -\gamma_{swash}\Gamma\frac{\max(D_w)}{\rho(s-1)g}\boldsymbol{\nabla} h \tag{2.136}$$

where $\gamma_{swash} \sim 0.5$ is a coefficient and $\max(D_w)$ is the maximum value of the wave breaking dissipation along shoreline-normal. In order to simplify calculations, shoreline parallel to y-axis is assumed to obtain $\max(D_w)$. This formula relates the loss energy in surf-zone with the intensity of the swash transport and varies with the water depth according with a shape function Γ:

$$\Gamma = \frac{1}{2}\tanh\left(\pi\left(2\frac{|\mathcal{D}| - D_{dry}}{A_{swash} - D_{dry}} - 1\right)\right) + \frac{1}{2} \tag{2.137}$$

where $A_{swash} \sim 0.3$ m is the amplitude of the run-up ,$D_{dry} \sim 0.01$ m determines the shoreline depth and $\mathcal{D}$ is the *effective depth* which is positive in the wet region and negative otherwise:

$$\begin{aligned} \mathcal{D} &= D && \text{for} \quad D > D_{dry} \\ \mathcal{D} &= z_b + z_{s,s} && \text{for} \quad D \leq D_{dry} \end{aligned}$$

where $z_{s,s}$ is the water elevation in the nearest shoreline cell. The value of A_{swash} can be defined by the user or estimated from wave characteristics at offshore (Nielsen 1991, Gomes da Silva 2018). According with Gomes da Silva (2018)

$$A_{swash} = 0.26\left(\frac{H_\infty \lambda_\infty}{\beta_s}\right) \tag{2.138}$$

where H_∞ and λ_∞ are the significant wave height and wave length at deep water ($D = 80$ m); β_s is the slope of the profile at swash zone.

Convective formulation transport with cross-shore transport

This parametrization is based on the Q2DMorfo model (Falques 2008, Van Der Berg 2012 and Arriaga 2018). In this formulation combine the advective model presented before with simplified version of the crosshore and diffusive transport of Q2DMorfo. Both contributions are slope-driven, so following the nomenclature of this section, both are considered as diffusive transport:

$$\vec{q}_{dif} = \gamma\left(\frac{\partial z_b}{\partial x} - \beta_e, \frac{\partial z_b}{\partial y}\right) \tag{2.139}$$

The x-component drives the bathymetry to a crosshore equilibrium profile β_e defined by the user. The y-component is the alongshore gravitational downslope transport.

The factor γ follows the equation:

$$\gamma = \gamma_0 \gamma_b^{-1/6} H_b(y)^{11/6} X_b(y)^{-1/3} \psi(\mathcal{D}) \tag{2.140}$$

where γ_0 is a calibration parameter, H_b is wave height at breaking determined for the maximum value of the wave breaking dissipation, γ_b is the wave saturation coefficient, X_b is the length of the breaking zone and Φ is a shape function:

$$\psi = \frac{1 + b + \tanh\left((\alpha D_c + z_b)/L_d\right)}{1 + b + \tanh(\alpha D_c / L_d)} \qquad \text{for} \quad \mathcal{D} > 0$$

$$\psi = \exp\left(-\left(\frac{x - x_s}{X_{swash}}\right)^4\right) \qquad \text{for} \quad \mathcal{D} \leq 0 \tag{2.141}$$

where D_c is the depth of closure, α and b are shape parameters and L_d controls the decay rate. For the dry region $\mathcal{D} \leq 0$, $(x - x_s)$ represents the distance to the shoreline and X_{swash} is the width of the swash zone:

$$(x - x_s) = \beta_s |\mathcal{D}|$$
$$X_{swash} = \beta_s A_{swash}$$

where β_s is the averaged equilibrium slope of the swash zone.

Quasi three-dimensional sediment transport formulation

This formulation is the equivalent of the showed in subsection 2.5.2, adding three-dimensionality and wave skewness and asymmetry to describe chrosshore processes. This contribution modify the advective term, while the diffusive transport remains equivalent to 2.130:

$$\vec{q}_{adv} = \vec{q}_{ss} + \vec{q}_{bl} \tag{2.142}$$

The first term represents the suspended sediment transport and the second the bed-load one.

The suspended transport is calculated integrating the time averaged sediment flux:

$$\vec{q}_{ss} = \frac{1}{D} \left\langle \int_{-\tilde{z}_b}^{\tilde{z}_s} C_z \vec{\tilde{u}}_w dz \right\rangle \tag{2.143}$$

where C_z is the depth-distribution of the volumetric sediment concentration can be formulated by:

$$C_z = C_0 \exp\left(-\frac{w_s}{\epsilon}(z + z_b)\right) \tag{2.144}$$

where w_s is the sediment fall velocity and ϵ is a eddy viscosity. Aagard (2002) found an empirical relation for w_s/ϵ from laboratoty data:

$$\frac{w_s}{\epsilon} = 17.543 \exp\left(-4.09 \frac{H_{rms}}{D}\right) \tag{2.145}$$

The value of C_0 can be estimated from SVR formula for the suspended wave stirring:

$$\begin{aligned}\alpha_{ss} &= C_0 \int_{-z_b}^{z_s} \exp\left(-\frac{w_s}{\epsilon}(z+z_b)\right)dz \\ &= C_0 \frac{\epsilon}{w_s}\left(\exp\left(-\frac{w_s}{\epsilon}D\right)-1\right)\end{aligned}$$

$$C_0 = \alpha_{ss}\frac{w_s}{\epsilon}\left(\frac{1}{\exp\left(-\frac{w_s}{\epsilon}D\right)-1}\right) \tag{2.146}$$

where α_{ss} is the suspended wave stirring:

$$\alpha_{ss} = A_{SS}\left[u_{str}-u_{crit}\right]^{2.4} \tag{2.147}$$

The bedload transport is estimated by:

$$\vec{q}_{bl} = \alpha_{bl}\left\langle \vec{u}_w(\delta)\right\rangle \tag{2.148}$$

where α_{bl} is the bedload wave stirring and $\vec{u}_w$ is the instaneous total velocity (equation 2.72) evaluated at $z+z_b = \delta$, where δ is the thickness of the wave boundary layer. From SVR formula:

$$\alpha_{bl} = A_{SB}\left[u_{str}-u_{crit}\right]^{2.4} \tag{2.149}$$

The value of δ can be estimated by (Van Rijn 1993):

$$\frac{\delta}{u_w(\delta)} = 0.072\left(\frac{u_w(\delta)}{k_a}\right)^{-0.25} \tag{2.150}$$

where $k_a = 30z_0$ is the apparent bed roughness and $u_w(\delta)$ is the amplitude of the wave orbital motion at $z+z_b = \delta$.

Realize that the advective transport computed from the SVR version of 2.120 is the limit of 2.142 for linear waves and neglecting the three-dimensionality and the undertow contribution in velocity description($u_d = 0$).

As we said before, the diffusive term is equivalent to the describe in 2.130. The only difference is the use of $-\nabla z_b$ instead of ∇h since this formulation includes crosshore description:

$$\vec{q}_{dif} = \left(\alpha\gamma_{dslp}\frac{u_{dslp}}{|\tan\phi_c - |\nabla z_b||} + \gamma_{swash}\Gamma\frac{\max(D_w)}{\rho(s-1)g}\right)\nabla z_b \tag{2.151}$$

Energetic total load transport model

Bagnold (1963, 1966) modelled suspended and bed-load sediment transport rates by relating proportionally with flow energy dissipation inside the bottom boundary layer. Based in this work, Bailard (1981) obtained a expression for the time-averaged sediment transport in terms of the near-bottom free stream velocity. From Bailard formula, Hsu et al. (2006) presented a model which split the total load transport in wave and wave+currents contribution. This description has been used in morphodynamic models to predict accurately the evolution of the cross-shore profile in laboratory and field experiments (Hsu 2006; Dubarbier et al. 2013; Fernandez-Mora et al. 2013, 2015). Based on Hsu 2006, the total load transport including the

correction due to down-slope transport and the swash-zone transport introduced in section before is given by:

$$\vec{q} = (\vec{q}_w + \vec{q}_c)\left(1 + \frac{\gamma_{dslp}}{|\tan\phi_c - |\nabla z_b||}\nabla z_b\right) + \gamma_{swash}\Gamma\frac{\max(D_w)}{\rho(s-1)g}\nabla z_b \tag{2.152}$$

where $\vec{q}_w$ and $\vec{q}_c$ represent the waves and wave+currents contributions to the sediment transport respectively..

$$\vec{q}_w = \frac{C_w}{(s-1)g}\left(\frac{\epsilon_{bl}}{\tan\phi_c}\left\langle|\vec{u}_w|^2_\delta\,\vec{u}_w(\delta)\right\rangle + \frac{\epsilon_{ss}}{w_s}\left\langle|\vec{u}_w|^3_\delta\,\vec{u}_w(\delta)\right\rangle\right) \tag{2.153}$$

$$\vec{q}_c = \frac{C_c}{(s-1)g}\left(\frac{\epsilon_{bl}}{\tan\phi_c}\left\langle|\vec{u}|^2_\delta\right\rangle\vec{U}(\delta) + \frac{\epsilon_{ss}}{w_s}\left\langle|\vec{u}|^3_\delta\right\rangle\vec{U}(\delta)\right) \tag{2.154}$$

where C_w and C_s (set to $C_w = 0.048$ and $C_s = 0.053$, according to Hsu et al. (2006)) are wave and current friction coefficients; ϵ_{bl} and ϵ_{ss} are the bed-load and suspended-load transport efficiency (set to $\epsilon_{bl} = 0.135$ and $\epsilon_{ss} = 0.015$, according to Thornton and Humiston (1996) ; Gallagher et al. (1998)); w_s (Soulsby 1997) is the sediment fall velocity and $\vec{U}(\delta)$ is the currents velocity at the boundary layer:

$$w_s = \frac{\nu}{d_{50}}\left(\sqrt{10.36^2 + 1.049D_*^3} - 10.36\right) \tag{2.155}$$

$$\vec{U}(\delta) = \vec{u} + \vec{u}_d(\delta) \tag{2.156}$$

2.5.3 Boundary conditions

Offshore boundary condition

For offshore, no bed-evolution is imposed:

$$\left(\frac{\partial z_b}{\partial t}\right)_{x=x'_{off}} = 0 \tag{2.157}$$

Therefore, for the sediment fluxes:

$$\left(\frac{\partial q_x}{\partial x} + \frac{\partial q_y}{\partial y}\right)_{x=x'_{off}} = 0 \tag{2.158}$$

where the offshore boundary condition is imposed in the same grid-position as the wave driver and the hydrodynamic solver (figure 2.9).

Onshore boundary condition

For onshore, in the dry region sediment transport is constraint for swash-transport shape functions (equations 2.137 and 2.141). At the limit of domain, no transversal sediment flux is imposed:

$$q_x(L_x, y) = 0 \tag{2.159}$$

2.5.4 Morphological accelerator

In order to reduce simulation time, MORFAC (MOrphological FACtor, Roelvink (2006)) approach is implemented in the code. The basis of MORFAC is the fact of morphological evolution time scale is significant lower than hydrodynamic one. Therefore, the required time step (Δt) for updating hydrodynamics to a bottom change is assumed several times smaller than the morphological time step Δt_m:

$$\Delta t_m = m_F \Delta t \tag{2.160}$$

where $m_F \geq 1$ is the morphological accelerator factor. In Chapter 3 the use of this factor for study surf-zone morphodynamic patterns is discussed.

2.5.5 Numerical integration

In this section the integration method of equation 2.116 is showed. For simplifying purposes, the sediment fluxes is written in the following generic form:

$$\vec{q} = \vec{q}_{adv} + \gamma \boldsymbol{\nabla} z_b \tag{2.161}$$

Numerical scheme

Applying second-order implicit schema to equation 2.116 , the bed elevation at $t = t + \Delta t$ can be estimated by:

$$z_b = z_b^t + \frac{\Delta t_m}{2} \left(p' \boldsymbol{\nabla} \cdot \vec{q} + \left(\frac{\partial z_b}{\partial t} \right)^t \right) \tag{2.162}$$

where $p' = 1/(1-p)$ and superscript "$t + \Delta t$" is omitted following the same nomenclature introduced in the hydrodynamic solver section. Combining 2.162 with 2.161:

$$\vec{q} = \vec{q}_0 + \gamma p' \frac{\Delta t_m}{2} \boldsymbol{\nabla} \left(\boldsymbol{\nabla} \cdot \vec{q} \right) \tag{2.163}$$

where $\vec{q}_0$ is defined by:

$$\vec{q}_0 = \vec{q}_{adv} + \gamma \left(\boldsymbol{\nabla} z_b^t + \frac{\Delta t_m}{2} \boldsymbol{\nabla} \left(\frac{\partial z_b}{\partial t} \right)^t \right) \tag{2.164}$$

The above variable can be computed directly. Nevertheless, solving quasi-3D or energetic total load sediment parametrizations (equations 2.143 and 2.152) is not trivial . In appendix **??** analytical and numerical solutions of auxiliary integrals used to calculate these equations are presented.

In a similar way to the hydrodynamical solver, 2.163 and 2.162 are solved in four-stage method: explicit estimation, semi-implicit calculation of q_x, calculation of q_y and bottom update.

1) Explicit estimation:

$$z_b^e = z_b^t + \frac{\Delta t_m}{2}\left(3\left(\frac{\partial z_b}{\partial t}\right)^t - \left(\frac{\partial z_b}{\partial t}\right)^{t-\Delta t_m}\right) \tag{2.165}$$

$$q_x^e = q_{adv,x} + \gamma\frac{\partial z_b^e}{\partial x} \tag{2.166}$$

$$q_y^e = q_{adv,y} + \gamma\frac{\partial z_b^e}{\partial y} \tag{2.167}$$

2) q_x **calculation:** Using q_y^e in the x component of equation 2.163, q_x is obtained solving implicitly along x-direction the following equation:

$$q_x = q_{0,x} + \gamma p'\frac{\Delta t_m}{2}\left(\frac{\partial^2 q_x}{\partial x^2} + \frac{\partial^2 q_y^e}{\partial x \partial y}\right) \tag{2.168}$$

3) q_y **calculation:** In this step, the y component of 2.163 is integrated along y-direction using q_x calculated in the previous step:

$$q_y = q_{0,y} + \gamma p'\frac{\Delta t_m}{2}\left(\frac{\partial^2 q_x}{\partial x \partial y} + \frac{\partial^2 q_y}{\partial y^2}\right) \tag{2.169}$$

4) Bottom update: Once the sediment fluxes are calculated, the bottom elevation z_b is computed directly from equation 2.162 and the water depth D is updated,

In order to avoid even-odd artefacts, for each time step the execution order of q_x and q_y stages is switched.

Linear system and boundary conditions

In the same way as in the hydrodynamic solver section, computation of explicit steps is no detailed. (stages **1** and **4** of the above section) since is considered straightforward. In this section implicit integration of q_x and q_y is explained

- q_x **calculation**: Figure 2.18 shows grid nodes (i, j), $(i+1, j)$ and the variables related with q_x calculation. In figure notation, super-index "i" is always omitted as well as "j" when is alone. Dicretization of equation 2.168 in this subdomain leads to the following linear system:

$$\ell_x q_x^- + d_x q_x + u_x q_x^+ = b_x \tag{2.170}$$

where coefficients ℓ_x, d_x and u_x are given by:

$$\begin{aligned} d_x &= 1 - \gamma p'\frac{\Delta t_m}{2}[d^2 x] \\ u_x &= -\gamma p'\frac{\Delta t_m}{2}[d^2 x^+] \\ \ell_x &= -\gamma p'\frac{\Delta t_m}{2}[d^2 x^-] \end{aligned} \tag{2.171}$$

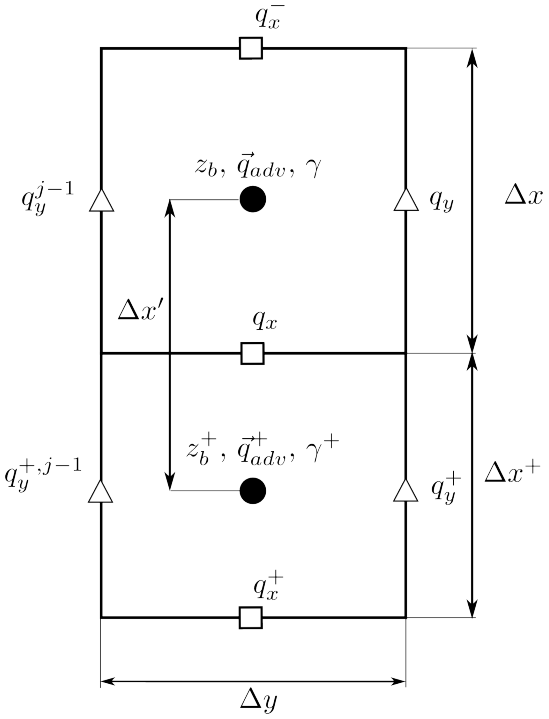

FIGURE 2.18: q_x node sub-domain

where $[d^2x^{\pm}]$ are the second-derivative coefficients defined in section 2.2.3 For coefficient b_x:

$$b_x = q_{0,x} - \gamma p' \frac{\Delta t_m}{2} \left(\frac{\left(q_y^e\right)^+ - \left(q_y^e\right)^{+,j-1} - \left(q_y^e\right) + \left(q_y^e\right)^{+,j-1}}{\Delta x' \Delta y} \right)$$

$$q_{0,x} = q_{adv.x} + \frac{\gamma}{\Delta x'} \left(z_b^+ - z_b + \frac{\Delta t_m}{2} \left(\frac{\partial z_b^+}{\partial t} + \frac{\partial z_b}{\partial t} \right) \right)^t \tag{2.172}$$

The above equations are defined in u-nodes, thus variables γ and $q_{adv,x}$ must be interpolated from c-nodes to u-nodes.

$$\gamma = \frac{\gamma^+ + \gamma}{2}$$

$$q_{adv,x} = \frac{q_{adv,x}^+ + q_{adv,x}}{2}$$

For a grid-column j, there are $n_x + 1$ unknowns and $n_x - 1$ equations as 2.170. The missing two equations required to solve the system are given by the off-shore and onshore boundary conditions. From section 2.5.3:

$$q_x^0 = q_x^1 + \Delta x_1 \left(\frac{\left(q_y^e\right)^1 - \left(q_y^e\right)^{1,j-1}}{\Delta y} \right) \tag{2.173}$$

$$q_x^{n_x} = 0 \tag{2.174}$$

The resulting tridiagonal-system is solved for each column applying Thomas algorithm, as was described in the hydrodynamic solver.

- q_y **calculation**: In this case, equation 2.169 is discretized in sub-domain (i, j), $(i, j+1)$, sketched in figure 2.19:

$$\ell_y q_y^- + d_y q_y + u_y q_y^+ = b_y \tag{2.175}$$

coefficients ℓ_y, d_y and u_y are given by:

$$\begin{aligned} d_y &= 1 + \gamma p' \frac{\Delta t_m}{\Delta y^2} \\ u_y &= -\gamma p' \frac{\Delta t_m}{2\Delta y^2} \\ \ell_y &= -\gamma p' \frac{\Delta t_m}{2\Delta y^2} \end{aligned} \tag{2.176}$$

For coefficient b_y:

$$\begin{aligned} b_y &= q_{0,y} - \gamma p' \frac{\Delta t_m}{2} \left(\frac{q_x^+ - q_x^{i-1,+} - q_x + q_x^{i-1}}{\Delta x \Delta y} \right) \\ q_{0,y} &= q_{adv,y} + \frac{\gamma}{\Delta y} \left(z_b^+ - z_b + \frac{\Delta t_m}{2} \left(\frac{\partial z_b^+}{\partial t} + \frac{\partial z_b}{\partial t} \right) \right)^t \end{aligned} \tag{2.177}$$

Analogously as was done for q_x, variables γ and $q_{adv,y}$ are interpolated from c-nodes to v-nodes.

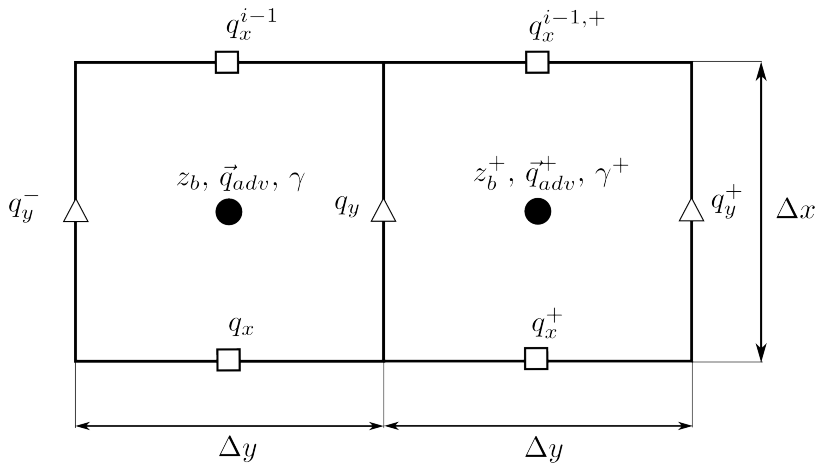

FIGURE 2.19: q_y node sub-domain

Periodic lateral boundary condition provides the required relations to define all unknowns of the above system:

$$\begin{aligned} q_y^0 &= q_y^{n_y} \\ q_y^{n_y+1} &= q_y^0 \end{aligned}$$

Once the periodic boundary condition is introduced, the resulting set of equation is a periodic tridiagonal system which is solved using El-Mikkay algorithm.

Prior to compute equation 2.162 to update the bottom elevation, the offshore value of the cross-shore sediment flux at the first u-node, q_x^0 is corrected:

$$q_x^0 = q_x^1 + \Delta x_1 \left(\frac{q_y^1 - q_y^{1,j-1}}{\Delta y} \right) \tag{2.178}$$