Simulating shoreline retreat in a 2-D hydro-morphologic model

Master of Science Thesis
by
Nguyen Quang Chien

Supervisor
Prof. Dr. Dano Roelvink

Examination Commitee
Prof. Dr. Dano Roelvink (UNESCO-IHE)
Ir. Mick van der Wegen (UNESCO-IHE)
Dr. Ap van Dongeren (WL—Delft Hydraulics)

Revised version
May 13, 2007

Dedicated to the memories of my grandfather

Abstract

XBeach is a simple 2-D, hydrodynamic-morphologic coupled model using a finite difference explicit scheme. One of the practical improvement for XBeach is to simulate bank erosion. We started with considering a simple case where a vertical bank retreats in a predefined direction. The shoreline is described as straight segments of line parallel to y-axis. Along with the retreat process of the shoreline, cells adjacent to the shoreline change their width, which has to be accounted in the continuity and momentum equations. Tests of shoreline retreat with straight and curved shoreline show stability of the model.

A shoreline with arbitrary orientation is then modelled as a polygon cutting through grid cells. We adjust the continuity equation and part of the advection terms in the momentum equation to make the model perform better in case of cut cells. Hydrodynamic tests with a straight, diagonal channel and bent channel show that the flow field calculated using the cut cell method is much better than the traditional "staircase" boundary. Finally, an application of modelling flow pattern among the groins in Haihau coast, Vietnam on real bathymetry data is carried out.

Keywords:
hydrodynamic modelling, vertical bank, shoreline retreat, two-dimensional model, finite volume method, cut cell method, explicit scheme, XBeach

Acknowledgements

This report has been done as my Master Thesis work in Coastal Engineering at UNESCO-IHE, Delft. The study is carried out in the framework of the XBeach project, co-operation between the US Army Corps of Engineers, UNESCO-IHE, Delft University and WL|Delft Hydraulics.

First, I would like to thank Rijkswaterstaat for funding my course in UNESCO-IHE. I have indeed learnt much, both from casual life and classes in Netherlands.

I would like to thank Prof. Dano Roelvink, my supervisor for his enthusiastic guidance throughout the work. As a senior researcher in coastal morphology, Prof. Roelvink can turn any serious problem in computational hydrodynamics into an interesting puzzle.

Prof. Roelvink, Dr. Ad Reniers, Dr. Ap van Dongeren and Dr. Walstra at WL | Delft Hydraulics together with U.S. experts initiated the XBeach project. I advocate XBeach and greatly appreciate the professionals who wrote the program, probably the first LGPL'ed code in 2D modelling of shallow water.

Many thanks to Mrs. Crosato in IHE for providing me insight knowledge on bank failure mechanisms along with relevant documents.

The data of Namdinh coast were provided by the Namdinh Dike Department. I would like to thank Mr. Thắng, deputy manager of the Department for kindly providing the data and guiding us in two field surveys to Namdinh.

This thesis is typeset in LaTeX, a GNU software system originally created by Donald Knuth and Leslie Lamport. Without softwares such as LaTeX, and other open source tools, this document would not come in digital format.

Also I appreciate the great help of IHE lecturers, patient librarians at IHE and TU-Delft for lending me helpful books.

My final thanks would be for my family and friends who encouraged me during my study in The Netherlands.

The image on the cover is reproduced from "Impact of a drop of water on a water surface". The picture was taken on 28 Jan 2006 by Roger McLassus and uploaded to Wikipedia Commons under the GFDL.

List of symbols

A	area of cell
C	Chézy coefficient
CD	Chart Datum
CFL	Courant-Lewy-Frederich coefficient
CV	computational volume
Δn_B	retreat distance of shoreline
$\Delta x, \Delta y$	cell size or grid spacing
Δt	timestep
δ	retreat distance in Sect. 2.3.2, or flow fraction of cell face (elsewhere)
η	water surface elevation
h	water depth
HW	High Water
i	water surface slope
$\mathbf{i}, \mathbf{j}$	unit vectors of x and y-axis
l_2	2^{nd} moment relative error
l_∞	maximum relative error
LW	Low Water
M	erodible coefficient (in m/s)
M^*	erodible coefficient (in kg/(m^{2}s))
$\mathbf{n}$	normal vector of a surface
nx, ny	number of cells in x, y direction
$\mathbf{q}$	flux vector
S	sediment transport rate
τ_w, τ_c	shear stress on bank and its critical (threshold) value
Θ	dimensionless area of cell
$\mathbf{u}, u, v, \lvert\mathbf{u}\rvert$	velocity vector, velocity in x, y direction, velocity norm (magnitude)
u_w, u_c	longitudinal near-bank velocity and its critical (threshold) value
x, y	cross-shore and longshore co-ordinates
z_b	bed elevation

Subscripts and superscripts

$_{i,j}$	subscripts denoting cell index (position) in mesh
$_{i+1/2}$	subscript denoting gridline index separating cells $(i, \cdot)$ and $(i+1, \cdot)$
$_{j+1/2}$	subscript denoting gridline index separating cells $(\cdot, j)$ and $(\cdot, j+1)$
$_{n,s,e,w}$	subscripts denoting faces of cell in 4 directions
^{n}	superscript denoting time level

Contents

Chapter 1

Introduction

1.1 An improvement for a hydrodynamic-morphologic model

In the past few decades, modelling of coastal areas has been developed rapidly. Using fast computers, coastal models currently can handle a broad spectrum of simulations, from wave propagation, hydrodynamics of coastal waters to marine sediment transport and salt intrusion. While in early times, analytical approaches could only give solutions to simple cases of straight coasts, at present many problems arising in areas with complex bathymetry have been solved. For real world projects, computer modelling is a must.

Popular models being used are 2D, quasi-3D and 3D models. For sediment transport models particularly, 3D models would be the most desirable, along with requirements of computer hardware, simulation time and available data needed for model calibration (O'Connor, 1994).

Although 3D models such as Delft3D are usually highly recommended for large projects, it requires long run time on computers with good hardware – and even with parallel networks. This is clearly not available for educational purposes, and a simplified model is worth to be considered.

UNESCO-IHE, WL | Delft Hydraulics and Delft University, in co-operation with US-ACE (US Army Corps of Engineers) have been developing XBeach, a simple 2DH hydrodynamic - sediment transport model. The project started in February 2006. XBeach is LGPL'ed[1] and therefore users can obtain it free of charge and modify the code to suit one's need. The program is coded in MatLab(R)[2], and Fortran, now executable and being incrementally developed.

XBeach implements an explicit finite difference staggered scheme to solve water surface levels, bed levels and depth-averaged velocities at grid points in the model. The design of model aims to couple the hydrodynamic and sediment transport processes. This "on-line" method is discussed in (Roelvink, 2006). Since XBeach currently uses uniform rectangu-

[1] GNU Lesser General Public Licensed. For further information, see `www.gnu.org`

[2] MatLab is a registered trademark of the MathWorks, `www.mathworks.com`

lar Cartesian grid cells, it is difficult to model the complex geometry of shoreline with an acceptable accuracy. Besides, XBeach lacks a module for calculating the retreat of shoreline.

The displacement of shoreline in a mildly sloped coast is usually calculated through the lowering of beach elevation of an area, which occurs when the inflowing volume of sediment is smaller than the inflowing volume of sediment. This method is quite mature for the case where depth contours are parallel to the coastline.

In nature, the shoreline may be a steep slope or even an almost vertical bank. Among various causes for the retreat of such a bank, hydrodynamic force is a main factor. Thus it is meaningful to couple the calculated flow pattern in hydrodynamic module with the erosion process. Flow velocity near the bank is considered an important parameter for actual erosion and transport of the bank material.

In this study, we get into the calculation by first considering the shoreline retreat in a predefined direction (which is perpendicular to the original shoreline). During the retreat process, the grid cells adjacent to the shoreline change in width, and their width must be updated after each timestep. The most important issue is that the continuity equation for these resized cells should remain correct.

The problem is then extended into a "true" 2-D case. Now representing the shoreline as line segments aligning in either x or y-direction ("staircase" representation of shoreline) is not enough. Orientation of any shoreline segment should be arbitrary, and so does the direction of bank retreat. When the shoreline is properly modelled in such detail, we not only obtain a more accurate shoreline in terms of geometry but also more correct description of flow field (thus hydrodynamic force) nearshore.

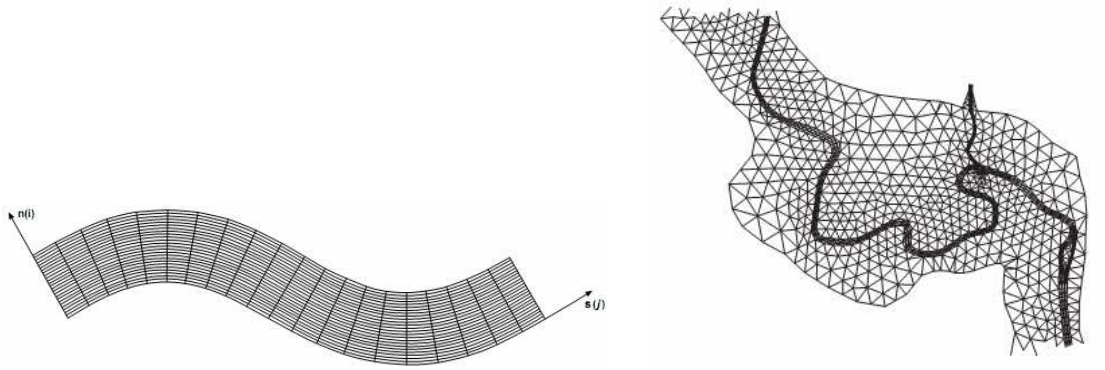
There have been three common methods of representing the boundary (shoreline) in high details:

- Boundary-conforming grid
- Unstructured conforming grid
- Cartesian cut cell

Nowadays, in 2-D modelling of coastal, estuarine areas and rivers, many software products introduce bank retreat (MIANDRAS, RIPA, MRIPA, MIKE-21C, Delft3D), which is based on the boundary conforming approach (gridlines are defined to follow the banks). Boundary-conforming methods, despite having the advantage of following the geometric boundary quite closely, require a grid generator; and for irregular boundaries it is not easy to implement.

Another difficulty for this method is that usually there are changes in geometry in simulation time. This leads to a need for some method of grid adaptation (Mosselman, 2005). Among the above models, MIKE-21C has a sophisticated grid adaptation scheme. MIANDRAS has a solution of adding/deleting grid points, while other models (RIPA, MRIPA) suffers from loss of smoothness and orthogonality.

Unstructured grid systems, based on the mature Finite Element Method (FEM), have the flexibility in complex geometries; but experience instability in cases where shock waves occur. A work on unstructured grid of coastal and oceanic waters is (Ham, 2006).



(a) Boundary conforming grid of a river. Source: (Darby, 2001)

(b) Unstructured grid of a floodplain. Source: (Bates et al., 2005)

Figure 1.1: Common types of grid in shallow water modelling

The cut cell method does not require a grid generator nor altering the gridlines during computation as the boundary-conforming method, and it uses simple 2-D arrays to store data (unlike the unstructured grid system), yet gives reasonably good results. Surprisingly, a study (Quirk, 1994), found that least attention was paid to the Cartesian cut cell method, presumably due to its conceptual simplicity.

Nevertheless, for a project that starts from scratch like XBeach, development of a simple method is encouraged. Hence, in this study we examine the use of cut cell method in representing flow adjacent to the irregular-shaped shoreline. The erosion rate of the bank due to flow field can then be estimated, which in turn results in the new position of shoreline in the next timestep.

Due to the limitation in time, the study offers issues in hydrodynamic computation in a grid with cut cells. Simple test cases are presented to show that by using cut cell method, the results can be improved compared with using a "staircase" representation of shoreline. After receiving reasonable flow field near the shoreline, retreat distance of shoreline within any cell can be roughly calculated using the procedures similarly to the simple case (whole shoreline retreats in a predefined direction).

As a test case, simulation was done for the coast of Haihau (a district in Namdinh province, Vietnam), where the land is bound by a low dike system, and frequently collapses due to high waves in storms. From 1905 to 1992, the loss of land in Haihau was up to 2.5 km wide within a strip of around 16 km long (Vinh et al., 1996).

In order to protect the land loss, the sea dike system has been improved. Several T-groins have been built in the area to trap sediment being transported away. In this study we run the model for simulating the flow field in the groin area during a short time period.

1.2 Objectives of the study

The study aims to solve following issues:

- Review the mechanism of shoreline erosion due to hydrodynamic forces of alongshore currents (Chap. 2),
- Defining a new computational scheme to calculate shoreline retreat in a predefined direction (Chap. 2),
- Implement a module in XBeach which simulates the spatial change of the coastline due to erosion (App. C),
- Improving the hydrodynamic module of XBeach to compute flow field in case boundaries do not follow, but cut through, gridlines (i.e. employing cut cell method) (Chap. 3),
- Validate the module by running XBeach for several simple cases (Chap. 3).
- Simulate the hydrodynamic process occurring at the groin area in Haihau coast. The resulting flow field is analysed and recommendations are drawn for the performance of XBeach (Chap. 3).

1.3 Outline

The thesis is divided into four chapters. We already had an overview of the problem and defined the aim of study. Before getting into numerical issues, Appendix A provides further details on XBeach. Then Chapter 2 elaborates on building the hydrodynamic calculation based on a simple assumption that the shoreline moves in a predefined direction. The chapter also reviews the mechanism of bank erosion.

In order to model erosion in reality, it is necessary to extend the computation into 2-D space. Chapter 3 presents a solution using a cut cell approach. Several simple cases are presented to validate this implementation. Appendix C will provide additional detail in implementing the code in MatLab.

As a practical situation, the flow field around T-groins in Haihau coast, Namdinh is investigated. Finally, in Chapter 4, conclusion and recommendations will be mentioned.

Chapter 2

Modelling coastline retreat in one dimension

In order to model the retreat of the coastline, a module for computing erosion has to be established in XBeach. This module is executed at the end of each time step, which gathers the nearshore flow field as input and issues the new position of coastline as an output. We will investigate the erosion module first, and later the hydrodynamics module. By doing so, after dealing with flow field we can write a ready-to-run routine which couples the nearshore hydrodynamic simulation and erosion of shoreline.

$$\begin{array}{ccccc} \text{Hydrodynamic} \\ \text{boundary condition} \end{array} \rightarrow \begin{array}{c} \text{Nearshore} \\ \text{flow field} \end{array} \rightarrow \begin{array}{c} \text{Erosion rate} \\ \text{or distance} \end{array}$$

2.1 Physical factors of bank erosion

The retreat of a shoreline depends on the characteristic of the water body and on the layout and sediment composition of the shore. At coastal zones, the damaging agents can be either sea level rise, deficit of alongshore sediment transport, overwash during a storm, or erosion of the bank/cliff. In rivers, the effect of waves are not present, and erosion is mainly due to floods in the rainy season and relates to soil pore pressure in the dry season.

In this study, we only consider the case with steep shorelines, which resembles a (river) bank. Failure of the bank can be due to three modes (Fischenich, 1989): 1) hydraulic forces that remove bed and bank material; 2) geotechnical instability; or 3) a combination of hydraulic and geotechnical factors.

Mosselman (1992) explains in more detail three causes for an erosive bank as follows:

- Erosion due to excessive shear stress. This directly relates to the alongshore velocity of the current.
- Degradation of the bed near the bank toe,
- Collapse of the bank in case the bank height exceeds a critical height.

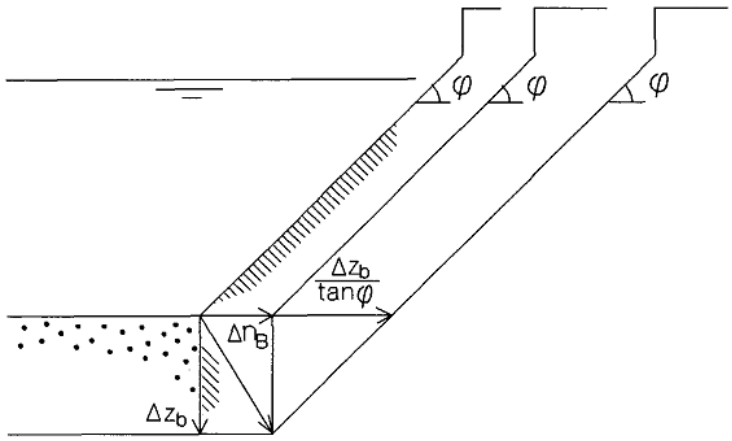


Figure 2.1: Erosion of a cohesive bank (Mosselman, 1992). Retreat distance of bank contains two components: the entrainment due to longshore current (Δn_B), and the component due to near-bank bed degradation ($\Delta z_b / \tan\phi$).

Sediment can generally be divided into two types: non-cohesive and cohesive; with different behaviour in erosion. In case of non-cohesive material, erosion is due to entrainment of discrete particles and can be calculated with Shield's formula, which is often found in textbooks such as (van Rijn, 1993).

In this study we concentrate on steep-sloped banks, which are often composed of cohesive material. This type of material is subjected to two types of erosion: (a) the entrainment of particles/aggregates and (b) the erosion of clusters or lumps. While the former is widely experimented, mathematical models for the latter are still simple with many assumptions (Zhu, 2006).

2.1.1 Effect of shear stress on bank

A commonly used formula for the surface erosion rate of cohesive material is mentioned in Parchure and Mehta (1985):

$$\frac{\partial n_B}{\partial t} = \begin{cases} M \cdot \dfrac{\tau_w - \tau_c}{\tau_c} & \text{for } \tau_w > \tau_c, \\ 0 & \text{for } \tau_w \leqslant \tau_c \end{cases} \tag{2.1}$$

where:

- M is the erodibility coefficient (m/s), which depends on the sediment characteristics,
- τ_w is the flow shear stress on the bank,
- τ_c is the critical shear stress,
- n_B is the transverse coordinate of the bank (m).

This formula assumes that erosion rate depends on the difference between τ_w and τ_c. There are also other agents which account for the erosion of bank, such as duration of the excessive shear stress on bank, number of shear stress peaks and variability of these peaks (Julian and Torres, 2006), among which the peak of excessive shear stress $(\tau_{bank} - \tau_c)_{max}$ is the most dominant factor.

In this study, we only consider the simple case where the magnitude $(\tau_w - \tau_c)$ is the only factor of erosion. So if $\tau_w < \tau_c$ then no erosion occurs.

Determining M

Partheniades (1965) proposed a formula to estimate the eroded mass of bed material in a second: $E = M^* (\tau_b - \tau_e)/\tau_e$ where M^* is a coefficient depending only on the geotechnical conditions; in reality $M^* = 10^{-5} \rightarrow 5 \times 10^{-4}$ kg/(m^{2}s).

Mosselman (1992) employed the formula of Partheniades and replaces the co-efficient M^* with a coefficient M represented in m/s. In a practical case of modelling the Ohře river (in Czech), M was chosen to be 10^{-7} m/s.

Winterwerp and van Kesteren (2004) suggested an analogous formula to be applied for sand (63 – 200 μm), but including additional parameters of grain size such as D_{50} and c_u.

Determining τ_w and τ_c

Normally the longitudinal component of the velocity adjacent to the bank is much larger then the perpendicular one, thus the longitudinal shear stress on the bank, τ_w can be calculated as suggested by Lane (1955).

$$\tau_w = \alpha_L \tau_{bs} \tag{2.2}$$

where $\alpha_L = 0.75$ for width-to-depth ratios above 5, which is met in the condition of most water bodies and τ_{bs} is the bed shear stress. The formula is equivalent to:

$$\tau_w = \alpha_L \frac{\rho g u_w^2}{C^2} \tag{2.3}$$

where u_w is the near-bank velocity and C is the Chézy coefficient.

For the value of τ_c, Julian and Torres (2006) suggested:

$$\tau_c = 0.1 + 0.1779(SC\%) + 0.0028(SC\%)^2 - 2.34 \times 10^{-5}(SC\%)^3 \tag{2.4}$$

where $SC\%$ is the silt-clay (< 0.063 mm) percentage. For pure sand, $\tau_c = 0.1$ N/m^2.

As an another approach, the average shear stress on the whole channel is derived from the following basic hydraulic formula:

$$\tau = \gamma R S$$

where the specific weight of water $\gamma = \rho g$, hydraulic radius of channel $R \approx h$ and S is the surface water slope in channel.

For shear stress on the river bank, Flintham and Carling (1988) suggests that $\tau_{bank} = \tau \cdot f(P_{bed}, P_{bank})$ where P_{bed} and P_{bank} are wet perimeters of the bed and bank, respectively, in a river cross-section. In our case with a coastline they are more difficult to define.

Lane (1953) calculates the critical shear stress on the bank as the shear stress on a transversal slope in case side slope of the bank is less than the friction angle of sediment material of bank. The formula can also be found in most books on sediment, such as (van Rijn, 1993).

An example on calculation of erosion rate due to excessive shear stress

We assume a vertical bank ($\phi = 90°$) with sediment $D_{50} = 0.2$ mm, composed of 90% sand and 10% clay subjected to a flow velocity $v = 0.3$ m/s. Assume a Chézy coefficient of 65 $\mathrm{m}^{0.5}/\mathrm{s}$.

The cohesive material is characterized by a value $M^* = 10^{-4}$ kg/(m^{2}s). Assume that the clay has an in-situ density $\gamma = 1500$ kg/m^3 then:

$$M = \frac{10^{-4}}{1500} = 6.7 \times 10^{-8}\mathrm{m/s}.$$

When calculating with cohesive material, it should be noted that the (buoyancy) weight of a particle is negligible compared to the attractive force between the particles. Therefore the shear stress on bank slope can be calculated without the effect of ϕ.

The shear stress of the slope, calculated after Formula 2.3 is: $\tau_w = 0.157$ N/m^2. The critical value, calculated after Formula 2.4 is $\tau_c = 0.118$ N/m^2 ($\rightarrow$ critical velocity $u_c = 0.260$ m/s). $\tau_w > \tau_c$, therefore erosion occurs and the calculated rate of erosion is:

$$E = 6.7 \times 10^{-8} \cdot \frac{0.157 - 0.118}{0.118} = 2.60 \times 10^{-8}\mathrm{kg/(m^2s)}$$

It means that under this condition, we will expect an erosion rate of 2.2 mm/day.

2.1.2 Bed degradation as a cause of bank collapse

Another eroding mechanism is the bank failure due to geotechnical instability. Mosselman (1992) suggested two types of mass failure due to bed degradation as follows. Firstly, if the bed lowered by Δz, in order to maintain the slope angle ϕ, the bank would retreat a distance of $\frac{\Delta z_b}{\tan\phi}$.

$$\frac{\partial n_B}{\partial t} = \begin{cases} -\dfrac{1}{\tan\phi}\dfrac{\partial z_b}{\partial t} & \text{for } \dfrac{\partial z_b}{\partial t} > 0, \\ 0 & \text{for } \dfrac{\partial z_b}{\partial t} \leqslant 0 \end{cases} \tag{2.5}$$

Secondly, during the erosion process, higher banks are more likely to collapse. The retreat rate accounting for the erosion due to bank height is:

$$\frac{\partial n_B}{\partial t} = \begin{cases} G \cdot \dfrac{H - H_c}{H_c} & \text{for } H > H_c, \\ 0 & \text{for } H \leqslant H_c \end{cases} \tag{2.6}$$

where

- G is the erodibility coefficient,
- H is the total bank height, $H = h_w + H_{fb}$ (water depth + free board).
- H_c is the critical bank height

2.1.3 Generalized formula for bank erosion

In case of a coastal area we focus on non-cohesive sediment. However, in general, we can consider a fraction ω of cohesive material and $(1 - \omega)$ of granular material (Mosselman, 1992). Then after erosion, only granular material is deposited on the bed, i.e. contributes in the (sediment) continuity equation. The rate of additional sediment volume produced from each length unit of the bank line is:

$$s = (1 - \omega)H\frac{\partial n_B}{\partial t}$$

For simplicity, in this study we only consider banks with vertical slope ($\phi = 90°$). Then the formula for bank retreat is:

$$\frac{\partial n_B}{\partial t} = M \cdot \frac{\tau_w - \tau_c}{\tau_c} + G \cdot \frac{H - H_c}{H_c} \tag{2.7}$$

As can be seen in Formula 2.3, τ_w is proportional to u_w^2. Thus we can change Formula 2.7 into:

$$\frac{\partial n_B}{\partial t} = M \cdot \frac{u_w^2 - u_c^2}{u_c^2} + G \cdot \frac{H - H_c}{H_c} \tag{2.8}$$

2.2 Modelling the shoreline as a polygon

In current version of XBeach, the shoreline is required to follow the edges of regular, rectangular grid cells. This is not flexible, as we cannot model the shoreline in an adequate resolution. So we investigate how it is done in the FLOW module of Delft3D.

2.2.1 Method employed in Delft3D-FLOW

In Delft3D-FLOW, the shoreline is considered as the gridline separating a dry cell ($z_b >$ water level) and a wet cell ($z_b <$ water level). The only way to perform shoreline retreat is to simulate erosion where the dry cell elevation decreases. As soon as the cell elevation is lower than water level, the cell has transformed from a dry cell into a wet one. It also means that the shoreline has retreated by one cell width.

Delft3D only modelled this process in an empirical way. Properties such as beach slope, diameter of sediment are not used; instead, a factor `THETSD` is employed. `THETSD` ranges

Figure 2.2: Representation of sediment transport between wet and dry cells in Delft3D. Sand is transported from dry cell (lightly shaded) into wet cells (darkly shaded).

from 0 to 1, where a value of 0 corresponds to non-erodible dry cells, and 1 for the case all effect of erosion in the neighbouring wet cell is transferred to the dry cell.

Thanh (2006) proposed an adjusted coefficient THET which limiting the erosion of dry cells:

$$\texttt{THET} = \min\left(\frac{h_1 - \texttt{SEDTHR}}{\texttt{HMAXTH} - \texttt{SEDTHR}} \cdot \texttt{THETSD},\ \texttt{THETSD}\right)$$

where h_1 is the local water depth, SEDTHR is the threshold depth for computing sediment transport, HMAXTH is parameter set to be larger than SEDTHR, e.g. HMAXTH = -999.

2.2.2 Proposed method for bank displacement in XBeach

The first attempt of approximation is to assume that shoreline moves in straight lines parallel to x-axis (see Fig. 2.3).

This kind of approximation has an advantage of being simple for being independent of the hydrodynamic part. It means that we only use the output of hydrodynamic calculation as source of erosion in each timestep; and the resulting change of shoreline does not affect the hydrodynamic condition in the next timestep.

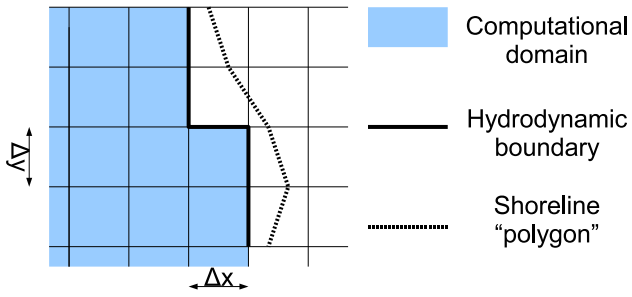


Figure 2.3: Schematization of shoreline as a boundary in model

The pseudo-code of this approach can be written as:

For each timestep **do**
    Calculate hydrodynamic condition
    Calculate rate of erosion e[j] **for** j = 1..ny

Calculate new position of shoreline xshore[j] = xshore[j] + e[j] * dt
End for

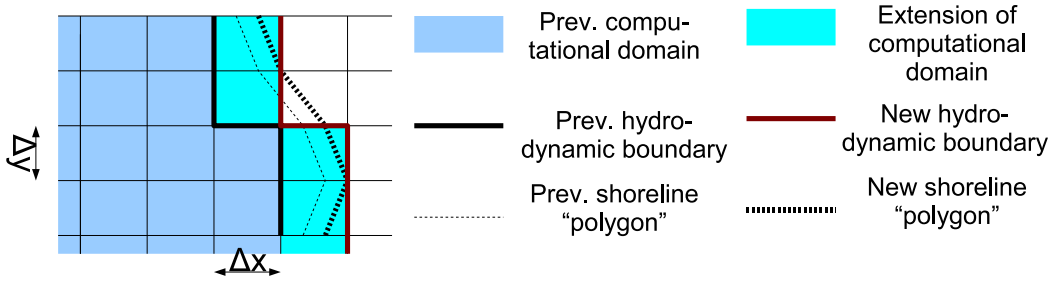


Figure 2.4: Extension of the computational domain due to change of shoreline

Despite the simplicity of this method, it is still necessary to adjust the computational domain as the shoreline retreats. Consider that after some timesteps, the shoreline has retreated over a distance of more than one grid cell. In order to reflect the system more correctly, the domain has to be extended one grid cell in width (i.e. x-direction in this case) (Fig. 2.4).

In this case, the pseudo-code can be described as:

XBound[j] = *Boundary of computational domain* **for** j = 1..ny
For each timestep **do**
Calculate hydrodynamic condition
Calculate rate of erosion e[j] **for** j = 1..ny
Calculate new position of shoreline xshore[j] = xshore[j] + e[j] * dt
If xshore[j] - XBound[j] > dx **then** XBound[j] = XBound[j] + dx
End for

This method fixes the problem of a rigid computational domain. Now the erosion rate of shoreline is computed from better hydrodynamic condition. However, even in this case, the hydrodynamic calculation is applied to only rectangular cells, which is an approximation. It would be more accurate if the grid cells can be distorted and follow the shoreline "polygon". This will be discussed in Sect. 2.3.

2.3 Displacement of shoreline in x-direction over time

The main point of this approach is to distort the grid cells which the shoreline "polygon" cuts through. For the first simple case, we assume that the shoreline moves in x-direction, which means Δy will always be constant during computation.

The basic assumption in this approach are:

- Each segment of shore line in a cell has an average retreat distance Δn_B in a computational timestep Δt.
- This distance is dependent on the v-velocity component of that cell (adjacent to the shoreline).
- The bank elevation z_{bank} and bed elevation z_b are constants in each cell. Bank slope is vertical.

The shoreline (assumed to run mostly parallel to y-axis) can be roughly split up into a number of straight segments, each segment is Δy in length and parallel to y-axis. The retreat of shoreline is simulated by displacing these segments in x-direction. In this method, we treat the shoreline in segments, hence the name "piecewise" method.

Alternatively, one can represent the shoreline by finding all the intersection between the shoreline polygon and gridlines parallel to x-axis. This chain of points (nodes) defines the shoreline in a "pointwise" way; displacement of shoreline is modelled by moving these points along the gridlines parallel to x-axis. Next we will discuss the calculation steps in each method.

2.3.1 "Piecewise" method

This method resembles the "partial step" approach which has been used in discretising uneven topography (Adcroft et al., 1997). In this method, the modelled shoreline consists of segments which are parallel to the gridlines (Fig. 2.5). The model is simple in terms of hydrodynamic computation and we can reconstruct the shoreline by joining the midpoint of each segment together (point I in Fig. 2.5). Furthermore, as XBeach defines the bathymetry at the centre of the computational cell, we can get the shoreline vertices (e.g., point I) with same y-coordinates as the bathymetry array.

For each segment, we calculate the mean (alongshore) velocity in the adjacent cell as:

$$v_{w,i,j} = \frac{1}{2}(v_{i,j-1} + v_{i,j}) \tag{2.9}$$

Denote Δn_B as the retreat distance (perpendicular to shoreline), then after Mosselman (1992):

$$\Delta n_B = \left(M \cdot \frac{v_w - v_c}{v_c} + G \cdot \frac{H - H_c}{H_c} \right) \Delta t \tag{2.10}$$

Here Δt is the morphologic time step, which is magnified f_{mor} times of the hydrodynamic time step, $\Delta t_{mor} = f_{mor}\, \Delta t_{hyd}$.

The new position of shoreline is calculated straightforwardly as in the following pseudocode:

Input *Initial position of the shoreline*

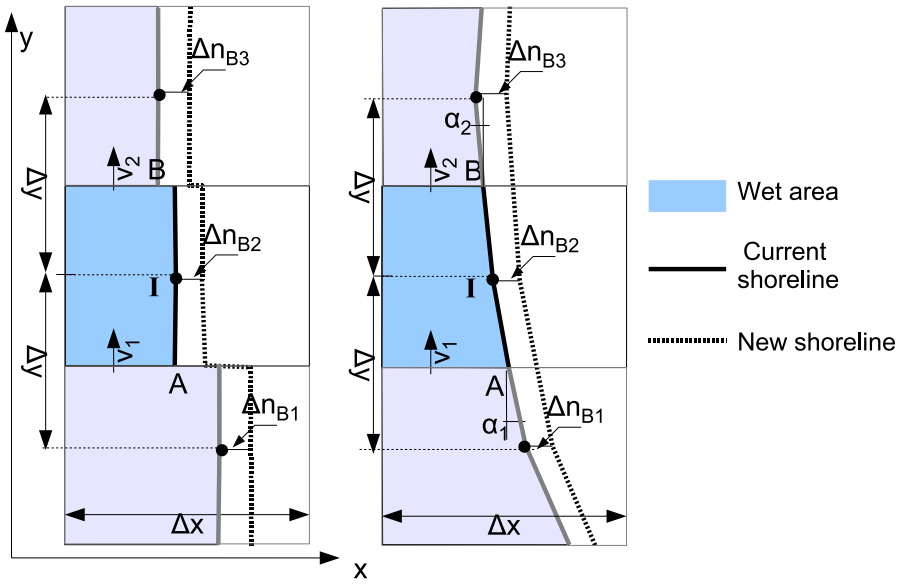


Figure 2.5: Schematization of coastline in "piecewise" approach.

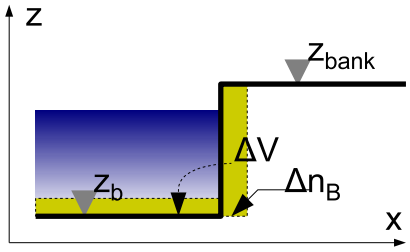


Figure 2.6: Sand volume ΔV to be eroded.

For each timestep **do**
Calculate hydrodynamic condition
Calculate retreat distance Dnb[j] **for** j = 1..ny
Calculate new position of shoreline xshore[j] = xshore[j] + Dnb[j]
End for

Along with erosion of the coastline, material from the eroded coast is transported to the nearby sea bed, hence causes change in bed elevation. To account for this variation, we use the continuity equation for sediment.

Suppose V is the volume of sediment in the cell, then $\Delta V_{bank}/\Delta t$ is the volumetric erosion rate inside the cell (Fig. 2.6), where:

$$\Delta V_{bank} = \Delta n_B \left(z_{bank} - z_b\right) \Delta y \tag{2.11}$$

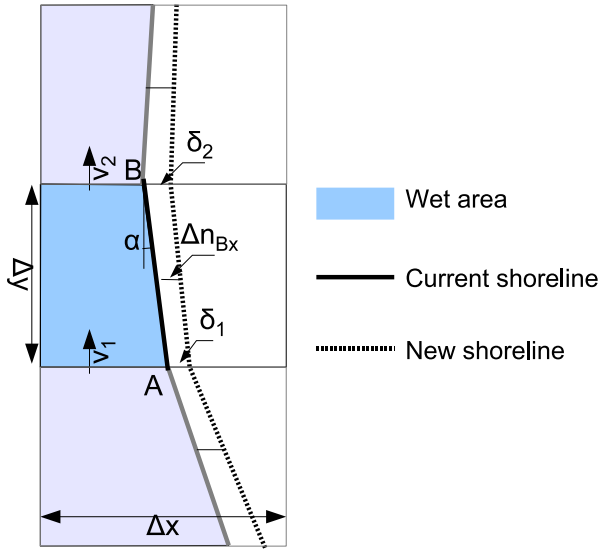


Figure 2.7: Schematisation of coastline in the "pointwise" approach.

Based on the sediment balance, the result change in bed elevation is calculated as:

$$\Delta z_b = \frac{1}{A_{wet}} \left(\sum S_{in} + \frac{\Delta V_{bank}}{\Delta t} \right) \Delta t \tag{2.12}$$

Here the net sediment transport S_{in} should be calculated according to the result from `transus` module. However, it can be simplified by assuming that there is no gradient in sediment transport between cells along the coastline, so $S_{in} = 0$. Another assumption is that the clay on bank and after settling to bed have the same porosity.

For cases with multiple shoreline retreat in one uniform direction, the method can still be applied. For instance, a simple case is a channel running in y-direction with two simultaneously eroding shorelines. However, this is only a trivial case of shoreline retreat in two dimensions, which will be discussed in Chap. 3.

2.3.2 "Pointwise" method

This approach, despite its more "natural" representation of shoreline polygon (whose vertices are on gridlines) is computationally more complex. In the pointwise method, the shoreline within a single cell sized $\Delta x \times \Delta y$ has a unique orientation characterized by the angle α (Fig. 2.7).

In this method, as each shoreline segment is inclined by an angle α, we have to determine the retreat distance in x-direction:

$$\Delta n_{Bx} = \frac{\Delta n_B}{cos\alpha}$$

Two points on the shoreline, A and B, has different retreat distance, δ_1 and δ_2, that:

$$\delta_1 + \delta_2 = 2\,\Delta n_{Bx}$$

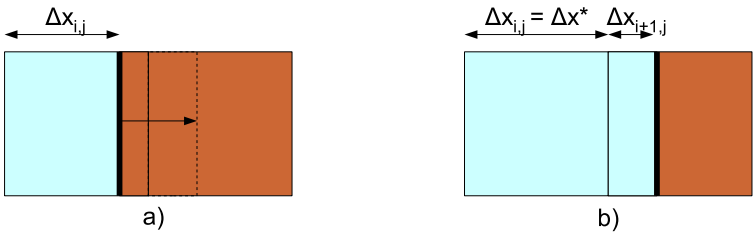


Figure 2.8: Change of cell widths in case the shoreline moves past the gridline

The problem raised above has $ny + 1$ unknowns, $\delta_1, \delta_2, \ldots, \delta_{ny+1}$. On the other hand, we only have ny equations: $\delta_{i-1} + \delta_i = \Delta n_{Bx,i}$ with $i = 2, \ldots, ny + 1$. If the value of δ_1 was initiated and calculation continued as: $\delta_i = 2\,\Delta n_{Bx,i} - \delta_{i-1}$ for $i = 2, \ldots, ny + 1$ then undershoot and overshoot would happen due to instability.

Instead, we work around by calculate $\delta_i = \frac{1}{2}(\Delta n_{Bx,i-1} + \Delta n_{Bx,i})$ and assign $\delta_1 = \Delta n_{Bx,1}$ and $\delta_{ny+1} = \Delta n_{Bx,ny}$. Then the new x-coordinates of the shoreline, x_i are updated with $x_i + \delta_i$.

The performance of the "piecewise" and "pointwise" methods are quite similar in case of straight shoreline and will not be presented here. In subsequent sections, however, only the "piecewise" method is used due to its simplicity.

2.4 Hydrodynamic adjustment for resized grid cells

2.4.1 Continuity and momentum equations

Continuity equation

Finding the new values of x_i is the solution for simple, fix grid case (sized Δx). In case of a resizing grid, these changes have to be accounted in cell sizes. Since the coastline divides the cell which it cuts through into two parts, this will narrow the width Δx of the cells adjacent to the shoreline.

We consider a uniform grid in x-direction (with grid spacing Δx^*) at $t = 0$. During simulation, width of the wet cells adjacent to the shoreline generally vary in the range (0, Δx^*] (some adjustments will be mentioned in Sect. 2.4.2). In a time period, the shoreline may move past a gridline. In this situation, the width of the wet cell exceeds Δx^*, and the extra width is assigned for the nearby cell (see Fig. 2.8).

An issue has to be adjusted when dealing with wet cells is that the correction of continuity equation. As in the continuity equation of XBeach:

$$\frac{\partial \eta}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} = 0$$

the flux terms are represented as hu and hv, which means that every cell has the same width and that width is suppressed in the equation.

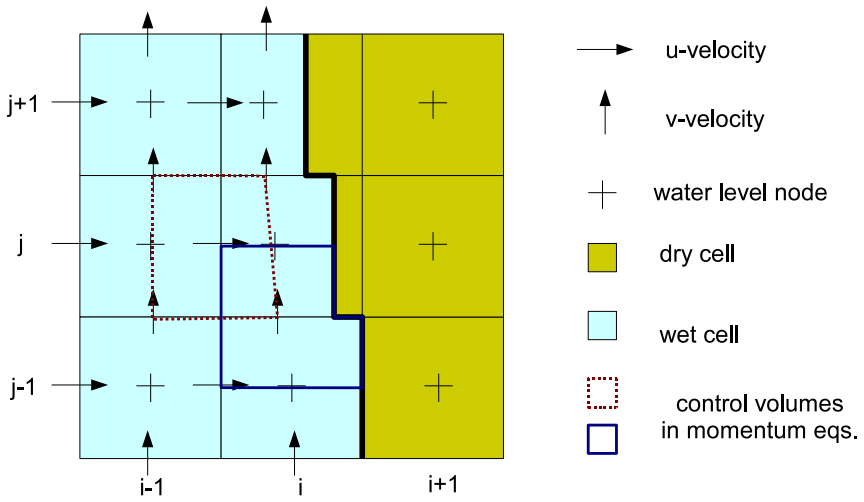


Figure 2.9: Staggered grid with variable Δx. Control volume with dotted line is used for calculating u-momentum of $u_{i-1/2,j}$; with solid line: for v-momentum of $v_{i,j-1/2}$

When dealing with cells of varying width, however, we have to consider the full a coefficient (δ) accounting for the width of irregular grid cells adjacent to the shoreline has to be included in the y-component of the flux.

$$\frac{\partial \eta}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial (\delta hv)}{\partial y} = 0$$

which results the discretisation as follows:

$$\frac{\eta_{i,j}^{n+1} - \eta_{i,j}^{n}}{\Delta t} + \frac{h_{i,j}^{n} u_{i,j}^{n} - h_{i-1,j}^{n} u_{i-1,j}^{n}}{\Delta x_{i,j}^{n}} + \frac{h_{i,j}^{n} v_{i,j}^{n} a_{i,j}^{n} - h_{i,j-1}^{n} v_{i,j-1}^{n} a_{i,j-1}^{n}}{\Delta x_{i,j}^{n} \Delta y} = 0 \tag{2.13}$$

where the aperture $a_{i,j}$ is the width where flow can pass between two cells $(i, j-1)$ and (i, j) (2.10):

$$a_{i,j} \equiv \min\left(\Delta x_{i,j-1}, \Delta x_{i,j}\right) \tag{2.14}$$

The only unknown in Eq. 2.13 is $\eta_{i,j}^{n+1}$. From now on, the superscript n denoting variables in the time level n will be suppressed for brevity.

Momentum equation

The momentum equations of XBeach read:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\tau_{bx}}{\rho h} - g\frac{\partial \eta}{\partial x} + \frac{F_x}{\rho h} \tag{2.15}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\tau_{by}}{\rho h} - g\frac{\partial \eta}{\partial y} + \frac{F_y}{\rho h} \tag{2.16}$$

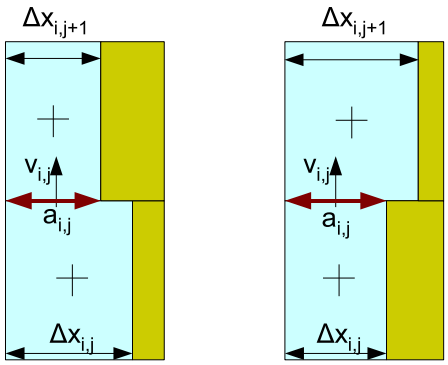


Figure 2.10: Aperture between cells (i, j) and $(i, j + 1)$

We have to adjust the discretisation of terms for resized cells. As can be seen from Fig. 2.9, along distorted cells, the v's are not aligned in y-direction as in a regular grid; the same problem holds for η. Therefore, we have to carefully specify the distance between computational nodes:

$$\left(\frac{\partial \eta}{\partial x}\right)_{i-1/2,j} \approx \frac{\eta_{i,j} - \eta_{i-1,j}}{x_{i,j} - x_{i-1,j}} = \frac{\eta_{i,j} - \eta_{i-1,j}}{\frac{\Delta x}{2} + \frac{\delta_{i,j}\,\Delta x}{2}}$$

where $\delta_{i,j} = \frac{\min(\Delta x_{i,j-1}, \Delta x_{i,j})}{\Delta x_{i,j}}$ ($= 1$ for the case in Fig. 2.9) .

$$\left(\frac{\partial v}{\partial x}\right)_{i,j-1/2} = \frac{v_{i+1,j-1/2} - v_{i-1,j-1/2}}{\frac{1}{2}(x^n_{i+1,j-1} + x_{i+1,j}) - x_{i-1,j-1/2}}$$

2.4.2 The CFL condition

For a regular grid with spacing $\mathcal{H}$, the timestep is chosen, considering stability, as follows:

$$\Delta t = CFL \cdot \min \frac{\mathcal{H}}{|\mathbf{u}| + \sqrt{gh}} \tag{2.17}$$

In case narrow cells present, normally we calculate Δt as:

$$\Delta t = CFL \cdot \min \left(\frac{\Delta x}{|u| + \sqrt{gh}}, \frac{\Delta y}{|v| + \sqrt{gh}} \right) \tag{2.18}$$

The presence of the term $\sqrt{gh}$ requires us to consider both x and y direction in Eq. 2.18. In cases where $\sqrt{gh}$ is negligible, $\frac{\Delta x}{|u|}$ and $\frac{\Delta y}{|v|}$ have the same order of magnitude even when Δx is very small. For a simple explanation, refer to (Kleefsman, 2005).

Normally with very narrow cells, $\Delta x \ll \Delta x^*$ then also $|u| \ll \sqrt{gh}$ and

$$\frac{\Delta x}{|u| + \sqrt{gh}} \ll \frac{\Delta y}{|v| + \sqrt{gh}}$$

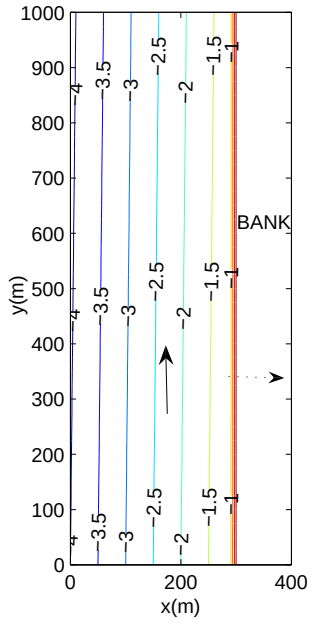


Figure 2.11: Bathymetry of a simple river used as a test case. The solid arrow shows flow direction, dotted arrow shows direction of shoreline retreat.

To fix this restriction in timestep we allow a cell width to be a bit greater than Δx^*, say a threshold $1.1\,\Delta x^*$. Then when a cell is wider than $1.1\,\Delta x^*$, it is split into two cells with width Δx^* and a small cell whose width is already greater than $0.1\,\Delta x^*$, which will yield less restriction in Δt.

2.5 Test cases

2.5.1 Simple test case with straight shoreline

Excess shear stress as the only cause for erosion

We perform a test in MatLab to estimate the distance of shoreline retreat. The computational domain contains 40×40 cells with $\Delta x = 10$ m and $\Delta y = 25$ m, featuring a straight shoreline of a river with a constant longitudinal slope of 10^{-4} (Fig. 2.11). The uniform (transversal) slope of bed is $1 : 100$, with elevation ranges from -4 m to -1 m. The bank ($+0.5$ m) is covered with grass, with a soil component of 90% sand and 10% clay, which results in a critical shear stress $\tau_c = 0.232$ N/m^2. The shoreline is taken as the gridline between cell with elevation -1 m and 0.5 m.

Velocity distribution follows the Chézy's formula: $v = C\sqrt{hi}$ with $C = 65$ m$^{0.5}$s^{-1} and $i = 10^{-4}$. This results in a nearshore $v = 0.65$ m/s. On the other hand, the critical velocity, u_c is calculated to be 0.365 m/s. Excess shear stress is the only cause for erosion; erodibility coefficient $M^* = 10^{-4}$ kg/(m^{2}s); which is equivalent to $M = 6 \times 10^{-8}$ m/s. The

morphological factor is set to 1000.

The upstream velocity boundary is of Dirichlet type and calculated according to Chézy's formula. The downstream water level boundary is of Neumann type: $(\partial\eta/\partial y)|_{y=ny\cdot\Delta y} = -i$. The boundary condition parallel to shore is $u = 0$ and the effect of waves is not included.

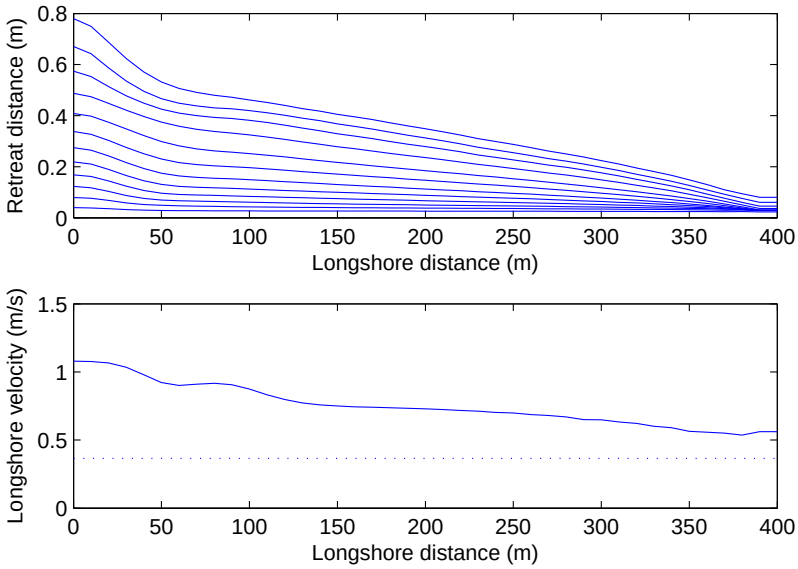


Figure 2.12: Shoreline retreat distance with $\Delta t = 6 \times 10^5$ s. Alongshore velocity at $t = 7.43 \times 10^6$ s

Result in Fig. 2.12 shows that initially the shoreline has a uniform retreat rate. For a longer simulation time, higher retreat distances are observed at the upstream boundary of the shoreline. The reason is that alongshore velocity is not uniform but decreases in the downstream direction (Fig. 2.12). It should be noted that the normal flow depth normally does not occur in natural rivers.

In the study, the upstream boundary is not proper in the sense that it introduce a slightly larger velocity compared to the whole section. However in case we do not determine a discharge in the river, an approximation of velocity following Chézy's formula is used.

Erosion due to excess shear stress and bank collapse

With the previous example, the bank height is 1.5 m. High banks are susceptible to failure due to geotechnical effect, yet it would be difficult to be modelled properly. An empirical "critical bank height" of 1 m is chosen. Also the coefficient G is set to be 10^{-6} m/s; and mass failure become the dominant factor of bank retreat in this case.

The result (Fig. 2.13) also shows a constant retreat velocity. However in this case, there is no presence of disturbation from downstreams.

So far the code had been tested for simple cases with uniform flow pattern in y-direction. The model performed stable even in case narrow cells exist. Next we will test the model

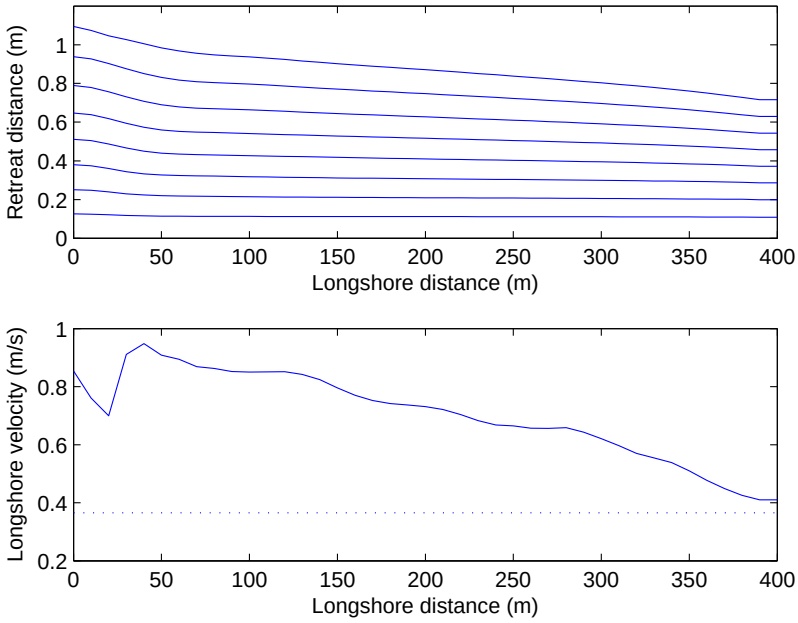


Figure 2.13: Bank retreat distance due to combined effect of excess shear stress and mass failure. Alongshore velocity at $t = 4.89 \times 10^6$ s

against the case of a curved shoreline, where the altered flow field may cause different retreat rate of the shoreline.

2.5.2 Test case – shoreline with a hump

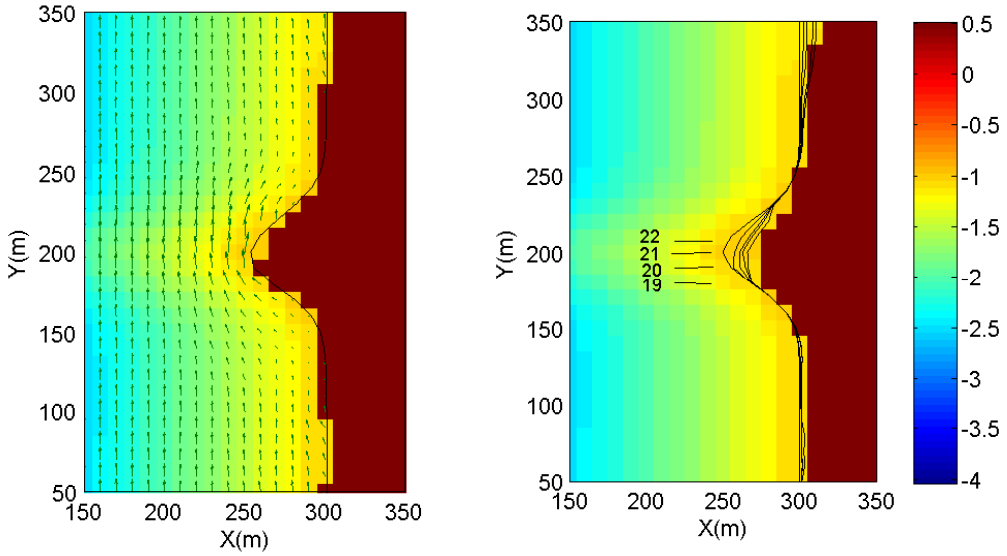
We perform a test case of the previous shoreline with $\Delta y = \Delta x = 10$ m. A Gaussian hump described with the following formula is super-imposed:

$$x = x_{base} - h_{hump} \cdot \exp\left(-\frac{y - y_0}{l^2}\right)$$

with $x_{base} = 330$ m ($\equiv$ base shoreline), $y_0 = 200$ m (midpoint of the shoreline) and parametric length $l = 3\,\Delta y = 30$ m.

During simulation, highest velocity occurs near the top of hump, 1.60 m/s due to contraction. The top of hump erodes with fastest speed compared to other positions on the coastline.

Fig. 2.15 shows the retreat distance of four selected points on the humped part of the shoreline (corresponding positions are shown in Fig. 2.14b). Point number 21 corresponds to the top of the hump retreat with the largest distance, as well as the largest retreat rate. In constrast, point 19 has very small retreat distance due to its small nearshore velocity. We also notice that after around timestep 1000, the retreat process of selected points seems to stop; this has to be checked afterwards.



(a) Velocity pattern

(b) Shoreline positions with $\Delta t = 3 \times 10^6$ s. Positions of four selected profiles are also shown.

Figure 2.14: Simulation of a hump-shaped shoreline.

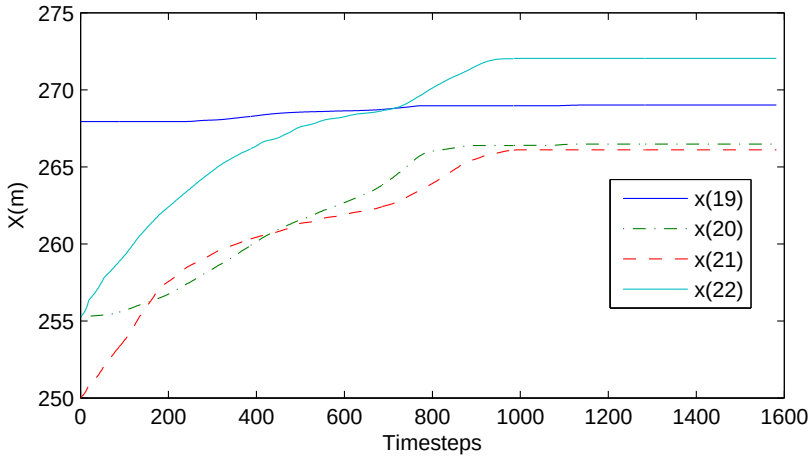


Figure 2.15: The retreat distance of four selected points

It can also be seen from Figs. 2.16 & 2.15 that, when the velocities drops below the critical value then retreat distances of corresponding points remain unchanged.

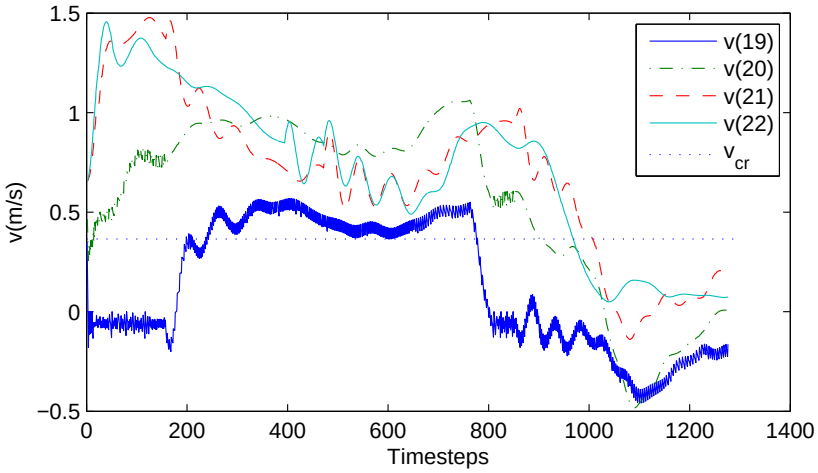


Figure 2.16: The nearshore velocity of four selected points

Chapter 3

Modelling shoreline retreat in two dimensions

The simple approach presented above is intuitive, yet has many shortcomings due to its simplicity. In nature, most shorelines are curve-shaped and therefore subjected to erosion in both directions. A more proper model should be established. In Computational Fluid Dynamics (CFD), currently there are two main methods to solve the problem:

- Using a boundary-fitted (boundary conforming), structured/unstructured grid, where grid cells can be resized during computation; in order to fit the closed boundary of computational domain with the real physical boundary.

- Using a immersed boundary method, where the closed boundary are polygons cutting through regular (Cartesian) grid cells. Unlike the former approach, here the grid can be predefined and independent of the physical geometry.

The use of boundary-fitted grid is well-known for its use in cases with complex geometrical boundaries. Unstructured grid are originated from the widely used finite element/finite volume method. However, boundary-fitted methods, when applied to modelling area of strongly varied depth, are susceptible to steep orography (Rosatti et al., 2005). For our erosion problem, remeshing a boundary-fitted grid relates to the matter of efficiency. Van der Meûlen (2003) suggests that CFD solvers using body-fitted grids often run into trouble when dealing with complex or moving boundaries, even when the grids are unstructured.

Another drawback of boundary-fitted grids is that they require a mesh generation process, and this takes a lot of resources, which can total to 25% of the computation time (Van der Meûlen, 2003). Cut cell methods, in contrary, can be applied to a simple rectangular Cartesian grid. This kind of grid can be quickly and automatically generated. During the computation, the grids always remain rectangular, only the state of some computational cells change. This allows us to get free from remeshing.

The non-conforming boundary method is also derived from Finite Volume Method (FVM), and therefore intrinsically ensures conservation. The challenge when employing

this method is to discretise the continuity and momentum equations for cells at the boundary, where distorted cells need special treatment in order to give reliable results, preferably second-order accuracy as with regular cells. Generally there have been two types of approach to this problem:

- The cut cell method, where the distorted shape of cells has physical significance in discretisation. Fluxes have to be determined for non-orthogonal cell faces. Elaboration on this can be found in (Kleefsman, 2005; Dröge, 2007).

- The ghost cell immersed boundary method (GCIBM), where the effect of irregular boundaries are converted into additional mathematical factors, thus discretisation is made as in the case of regular grid but with these extra factors. This method has been discussed by Shyy et al. (1996); Ye et al. (1999); Tseng and Ferziger (2003).

The ghost cell method, despite its attractive basis, also requires careful treatment on special cases where the boundary lies very near gridlines (Tseng and Ferziger, 2003). In this study we implement the cut cell method. Due to the limitation in time, only discretisation of continuity equation is considered in detail. In a staggered grid, control volumes for momentum equation have much more complex configurations and requires much more effort to treat.

3.1 Modelling the closed boundary

The physical shoreline is modelled as polygon/polyline cutting through grid cells (Fig. 3.1). The shoreline may be predefined as a set of data points such as (x_1, y_1), (x_2, y_2), $\ldots$, (x_n, y_n), and our task is to find the intersection between that boundary and the background Cartesian mesh (further detail can be found in (Causon et al., 2000)). Being cut by the boundary, any cell in the grid may fall into one of the following categories: *flow cell*, *cut cell* and *solid cell*[1]. Regarding the shape of flow part, a cut cell may fall into one of the three subcategories: triangular, trapezoidal and pentagonal (Fig. 3.2). In order to determine the centroid of cell (see Sect. 3.2), the orientation of flow part in a cut cell is also considered, and it results in four subtypes for each cell type. An exhaustive classification is mentioned in (Tu and Ruffin, 2002) (see Fig. B.1).

Given the grid and shoreline polygon, we need to do a preliminary scan in order to classify the cells in grid. At first, we assign all the cells inside a `nx` $\times$ `ny` grid to be of flow type. Then after having the co-ordinates of the land polygon, we can determine which cells lie inside the polygon; those are the solid cells. The rest are cut cells.

Next, we have to specify the dimensions of cut cells, namely the flow area of cell $A_{i,j}$, relative lengths δ of the flow boundary compared to the cell dimension (see Fig. 3.3). These δ values have the same physical meaning as in the 1D-case (Sect. 2.3), but to be determined

[1]Koh et al. (2005) classify the cell into four types. The extra type, called "boundary cell" is a fluid cell adjacent to a cut cell, whose discretized equations are different from those of "pure" fluid cells.

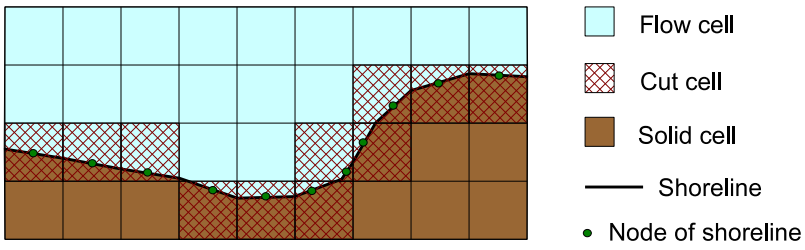


Figure 3.1: Representation of shoreline

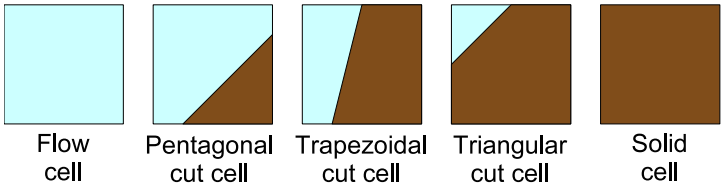


Figure 3.2: Different cell types.

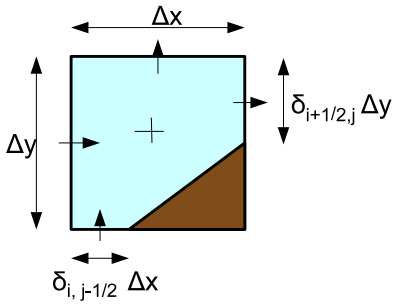


Figure 3.3: Dimension of a pentagonal cut cell. δ's are dimensionless quantities representing the fraction of cell sides which flows exist. In this case, for the left and the upper side of cell, $\delta_{i-1/2,j} = \delta_{i,j+1/2} = 1$.

in a slightly different way. Since there are no contractions/expansions of flow between cells, we do not have to determine minimum width between two cells.

For cell (i, j) we have four values $\delta_{i-1/2,j}$, $\delta_{i+1/2,j}$, $\delta_{i,j-1/2}$ and $\delta_{i,j+1/2}$. In a solid cells, all these $\delta = 0$. In a flow cell, all $\delta = 1$; and a cut cell otherwise. For a cut cell, its type is determined by the number of $\delta = 0$. Type 3 has two, type 4 has one and type 5 has none.

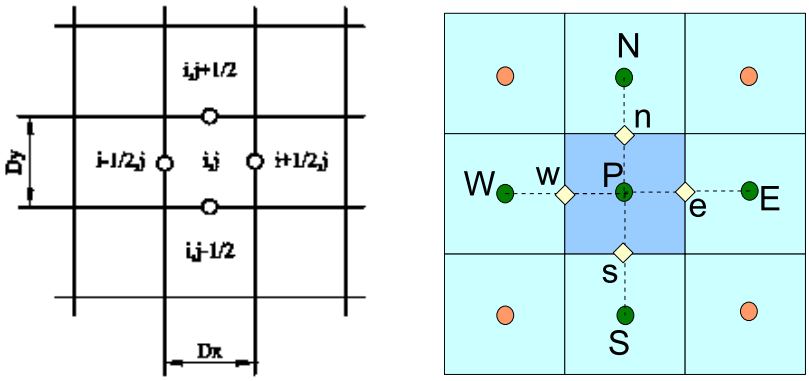
The cut cell orientation is needed to determine the centroid of the cut cells. Each cell type (3, 4 or 5) corresponds to four cell subtypes, with the flow part resides on four different directions (Fig. B.1).

It is not very difficult to automate the task of cell classification when the polygon is simple in topography. Graphic algorithms to determine the inclusion of a cell inside a polygon are available, for example the *floodfill* method, which is based on breadth-first search algorithm; or the ray-crossing method (O'Rourke, 1994). In this study, we

will not consider it in detail. For test cases with a small-scale grid, all the work of cell classification and geometric calculation will be determined by human common sense with little calculation.

3.2 Modelling hydrodynamic process

An Arakawa-C type staggered grid is established in the computational domain. The *integer nodes*, such as (i, j), locate at center of each cell, while *half-integer nodes* such as $(i-1/2, j)$, $(i+1/2, j)$, $(i, j-1/2)$ and $(i, j+1/2)$ locate on edges of each cell (Fig. 3.4). Beside the use of number notation, we also adopt the traditional notation where the considered cell has the centre node P and its four neighbours (N, S, E, W) corresponding to four directions. Lower case letters are reserved for midpoint on cell faces (n, s, e, w).



(a) Half integer notation(Rosatti et al., 2005)

(b) Directional notation

Figure 3.4: The five-point stencil in a staggered grid.

The water elevation values, $\eta_{i,j}$ are positioned at integer nodes, while velocities $u_{i\pm1/2,j}$ and $v_{i,j\pm1/2}$ are positioned at half-integer nodes. The grid is called staggered as different components of solution are positioned at different point sets in the computational space.

The similar configuration is applied to a cut cell, but with the cell area and position of nodes altered (Fig. 3.5). Also, in cell type 4 and 3, nodes on the solid edges are suppressed.

The flow area, A within a cut cell is calculated after Clarke et al. (1986):

$$A = \frac{1}{4}\left(S_x S_y + S_x D_y + S_y D_x - D_x D_y\right)$$

where S and D are sum and difference, respectively, of flow lengths in x and y directions.

$$S_x = \delta_w + \delta_e, \quad S_y = \delta_s + \delta_n, \quad D_x = |\delta_w - \delta_e|, \quad D_y = |\delta_s - \delta_n|.$$

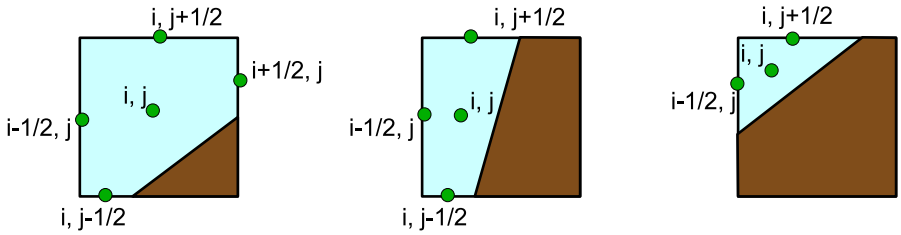


Figure 3.5: Positions of nodes in cut cells.

Beside the values of δ's, the orientation of the cell should be specified in order to correctly locate the centroid of cell. For example, with cut cell type 33 (according to the classification of Tu and Ruffin (2002)) we have:

$$\begin{cases} x_P = x_w + \frac{1}{3}\delta_n \Delta x \\ y_P = y_n + \frac{1}{3}\delta_w \Delta y \end{cases}$$

where x_P and y_P are co-ordinates of the cut cell centroid, x_w and y_n: for the cell edges.

3.2.1 Discretising the continuity equation

The continuity equation for a normal cell P (see Fig. 3.4b) reads:

$$\frac{\partial \eta}{\partial t} + \frac{\partial uh}{\partial x} + \frac{\partial vh}{\partial y} = 0 \tag{3.1}$$

Discretise 3.1 using directional notation:

$$\frac{\eta_P^{n+1} - \eta_P^n}{\Delta t} + \frac{u_e^n h_e^n - u_w^n h_w^n}{x_e - x_w} + \frac{v_n^n h_n^n - v_s^n h_s^n}{y_n - y_s} = 0 \tag{3.2}$$

Here all v and h are values in the time level n; from now on this superscript is suppressed for brevity. The unknown, $\eta_{i,j}^{n+1}$ is then calculated as:

$$\eta_P^{n+1} = \eta_P - \frac{\Delta t}{\Delta x}\left(u_e h_e - u_w h_w\right) - \frac{\Delta t}{\Delta y}\left(v_n h_n - v_s h_s\right) \tag{3.3}$$

Dealing with a cut cell P, the continuity equations must be rewritten. Denote δ_n, δ_s, δ_e, δ_w as relative section of flow on a cut cell edge, A_P as the area of cut cell and

$$\Theta_P \equiv \frac{A_P}{\Delta x \Delta y}$$

as the "dimensionless area". We consider the flow part of cut cell as a control volume V with variable water surface level η, boundary S and outer normal vector $\mathbf{n}$, with flux (discharge) $\mathbf{q}$ then the generalized continuity equation for the volume is:

$$\frac{\partial}{\partial t}\int_A (\eta_{i,j} - z_{b,i,j})\, dx\, dy + \oint_S \mathbf{q}\cdot\mathbf{n}\, dS = 0 \tag{3.4}$$

where the integrals are evaluated as:

$$\int_A (\eta_{i,j} - z_{b,i,j})\, dx\, dy = A_{i,j}\, (\eta_{i,j} - z_{b,i,j})$$

$$\oint_S \mathbf{q} \cdot \mathbf{n}\, dS = \sum_{n,s,e,w} (\text{velocity}) \cdot h \cdot \delta \cdot (\Delta x \text{ or } \Delta y)$$

Discretising Eq. 3.4 using the above relations:

$$A_P \frac{\eta_P^{n+1} - \eta_P}{\Delta t} + \frac{u_e h_e \delta_e \Delta y - u_w h_w \delta_w \Delta y}{x_e - x_w} + \frac{v_n h_n \delta_n \Delta x - v_s h_s \delta_s \Delta x}{y_n - y_s} = 0 \tag{3.5}$$

Subtituting $A_{i,j} = \Theta_{i,j} \Delta x \Delta y$ yields:

$$\Theta_P\, \eta_P^{n+1} = \Theta_P\, \eta_P - \frac{\Delta t}{\Delta x} (u_e h_e \delta_e - u_w h_w \delta_w) - \frac{\Delta t}{\Delta y} (v_n h_n \delta_n - v_w h_w \delta_w) \tag{3.6}$$

So Eq. 3.6 is very similar to Eq. 3.3, the only difference are dimensionless quantities: Θ and δ. The introduction of Θ gives more convenience to calculation, as other authors suggested the use of similar concepts: "cell capacity" $\kappa_{i,j}$ (Calhoun and LeVeque, 1999; Colella et al., 2004), or "volume fraction" $\Lambda_{i,j,k}$ (Pember et al., 1993; Johansen and Colella, 1998).

There remains a problem in Eq. 3.6: how are the values $h_{i\pm1/2,j}$ and $h_{i,j\pm1/2}$ evaluated? The generalized form is:

$$h_{i+1/2} = \lambda h_i + (1 - \lambda) h_{i+1}$$

where the coefficient $\lambda \in [0, 1]$. This value cannot be arbitrarily chosen, but depends on the flow itself. Various references, such as (P. W. Hemker; Ferziger and Perić, 2002), pointed out that in order to acquire stability, the scheme should be *upwind*, i.e. if point i is upstreams w.r.t. point $i + 1$ considering the flow in x-direction, then we should let $h_{i+1/2} = h_i$, otherwise we let $h_{i+1/2} = h_{i+1}$.

The timestep Δt is chosen based on the Courant condition, but for different cases with different cut cell configurations, we use a simple "equivalent grid spacing" $\sqrt{\Theta\, \Delta x\, \Delta y}$ instead of Δx and Δy:

$$\Delta t = CFL \cdot \min \frac{\sqrt{\Theta\, \Delta x\, \Delta y}}{|\mathbf{u}| + \sqrt{gh}} \tag{3.7}$$

Certainly, for cut cells in irregular shapes, one have to consider the time restriction both in x and y direction, see e.g. (Causon et al., 2001; Mingham, 2003). However, our point is to make as simple as possible, and the condition (3.7) work reasonably well with the test cases in Sect. 3.3.

3.2.2 The momentum equations

The momentum equations in general case read:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\tau_{bx}}{\rho h} - g\frac{\partial \eta}{\partial x} + \frac{F_x}{\rho h} \tag{3.8}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\tau_{by}}{\rho h} - g\frac{\partial \eta}{\partial y} + \frac{F_y}{\rho h} \tag{3.9}$$

Discretising momentum equations for cut cells can be difficult due to the various complex types of control volumes for u and v (For detail, refer (Kleefsman, 2005; Dröge, 2007)). In shallow water flow, advection is an important term, and we will discuss more about it compared to other terms.

The advection term

We consider the horizontal momentum in a cell with the enclosed boundary S. The cell has four faces called S_e, S_n, S_w, S_s corresponding to the four directions. Then:

$$\text{Momentum flux} \equiv \oint_S (\mathbf{u}\cdot\mathbf{n})u\,dS = \int_{S_e} uudy + \int_{S_n} vudx - \int_{S_w} uudy - \int_{S_s} vudx$$

We evaluate the integrals by first averaging u over each surface to get $\bar{u}_n$, $\bar{u}_s$, $\bar{u}_e$, $\bar{u}_w$ at half-integer nodes and to place it outside the integrals.

The $u\frac{\partial u}{\partial x}$ (or $v\frac{\partial v}{\partial y}$) term: Convection term is discretised using the upwind method. For rectangular cells in "piecewise" method (Sect. 2.3.1), this discretisation seems to work well inside the hydrodynamic module. However, in cases where cut cells are presented, the cells are distorted and the location of u values are neither equally spaced nor aligned. Furthermore, we have to account for the flow fraction between u- and v-CVs.

The $v\frac{\partial u}{\partial y}$ (or $u\frac{\partial v}{\partial x}$) term: We discretise the term $v\frac{\partial u}{\partial y}$. In flow cells, it is straightforward to write:

$$\left(v\frac{\partial u}{\partial y}\right)_{i+1/2,j} \approx v_{i+1/2,j}\frac{u_{i+1/2,j+1} - u_{i+1/2,j-1}}{2\,\Delta y} \tag{3.10}$$

However with the presence of cut cell, we have to account for the flow fraction. For instance, in Fig. 3.6, we propose:

$$v\frac{\partial u}{\partial y}\bigg|_{12} \approx v_{12}\frac{u_{13} - u_{11}}{2\,\Delta y} \tag{3.11}$$

$$v\frac{\partial u}{\partial y}\bigg|_{22} \approx v_{22}\frac{u_{23} - u_{21}}{\frac{3}{2}\Delta y + \frac{\delta_{21}}{2}\Delta y} \tag{3.12}$$

$$v\frac{\partial u}{\partial y}\bigg|_{32} \approx v_{32}\frac{u_{33} - u_{32}}{\frac{1}{2}\Delta y + \frac{\delta_{21}}{2}\Delta y} \quad \text{(only two CVs 33 and 32 are involved)} \tag{3.13}$$

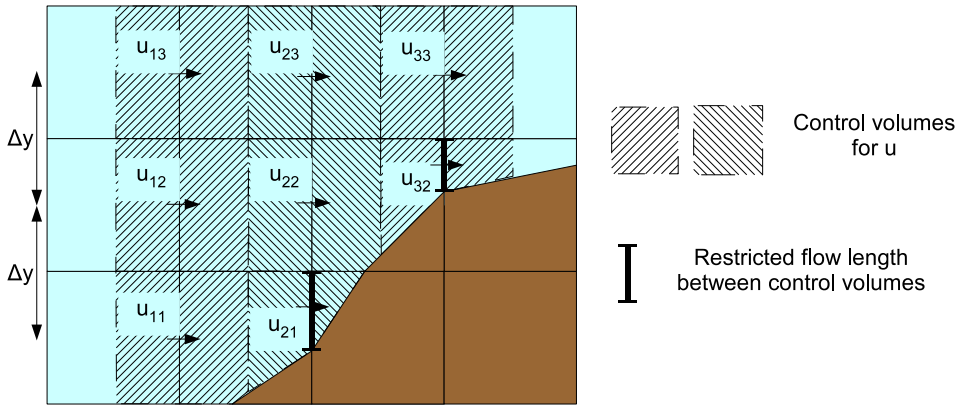


Figure 3.6: Schematization for discreting the $u\frac{\partial u}{\partial y}$ term

Source terms

In momentum equations, source terms include water gradient force, friction and wave force. Due to the limitation of this study, we will not consider them in detail. Furthermore, wave force is excluded from calculation in test cases of Sect. 3.3.

For the water level gradient at cell faces, XBeach currently uses a central approximation:

$$\left(\frac{\partial \eta}{\partial x}\right)_{i+1/2,j} = \frac{\eta_{i+1,j} - \eta_{i,j}}{x_{i+1,j} - x_{i,j}} \tag{3.14}$$

This linear interpolation gives second-order accuracy (Ferziger and Perić, 2002), but only at interface between two flow cells. In case of a cut cell, the cell centres are not aligned. Although there are more complex way to discretise this term (e.g. Dröge (2007)), we will still use this simple approximation.

The friction is computed from averaged values inside the u-CV or v-CV and generally need no special treatment in a cut cell.

3.2.3 Issues on moving the boundary in a cut cell grid

Along with the hydrodynamic process, the closed boundary constantly displaces due to erosion. In our model we expect a slow-moving boundary such that little perturbation is made to the flow field. The shoreline is essentially a reflective, free-slip boundary, i.e. nearshore velocity is tangential to the shoreline. The shoreline update is done after each hydrodynamic calculation, and new shoreline is taken into account for calculating flow field in the next timestep.

The movement of shoreline polygon is represented through movement of individual control points on cut cell faces. Let us consider a cell (Fig. 3.7) being cut by a polygon segment $A(x_a, y_a)$, $B(x_b, y_b)$ with the midpoint $M(\frac{x_a+x_b}{2}, \frac{y_a+y_b}{2})$. Then we are able to find

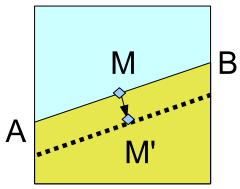


Figure 3.7: A shoreline segment within a cut cell

the unit normal vector of cut face $\mathbf{n}$,

$$\mathbf{n} = \frac{y_a - y_b}{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}}\mathbf{i} - \frac{x_a - x_b}{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}}\mathbf{j} \tag{3.15}$$

so that $\mathbf{n} \cdot \mathbf{AB} = 0$. Once the retreat distance Δn_B is determined (similarly to Sect. 2.3), the midpoint is moved to the new position M' such that $\mathbf{MM'} = -\Delta n_B \, \mathbf{n}$.

Simply drawing the new position of shoreline through M' and parallel to AB would not work since different segments of the shoreline retreat in different speeds. We have to establish a curve joining the M'_{ij} points and find the intersections of this new curve with gridlines to form the shoreline polygon at the new timestep. During the process, there may be times that a cell changes type in one timestep. This creates new flow fractions and therefore we have to carefully evaluate these quantities such that water volume is conserved.

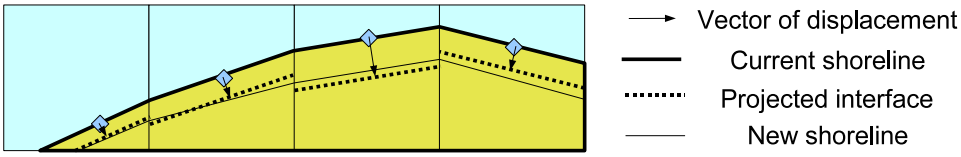


Figure 3.8: Movement of a shoreline segment

Modelling the moving the boundary in a Cartesian cut cell grid is a difficult problem and we will not go further in this study; instead we do some numerical tests on hydrodyamic modelling of the cut cells.

3.3 Hydrodynamic test cases

In this part, we examine the flow field in two simple cases using the cut cell method. The results are then compared to the case of using (regular) "staircase" boundary. The boundaries are hard, i.e. no erosion and sediment transport is accounted in this section.

3.3.1 Test case of a diagonal channel

Simulation of a straight channel aligned diagonally to gridlines is used widely as a test for shallow water flow models, such as in (Rosatti et al., 2005; Shyy et al., 1996). Here

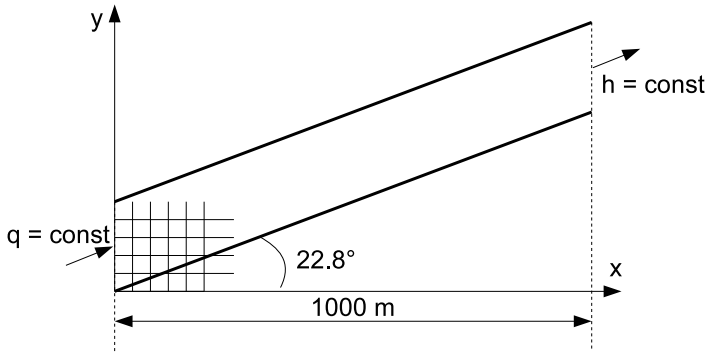


Figure 3.9: Parameters of the tested diagonal channel

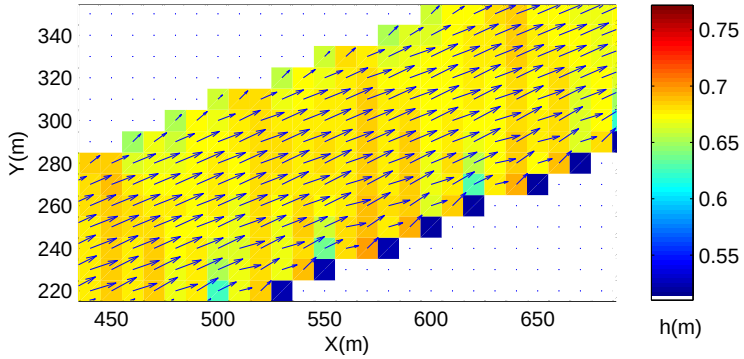


Figure 3.10: Flow pattern in the middle section of channel

we performed a test with a channel making an $\alpha = 22.8°$ angle with horizontal gridlines (Fig. 3.9). The channel is $100 \cos \alpha = 92.9$ m wide, with uniform bed slope $i_b = 10^{-3}$, and is supplied by an inflow of 100 m^3/s. Gridcell size is $\Delta x = \Delta y = 10$ m. Boundary condition for the inlet is a constant discharge and for the outlet — a constant depth.

This problem on uniform flow has an analytical solution with water depth $h_0 = 0.618$ m and velocity $|\mathbf{v_0}| = 1.741$ m/s, where the steady state is reached. In order to examine the discretisation error, we would first simulate the flow on square grid cells.

The result (Fig. 3.15) shows that the flow generally follows the channel orientation. On the centre of channel, the velocity and water elevation is quite uniform, while on the edges they greatly vary. In the cell corners, water is stagnant giving a rise in water level and a decrease of velocity.

By comparing the water depth and velocity components (u, v) at all flow cells against the analytical solution (0.618 m, 1.605 m/s and 0.675 m/s, respectively), we obtain the result in Table 3.1. The criteria to be used are relative errors l_∞ and l_2, which are calculated as:

$$l_\infty = \frac{\max |u_{i,j} - u_a|}{u_a} \quad \text{and} \quad l_2 = \frac{1}{N} \sqrt{\sum_{i,j} \left(\frac{u_{i,j}}{u_a} - 1 \right)^2}$$

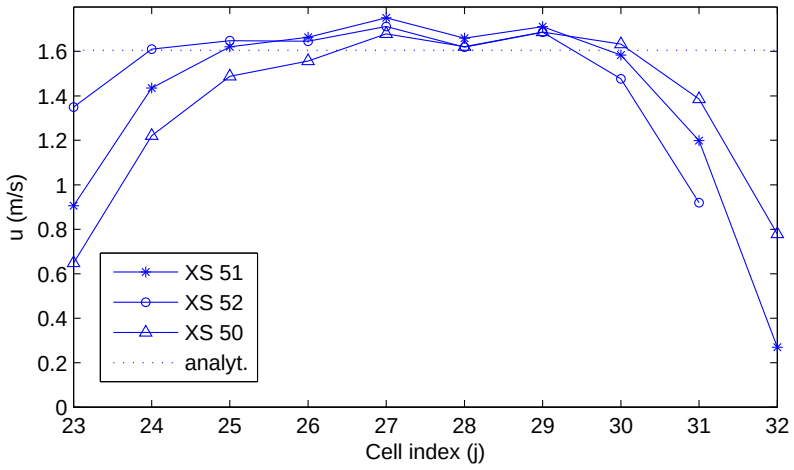


Figure 3.11: u velocity component at 3 consecutive cross-sections compared to the analytical value (square cell method)

where $u_{i,j}$ is the variable to be calculated at cell (i, j) and u_a is its analytical solution.

It should be noted that, although the l_∞ relative error is large, the l_2 relative error is much better; since there are a small number of cells near the edge of channel where flow is severely affected by the "staircase" boundary. The rest cells in the middle of channel showed quite good flow pattern, with flow direction parallel to the channel axis. The u velocity component tends to approximate the analytical value $|\mathbf{v_0}| \cos\alpha = 1.605$ m/s (Fig. 3.11).

The distribution of depth in the channel (Fig. 3.11) shows shallow water in the corners of the Southern shoreline.

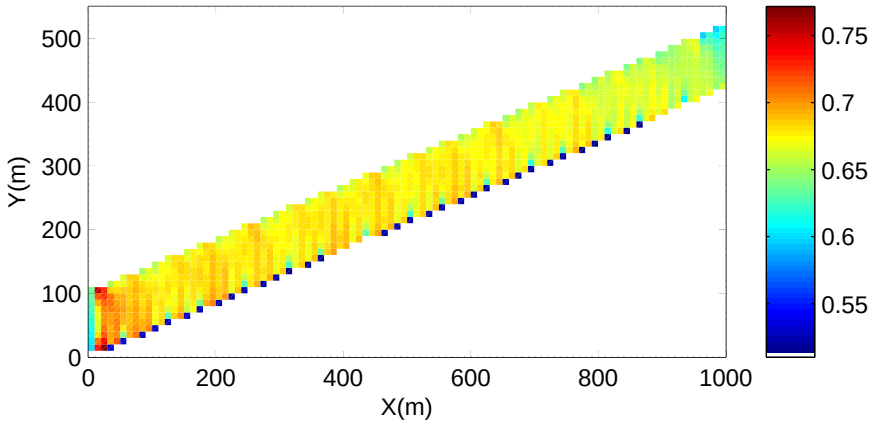


Figure 3.12: Distribution of flow depth in channel, staircase method

In this problem where analytical solution is achieved in steady state, it is necessary

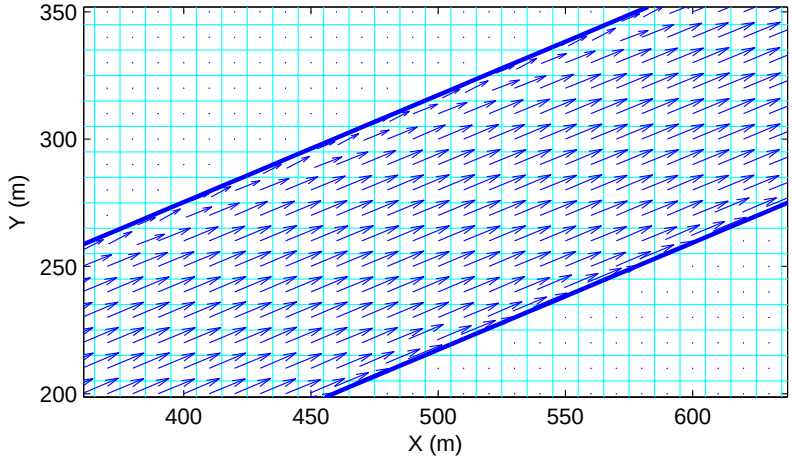


Figure 3.13: Flow field in the middle section of channel obtained from cut cell method

that the advection term is one order smaller than gravity or friction. For the simulation above, after averaging the gravity (G), friction (F) and advection term (A) in x direction for the whole flow domain, we get the ratio $G : F : A = 5.8 : 5.4 : 1$, implying that the advection effect is relatively small, which reflects the fact that the flow tends to reach the uniform state. If advection is excluded from calculation, we obtain the result in Table 3.1. So without advection, the calculated water depth is improved.

Table 3.1: Error of simulation on square cells

Advection	Criterion	h	u	v
Yes	l_2	0.0029	0.0077	0.0084
	l_∞	0.249	0.839	0.777
No	l_2	2.13×10^{-4}	0.0032	0.0086
	l_∞	0.029	0.334	0.801

Next we employ the cut cell description of channel boundary. Of the 1142 cells in the flow domain, 25% are cut cells and 1.6% are "very small" cells ($\Theta < 10^{-4}$). The typical Δt for this run is 0.0035 s.

At $t = 347$ s (nearly steady state), we obtain the flow field shown in Fig. 3.13. The errors due to cut cell simulation (Table 3.2) is notably smaller than those of staircase method. The flow profile (Fig. 3.14) is also better modelled, especially for cells near the boundary.

The work of Rosatti et al. (2005) on the same test case (Table 3.3) shows that the errors on square cells are of the same magnitude compared to this study. However, their result on cut cells are much more accurate, presumably because semi-Lagrangian method is applied in their model.

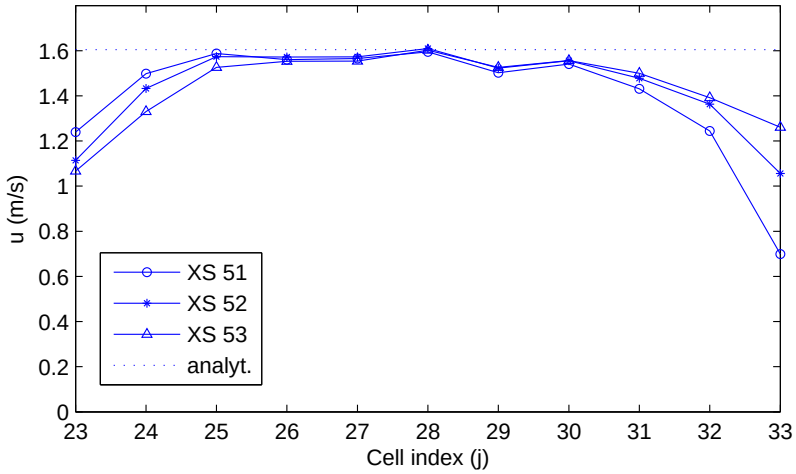


Figure 3.14: u velocity component at 3 consecutive cross-sections compared to the analytical value (cut cell method)

Table 3.2: Error of simulation on cut cells

Advection	Criterion	h	u	v
Yes	l_2	2.109×10^{-4}	0.0044	0.0045
	l_∞	0.030	0.620	0.621
No	l_2	5.56×10^{-4}	0.0024	0.0017
	l_∞	0.039	0.381	0.364

Table 3.3: Results of Rosatti et al. (2005)

Case	Criterion	h	u	v
Square cells, with advection	l_2	n/a	n/a	n/a
	l_∞	0.1	0.56	0.73
Square cells, w/o advection	l_2	n/a	n/a	n/a
	l_∞	0.0145	0.4	0.65
Cut cells, with advection	l_2	3.2×10^{-8}	1.6×10^{-7}	5.31×10^{-8}
	l_∞	6.8×10^{-5}	3.55×10^{-4}	0.0253
Cut cells, w/o advection	l_2	n/a	n/a	n/a
	l_∞	3.39×10^{-5}	2.2×10^{-3}	2.02×10^{-2}

3.3.2 Test case of a bent channel

A bent channel, with rectangular cross-section, 25 m wide joining two orthogonal straight channel sections is considered. The bed slope is 0.35×10^{-3}, Chézy coefficient 83.4 $\mathrm{m}^{0.5}\mathrm{s}^{-1}$. The bending curve is 90°, with inner radius $r_i = 125$ m and outer radius $r_o = 150$ m. Grid cell size is 5 m for both Δx and Δy.

The unit inflow discharge is 0.0617 m^2/s. Analytical solution for a straight channel

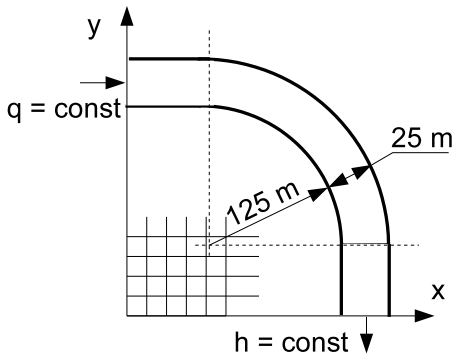


Figure 3.15: Parameters of the tested bent channel

with the same cross-section would be $h_0 = 0.116$ m and $v_0 = 0.532$ m/s.

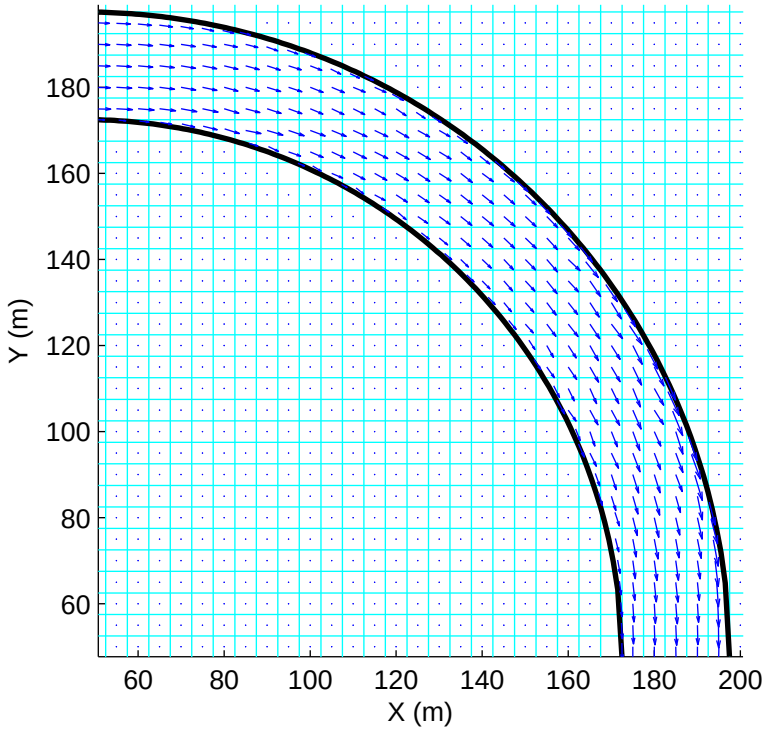
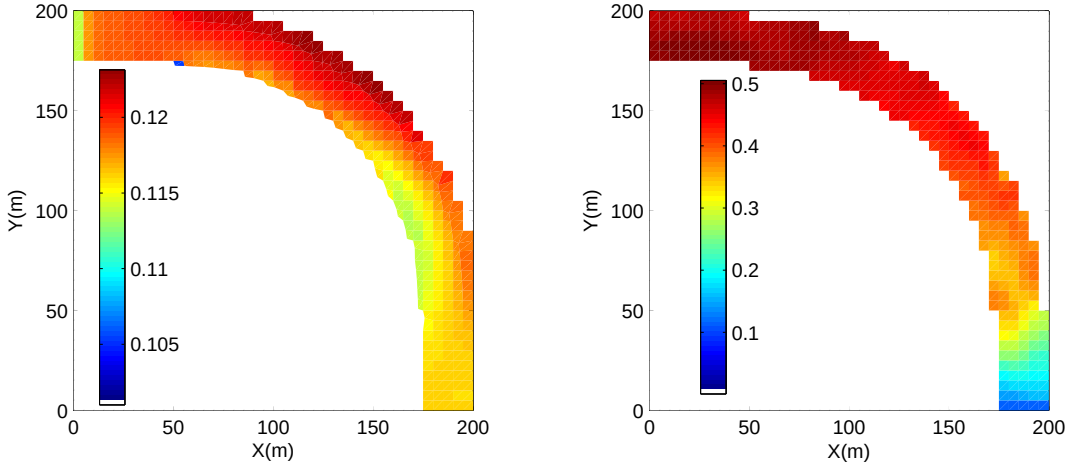


Figure 3.16: Flow field in the bending section

The resulting flow field (Fig. 3.16) shows that the flow follow the bend properly. This is a good result considering that the channel width is only $5\,\Delta x$ and the flow area comprises up to 28% cut cells.

The "staircase" method, in comparison, cannot give such reasonable output. Velocity

near the contraction usually turn in transversal direction, due to the abrupt change in geometry. This results in an unrealistic flow pattern and therefore cannot be used for our purpose.



(a) Cut cell, CFL = 0.5 (b) Square cell, CFL = 0.1

Figure 3.17: Distribution of depth (m) in bent channel

Regarding the distribution of depth in channel, the cut cell method (Fig. 3.17b) clearly yields better result compared to the square cell method (Fig. 3.17b). Even only with smaller CFL, the depth produced by cut cell shows a deeper area in the outer bend, which cannot be observed in the square cell case.

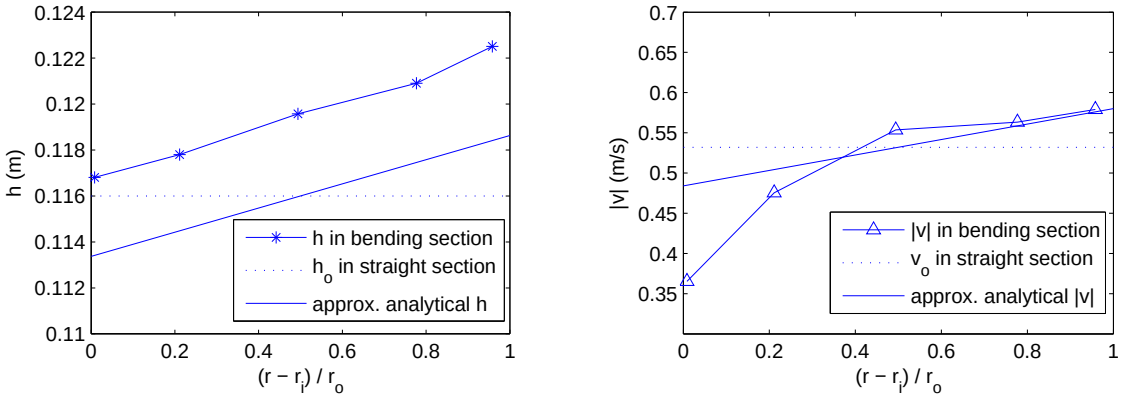
For this bending channel, no analytical solution is available. However, as suggested by Rosatti et al. (2005), we can examine the transversal slope in the middle section of the curved channel. There the flow is assumed to be in static rotation with a parabolic surface. The difference between water level (or water depth) of outer and inner bend is given by:

$$\Delta\eta = \frac{1}{g} v_0^2 \ln\left(\frac{r_o}{r_i}\right) = 5.260 \times 10^{-3}\text{m}$$

The result (Fig. 3.18a) shows that generally the water depth rises from inner to outer bank as expected. The difference in η is $5.7{\times}10^{-3}$ m. Similar to water depth, the velocity norm $|\mathbf{v}|$ also rises in the outward direction (Fig. 3.18b).

Compared to the approximated analytical values, the water depth is over-estimated in our model. The model also predicts low velocity at the inner bend. However, the performance when using cut cell method is much better than that of the "staircase" method. Velocity calculated from the "staircase" method is small, especially cells near the boundary; while the water depth is too large and is not shown in the figure.

In the test case of a diagonal channel channel, we already see that flow velocity in the cut cells of the North shoreline is significantly under-estimated. In this test case, velocity



(a) Water depth

(b) Velocity norm

Figure 3.18: Distribution of flow in the bent cross-section.

in the cut cells of the inner bend is too low. The common point in both cases is the velocity tends to reduce when flow gets in a "narrowing" cut cell. The problem possibly lies in the upwind discretisation of the advection terms. In this study, we will not examine this problem further. A concluding remark in this section is that by applying the cut cell method with appropriate treatments to the continuity and momentum equations, one can simulate the flow within irregular computational domain with better accuracy compared to a conventional "staircase" approach.

3.4 Flow pattern around T-groins in Haihau

3.4.1 The coast of Haihau

Topography and sediment

The coastline in Haihau — 27 km in length — has been experiencing severe erosion. At the erosion sites, the beach is narrow, only 100 – 200 m at low tide. The erosion rate, as provided by the Provincial Dike Department in Namdinh, is around 10 m/year around the period 1973–1990, which is quite alarming.

The beach of Namdinh contains fine-grained sand, with diameter from 0.1 mm to 0.15 mm. The thickness of this layer ranges from 0.5 m to 2.0 m. Samples analysed by Namdinh Sea Dike Service Department, 2001 showed that 98% of sediment belongs to sand types, the remaining 2% is of silt type ($D_{50} = 0.157$ mm, $D_{85} = 0.199$ mm).

Climate and meteorology

Namdinh is in tropical monsoon area with an annual average rainfall of 1600 – 1800 mm, 85% of which occurs in rainy season (from April to October). Typhoon and storms are

(a) Shoreline of Namdinh and Haihau district

(b) Groin field in Haihau under wave attack

Figure 3.19: The coast of Namdinh.

frequent between July and October. During the period 1911 – 1965, the region had experienced 40 typhoons. Strong typhoons usually originate in the West of Pacific Ocean, travelling through the Philippines. The average number of typhoons per year is around 5; however 10 typhoons were observed in each year 1964, 1973 and 1989. Typhoon Nikki hit Namdinh in 1996, causing a surge of 3.11 m.

3.4.2 Hydrodynamic boundary conditions

Tides

Based on the tidal map of Vietnam, Namdinh has an irregular diurnal tidal scheme with tidal range varying from 3 – 4 m. Data of Vietnamese Water Resources Institute, 2002 shows that extreme tidal current offshore of Van Ly village (in Haihau) is 0.45 m/s with a direction 310° w.r.t. N in flood tide, and 0.37 m/s with a direction 159° w.r.t. N in ebb tide.

Also in Van Ly, measured data in 19 years gives an MSL of 185 cm CD, max. HW 345 cm CD and min. LW −7 cm CD.

Field observations done by Hung et al. (2002) revealed that longshore current of Namdinh coast has an average velocity from 0.2 to 0.4 m/s and a maximum of 0.7 to 1.0 m/s at the depth of 2.5 m. The figures include tide-induced velocity. Longshore currents south-westwards in winter and north-eastwards in summer.

Wind and wave

There are two periods in a year in which wind blows in different directions. In winter time (from October to March), the dominant wind direction are north, northeast and east. In summer (from May to August), the dominant wind direction are south, southeast and southwest.

Together with long wind fetch, the wave height can be large, especially in typhoons. Observations in 1975 – 1987 showed that:

- In winter (from September to March), the sea was much rougher than in summer. Wave height is about 0.8 m - 1.0 m, with periods varying from 7 to 10 seconds. Predominant wave direction was northeast, and makes angles of about 30° to 45° with the shoreline.

- In summer (from April to August), there are fewer days with rough sea. However, strong storms usually happen in this season causing severe damage to the dike system. Average wave height varies from 0.65 m to 1.0 m with period ranging from 5 to 7 seconds. The prevailing wave direction is south and southeast.

3.4.3 Model set-up

Bathymetry

Namdinh has a system of sea dikes, with average crest elevation of 5.5 m. In calculation, the sea dikes can only withstand stroms with wind of 9 Beaufort scale and tidal water level of 5%-quantile. Along with upgrading the dike body, alternative measures have been taken to protect the erosion of foreshore which negatively affects the toe of dike. In Van Ly, five T-groins had been built to protect foreshore erosion (Fig. 3.20). This is the area to be modelled.

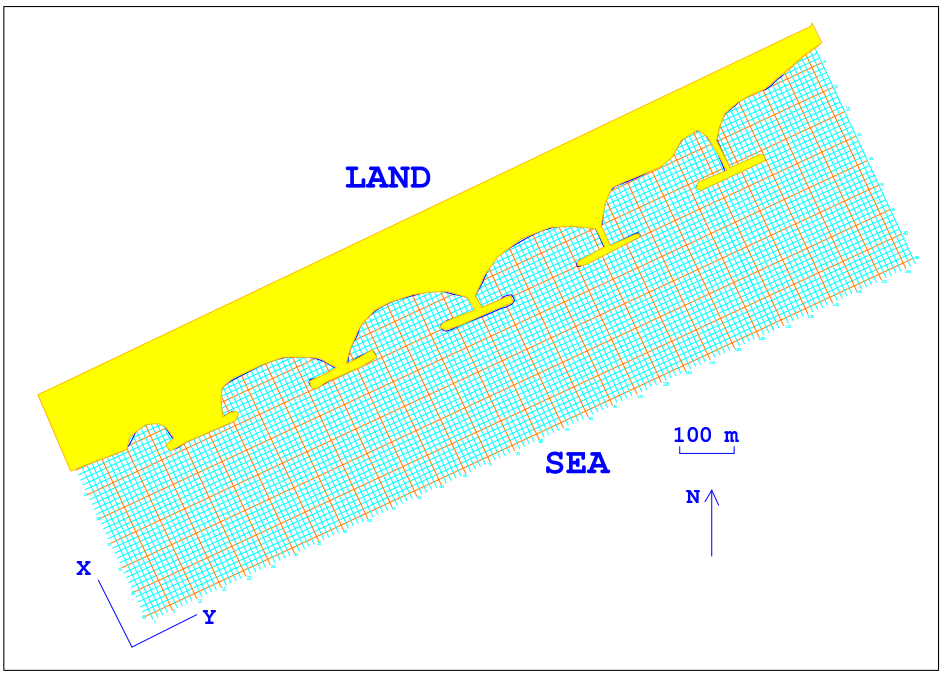


Figure 3.20: Simulation area

Table 3.4: Average wave height, period and number of waves for different directions, measured at 17 m depth contour offshore Haihau, Namdinh (Hung et al., 2001)

	N	NNE	NE	ENE	E	ESE	SE	SSE	S	Calm
H_{m0} (m)	1.22	1.60	2.05	1.47	1.35	1.15	1.54	1.99	1.73	-
T(s)	5.0	6.0	7.0	6.0	6.0	6.0	6.0	8.0	7.0	-
NR	1604	3211	7962	1466	3054	972	1850	1415	4922	2028

A map of the area is available with the scale 1:500. It covers a coastline length about 1800 m. The width of surfzone to be measured on the map is only 100 m, corresponding to a water depth of 2 m (in case of LW).

It can be seen that before the dike was built, the land has an elevation of around 1.5 m (depths of map are in + m CD). However, since we do not have bathymetry data of the site before groins were built, only simulation of flow pattern are performed instead of modelling the retreat process of shoreline.

Groins are made of big concrete structures, typically 1.5 m to 2 m above CD, with a length of 55 m, width of T-bars 60 m, and crest width of 2 m. The space between two consecutive groins is 275 m. Without the protection of groins, the land behind with elevation from +1 to 1.5 m CD would be eroded by longshore currents, and especially, by waves in severe storms.

The coast orientation is 65° w.r.t. North. We superimposed a grid consisting of 45 × 157 cells, with cell size of 5 m × 5 m. The x-axis points landwards, and the y-axis tends to go along the dike route. By defining this grid, the coastline is nearly straight compared to y direction.

The foreshore slope is quite mild, and applying the cut cell method for the shoreline is not feasible. However, for the groins where the slope of these constructions is substantial, we can assign them with solid cell type. The flow fraction between these cells and surrounding flow cells are set to 0; which means that the boundary between the groins and water is fixed; groins are never eroded (Fig. 3.21).

Wave data

For long term wave statistics, observed data (Table 3.4) is available at depth contours 17 m offshore Haihau (Hung et al., 2001).

For testing purpose we simulate the flow pattern in several simple cases where we assume a steady hydrodynamic boundary. For waves we choose two scenarios:

- Waves coming from the South with $H_{rms} = 1.2$ m (corresponding to $H_{m0} = 1.73$ m in Table 3.4 at the local depth $h = 17$ m). The wave incident angle is $\theta_0 = 25°$ and wave period is $T = 7.0$ s.

- Waves coming from the East with $H_{rms} = 0.93$ m (corresponding to $H_{m0} = 2.05$ m in Table 3.4 at the local depth $h = 17$ m); $\theta_0 = -65°$ and $T = 6.0$ s.

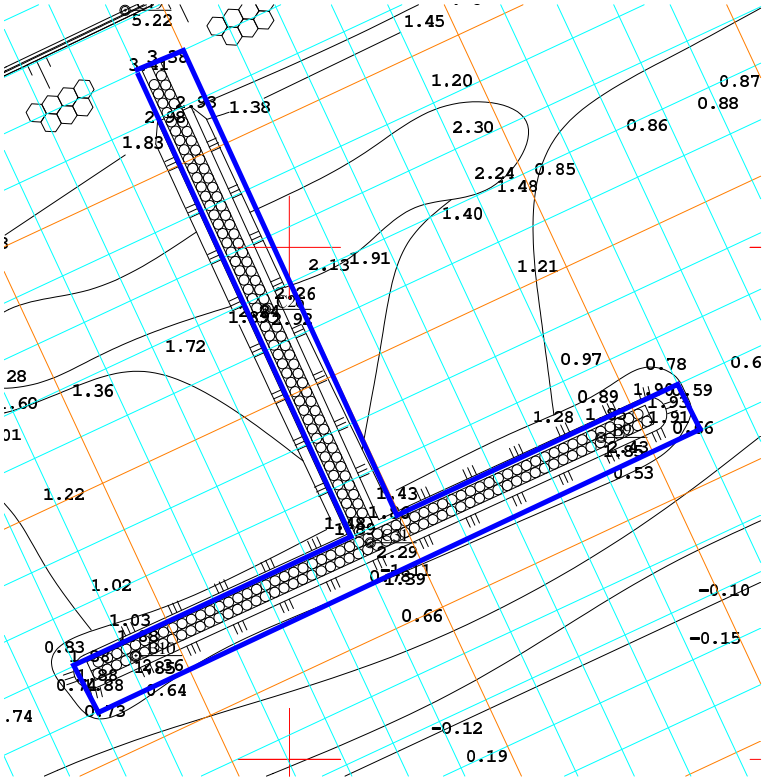


Figure 3.21: Flow fractions between groins and surrounding flow cells (marked with thick lines) are set to zero.

3.4.4 Results

Fig. 3.22 shows that wave height increases shorewards, which resembles the wave shoaling phenomenon. The pattern of water surface elevation shows set-down and set-up of water surface level. Velocity in y-direction, as the main component of longshore current, directs from South to North in the computational co-ordinate system, with highest value reaching 2 m/s.

Fig. 3.23 shows flow field around the groins in different scenarios.

Fig. 3.24 presents the case with low water, in which the longshore current is bended near the head of each groin, along with rip currents.

Through the example of simulating flow around the groins in Haihau, we can see that XBeach can handle quite complex geometry. In principle, we can couple the cut cell method inside a conventional whole cell XBeach model and make use of the new hydrodynamic module with the existing `transus/bed_update` modules. Therefore, it is hopefully that XBeach can be applied with real applications.

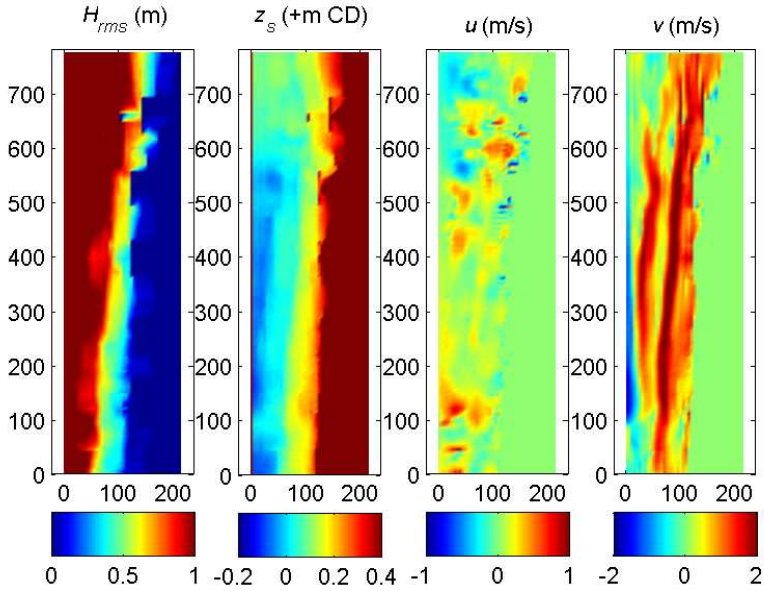
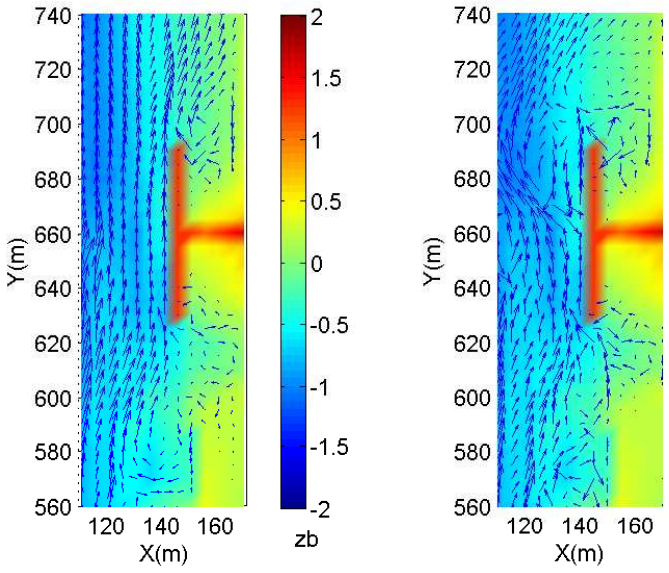


Figure 3.22: Result of simulation with wave from the South and low water at $t = 300$ s



(a) South, LW (b) West, LW

Figure 3.23: Simulated flow field in different wave and water level scenarios.

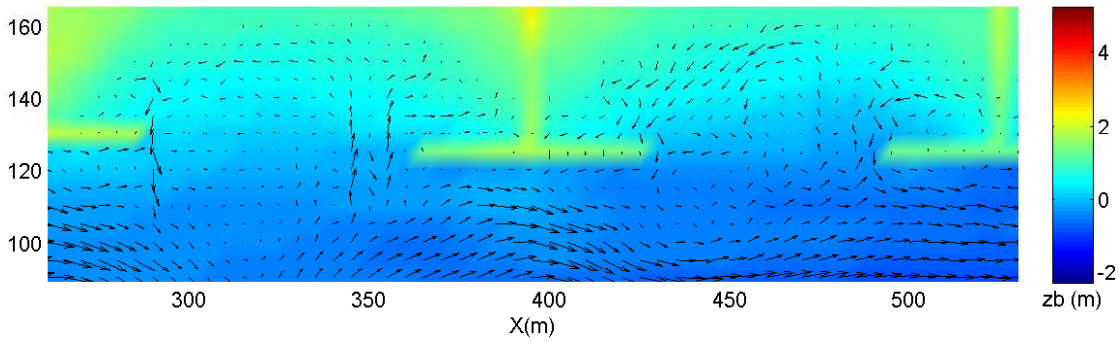


Figure 3.24: Flow pattern among the groins at $t = 300$ s

Chapter 4

Conclusion

4.1 Discussion on result

In this study, we have examined a method of simulating the flowfield near a vertical bankline in a 2-D model. The retreat process of bankline is computed in the 1-D case in coupling with the hydrodynamic-sediment transport process. For 2-D representation of shoreline, hydrodynamic tests were done using cut cell method.

1. Calculation on the retreat of a shoreline in a predefined direction show stable result, even in case of shoreline with a hump and the retreat distance is significant. It means that when feedback to hydrodynamic condition is substantial, the model continues to work without producing spurious results. An issue to be improved is calculating the retreat rate near the boundary.

2. Hydrodynamic tests on cut cell has shown better result compared to the whole cell calculation. Not only is the boundary more correct geometrically but we can also get more accurate flow field near the bank. Assuming a free-slip boundary condition and by adjusting descretisation part of advection term, we get velocity vector nearly tangential to the shoreline. However, the magnitude of these velocity should further be examined.

3. A test case for Haihau coast, Vietnam shows that XBeach can operate with complex topography. Although the cut cells are not presented in the test case, it would work in principle and give more realistic results.

4.2 Further study

Although reasonable solutions are made during computation, there are still many features which can be done to improve the model:

1. *More proper treatment for advection term*: Currently used upwind method helps avoid employing a Riemann solver; however the method is only first-order accurate. In the

test of diagonal channel (Sect. 3.3.1), different flow velocities were found nearby two banks, from which we can question the effect of upwind method, since the type of upwind cells are different in these banks.

2. *Validate the model by further tests / Calibrate the model using measured data*: We need to validate the performance of hydrodynamic calculations in various cases, preferrably with analytical solution.

3. *Account for waves, wind shear stress, Coriolis force*: Different forces can be treated in the momentum equations in a similar manner to currently accounted components, e.g. water surface slope and friction.

4. *(Semi-)implicit scheme or using Runge-Kutta method*: The restriction in time step of an explicit model may require excessive computational resource. In our tests, the running time for the diagonal channel (with only $100 \times 55 = 5500$ cells) is even longer than that of the bent channel ($80 \times 80 = 6400$ cells), due to the existence of some cells with tiny size leading to very small Δt. Implicit schemes can help overcome this difficulty.

5. *Grid refinement*: Usually the cut cell method does not provide sufficient accuracy if the shoreline bends too sharply. We already see that water depth distribution in the case of bent channel is not as good as straight channel. Refinement methods, which reduces cell size locally in the zone of high gradient, is a solution to the problem. Computational aspects of refinement can be found in e.g. (Wackers, 2003).

6. *Account for different types of shoreline*: We only considered the case where the bank is vertical and therefore the top and toe of bank coincide, where shoreline can represent as a single polyline. If the shore is considered as a sloped zone, we may need to have different polylines for bank top and bank toe (Mosselman, 2005).

Appendix A

Overview of XBeach model

A.1 Basic characteristics of a coastal morphodynamic model

In pure hydrodynamic models, only flows are simulated and partial derivation equations (PDEs) are solved using either an explicit scheme or implicit scheme. In these cases, the bottom (bed) of channel or coastal body is considered to be fixed, i.e.

$$\frac{\partial z_b}{\partial t} = 0$$

Thus the bed is considered "closed boundary", along with the "open" boundaries.

Hydro-morphological models, on the other hand, simulate the variation of bed levels. So along with the hydrodynamic process, the elevation of bed, (z_b) changes. This change in turn affects the flow condition. Thus, hydrodynamic and morphologic processes interact with each other.

A common way to think in developing a hydro-morphological model is that after calculating the flow field, the rate of sediment transport (S) is calculated. Then using the sediment balance, it is able to estimate the change in bed elevation (Δz_b) within a time step (Δt). So the new bed elevation ($z_b + \Delta z_b$) is updated, which effects the flow depth h.

In reality, the change in bathymetry is usually very small compared to change in flow. Thus it is also reasonable to update the bathymetry once every a number of flow timesteps. The "tide-averaging approach" and "RAM approach" are typical methods of such kind. They are discussed in (Roelvink, 2006).

Also in his article, Roelvink (2006) showed that there are also other effective approaches, named "online" and "parallel online", where sediment transport and bottom updating are calculated at the same time steps as the flow field. However, as there is a difference in time scale between flow and morphology, a coefficient should be considered – the "morphological factor" (Lesser et al., 2004). By using a morphological factor n, the change in bed level calculated in the model multiplied by n, which represents the morphological changes over n cycles. See Figure A.1.

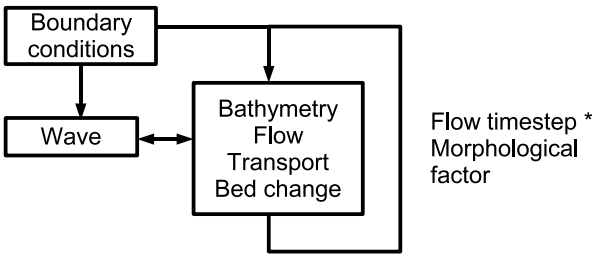


Figure A.1: Flow diagram of "online" morphodynamic model setup. (Roelvink, 2006)

This 'online' method is applied in XBeach. The use of 'online' approach has an advantage of ability to represent short-term processes which make it easy to inculde various interactions between flow, sediment and morphology (Roelvink, 2006).

A.2 Components in XBeach model

The program XBeach consists of a main MatLab script, `xbeach.m`, and a number of functions that operate on two *structures*:

- `par` – this contains general input parameters
- `s` – this contains all the arrays for a given computational domain, i.e. the "state" of system during computation

As a typical implementation in MatLab, XBeach are divided into many modules that operate on the structures `s` and `par`. A schematization of XBeach modules is described in Figure A.2.

The following sections give an overview on numerical calculation in each module of XBeach. Due to the limitation of space, we focus on the hydrodynamic module. Wave action and sediment transport are slightly mentioned.

A.3 Hydrodynamic module

A.3.1 Equations

XBeach uses shallow water equations, Coriolis and horizontal diffusion terms are neglected.

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\tau_{bx}}{\rho h} - g\frac{\partial \eta}{\partial x} + \frac{F_x}{\rho h} \tag{A.1}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\tau_{by}}{\rho h} - g\frac{\partial \eta}{\partial y} + \frac{F_y}{\rho h} \tag{A.2}$$

$$\frac{\partial \eta}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} = 0 \tag{A.3}$$

Module name	Description
`wave_input`	initializes structure `par` containing wave input parameters
`flow_input`	appends flow parameters into structure `par`
`sed_input`	appends sediment parameters into structure `par`
`grid_bathy`	creates grid and bathymetry, then store them iin structure `s`
`wave_dist`	creates initial directional spectrum at sea boundary
`flow_init`	initializes arrays (as elements of `s`) for flow computations
`sed_init`	initializes arrays (as elements of `s`) for sediment computations
`wave_bc`	wave boundary conditions update, each timestep
`flow_bc`	flow boundary conditions update, each timestep
`wave_timestep`	calculation of wave in one timestep
`flow_timestep`	calculation of flow in one timestep
`transus`	calculation of suspended transport in one timestep
`bed_update`	calculation of new bed level in one timestep

Table A.1: List of modules in XBeach program

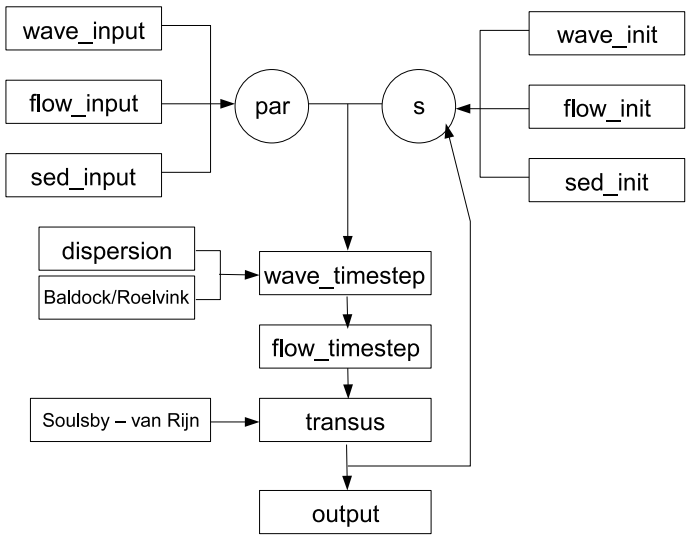


Figure A.2: Schematization of XBeach model

where h is the water depth; u, v are velocities in x, y directions, τ_{bx} and τ_{by} are bed shear stress components; g is the acceleration of gravity, η is the water level and F_x, F_y are the wave-induced stresses.

A.3.2 Discretisation

A staggered grid is applied, with bed levels (z_b) and water levels (z_s) are defined in the centre of cells, and velocities components (u, v) at the cell interfaces (the Arakawa-C scheme).

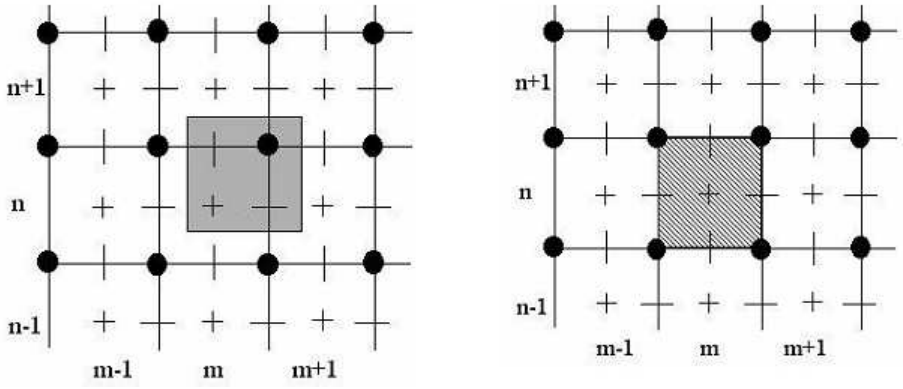


Figure A.3: Left figure: The staggered grid system used in Delft3D-FLOW. • : depth points, + : water level points, − : u-velocity points, | : v-velocity points. All the points in a highlighted squares have the same index in computation. Right figure: a computational cell is highlighted. (Delft Hydraulics, 2005).

This grid is analogous to that used in Delft3D-FLOW module, the only difference is in XBeach, the depth points and water level points are identical.

Let `nx` and `ny` be the number cells in both directions, then water level points are numbered from `1` to `nx + 1` and from `1` to `ny + 1`.

The gradient of water level between two adjacent cells are computed as:

$$\left(\frac{\partial \eta}{\partial x}\right)_{i,j} = \frac{\eta_{i+1,j} - \eta_{i,j}}{x_{i+1,j} - x_{i,j}}$$

$$\left(\frac{\partial \eta}{\partial y}\right)_{i,j} = \frac{\eta_{i,j+1} - \eta_{i,j}}{y_{i,j+1} - y_{i,j}}$$

The water depth in each cell centre is computed as:

$$h_{i,j} = \eta_{i,j} - z_{b,i,j}$$

The advection terms of equations are discretized using the *upwind method*. Since XBeach implements an explicit scheme, the upwind method is a pre-requisite for stability. Additional information can be referred in (P. W. Hemker).

For choosing depth at edges of cells, we have to cope with a discontinuity of elevation since each cell has its own depth $h_{i,j}$. If there is substantial change in elevation there would be analogous to the case of flow contraction and expansion, where Stelling and Duinmeijer (2003) presented an effecient treatment.

Principle of the upwind method for the depths in continuity equation is to apply the depth of upwind cell when u is noticeable, i.e. $|u| > u_{min}$ (where u_{min} is a predefined

minimum velocity), and the avarage depth of two cells otherwise:

$$h_{u,i,j} = \begin{cases} h_{i,j} & \text{if } u_{i,j} > u_{min} \\ h_{i+1,j} & \text{if } u_{i,j} < -u_{min} \\ \frac{1}{2}(h_{i,j} + h_{i+1,j}) & \text{if } |u_{i,j}| < u_{min} \end{cases}$$

For $u\frac{\partial u}{\partial x}$ we have:

$$u\frac{\partial u}{\partial x} = \frac{1}{2}\frac{h_{u,i,j}u_{i,j} + h_{u,i-1,j}u_{i-1,j}}{h_{mu,i,j}}\frac{u^n_{i,j} - u^n_{i-1,j}}{x^n_{i,j} - x^n_{i-1,j}} \quad \text{for } u^n_{i,j} > 0$$

$$u\frac{\partial u}{\partial x} = \frac{1}{2}\frac{h_{u,i,j}u_{i,j} + h_{u,i+1,j}u_{i+1,j}}{h_{mu,i,j}}\frac{u^n_{i+1,j} - u^n_{i,j}}{x^n_{i+1,j} - x^n_{i,j}} \quad \text{for } u^n_{i,j} < 0$$

$$v\frac{\partial u^n}{\partial y} = v^n_{u,i,j} \cdot \frac{u^n_{i,j+1} - u^n_{i,j-1}}{y^n_{i,j+1} - y^n_{i,j-1}}$$

Similar discretization applies for $v\frac{\partial v}{\partial y}$ and $u\frac{\partial v}{\partial x}$.

The momentum equations are discretized as:

$$\frac{u^{n+1}_{i,j} - u^n_{i,j}}{\Delta t} = -u\frac{\partial u^n}{\partial x_{i,j}} - v\frac{\partial u^n}{\partial y_{i,j}} - \frac{g u^n_{i,j}\sqrt{{u^n_{i,j}}^2 + {v^n_{u,i,j}}^2}}{h^n_{u,i,j}C^2} - g\frac{\eta^n_{i+1,j} - \eta^n_{i,j}}{x_{i+1,j} - x_{i,j}} + \frac{F_{x,i,j}}{\rho h_{u,i,j}} \tag{A.4}$$

$$\frac{v^{n+1}_{i,j} - v^n_{i,j}}{\Delta t} = -v\frac{\partial v^n}{\partial y_{i,j}} - u\frac{\partial v^n}{\partial x_{i,j}} - \frac{g v^n_{i,j}\sqrt{{u^n_{v,i,j}}^2 + {v^n_{i,j}}^2}}{h^n_{v,i,j}C^2} - g\frac{\eta^n_{i+1,j} - \eta^n_{i,j}}{y_{i,j+1} - y_{i,j}} + \frac{F_{y,i,j}}{\rho h_{v,i,j}} \tag{A.5}$$

After computing velocities $u^{n+1}_{i,j}$ and $v^{n+1}_{i,j}$ at the new time step, the water level is then updated basing on the continuity equation (A.3). The problem is the discretized value of h in equation.

$$\frac{\eta^{n+1}_{i,j} - \eta^n_{i,j}}{\Delta t} = -\frac{u^{n+1}_{i,j}h^n_{i,j} - u^{n+1}_{i-1,j}h^n_{i-1,j}}{x_{u,i,j} - x_{u,i-1,j}} - \frac{v^{n+1}_{i,j}h^n_{i,j} - v^{n+1}_{i,j-1}h^n_{i,j-1}}{y_{v,i,j} - y_{v,i,j-1}}$$

A.3.3 Use of Generalized Lagrangian Mean method

It is noted that in case that waves present, velocities in momentum equations are in Lagrangian form (u^L, v^L) instead of the usual Eulerian form (u^E, v^E):

$$u^L = u^E + u^S \quad \text{and} \quad v^L = v^E + v^S$$

where u^S and v^S are Stokes drift in x and y directions:

$$u^S = \frac{E_w \cos\theta}{\rho h c} \quad \text{and} \quad v^S = \frac{E_w \sin\theta}{\rho h c}$$

An exceptional use of Eulerian velocities is in bed shear stress calculation: $\tau_b = f(u^E, v^E)$.

A.4 Sediment transport module

The continuity equation for sediment reads:

$$\frac{\partial z_b}{\partial t} + \frac{\partial S_x}{\partial x} + \frac{\partial S_y}{\partial y} = 0 \tag{A.6}$$

In computation, it is discretized using a morphological factor f_{mor}:

$$\frac{z^{n+1}_{b,i,j} - z^n_{b,i,j}}{\Delta t} + f_{mor}\left[\frac{S^n_{x,i,j} - S^n_{x,i-1,j}}{\Delta x} + \frac{S^n_{y,i,j} - S^n_{y,i,j-1}}{\Delta y}\right] = 0 \tag{A.7}$$

$$\frac{\partial hC}{\partial t} + \frac{\partial hCu^E}{\partial x} + \frac{\partial hCv^E}{\partial y} + \frac{\partial}{\partial x}\left[D_h h\frac{\partial C}{\partial x}\right] + \frac{\partial}{\partial y}\left[D_h h\frac{\partial C}{\partial y}\right] = \frac{hC_{eq} - hC}{T_s} \tag{A.8}$$

A.5 Avalanching

Whenever the slope of bed exceeds a predefined critical slope m_{cr}, the avalanching process occurs. Sediment from cell with higher elevation drops to cell with lower elevation.

In computation, bed slope regarding in x-direction is estimated as:

$$\frac{\partial z_b}{\partial x} = \frac{z_{b,i+1,j} - z_{b,i,j}}{\Delta x}$$

If this slope exceeds the critical value then the change in bed elevation is:

$$\Delta z_b = \min\left(\left(\left|\frac{\partial z_b}{\partial x}\right| - m_{cr}\right)\Delta x, 0.005\right)$$

This change will be only 0.005 m by maximum. The new bed elevation are:

$$z^{n+1}_{b,i,j} = z^n_{b,i,j} + \Delta z_{b,i,j}$$

$$z^{n+1}_{b,i+1,j} = z^n_{b,i+1,j} - \Delta z_{b,i,j}$$

Appendix B

Classification of cut cells

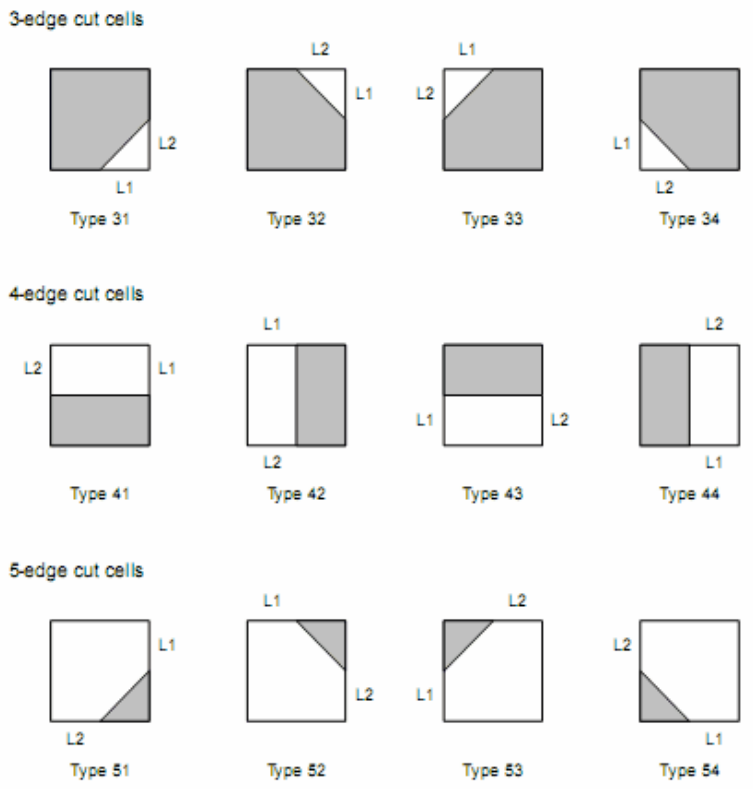


Figure B.1: Classification of cut cells (Tu and Ruffin, 2002). wet part in a cell is shaded

Appendix C

MatLab code

For 1-D case, we add a `shoreline` module to track the position of shoreline over time. This module is placed after the `flow_timestep` module and is invoked in each computational iteration.

For 2-D case, we improved the `flow_timestep` to deal with cut cell. Continuity and momentum equations are adjusted accounting for the factors Θ and δ. For the initial bathymetry, Θ and δ are prescribed in `grid_bathy`.

C.1 Hydrodynamic module (2-D)

- Additional parameters: `Th(1:nx,1:ny)` (Θ), `delta_u(1:nx,1:ny)` (δ_w or δ_e); `delta_v(1:nx,1:ny)` (δ_n or δ_s)
- Additional state variables: [None]

C.2 Shoreline module (1-D)

- Parameters: `E, G, Hc`
- State variables: `Dnb(1:ny)` (Δn_B), `DV`, `DAwet`, `Dzb`, `zbank`, `Hfb(1:ny)` `vmid(1:ny)`; `x(1:ny)` (x co-ordinates of shoreline).

References

A. Adcroft, C. Hill, and J. Marshall. Representation of topography by shaved cells in a height coordinate ocean model. *Monthly Weather Review*, 125(9):2293–2315, 1997.

P. D. Bates, M. S. Horritt, N. M. Hunter, D. Mason, and D. Cobby. *Numerical modelling of floodplain flow*. John Wiley & Sons, 2005.

D. Calhoun and R. LeVeque. A Cartesian grid finite-volume method for the advection-diffusion equation in irregular geometries. *J. Comput. Phys.*, 1999.

D. M. Causon, D. M. Ingram, C. G. Mingham, G. Yang, and R. V. Pearson. Calculation of shallow water flows using a Cartesian cut cell approach. *Advances in Water Resources*, 23(5):545–562, 2000.

D. M. Causon, D. M. Ingram, and C. G. Mingham. A Cartesian cut cell method for shallow water flows with moving boundaries. *Advances in Water Resources*, 24:899–911, 2001.

D. K. Clarke, M. D. Salas, and H. A. Hassan. Euler calculatons for multi-element airfoils using Cartesian grids. *AIAA*, 24(3):353–358, 1986.

P. Colella, D. T. Graves, B. J. Keen, and D. Modiano. A Cartesian grid embedded boundary method for hyperbolic conservation laws, 2004. URL `repositories.cdlib.org/lbnl/LBNL-56420`.

S. E. Darby. RIPA model description, 2001.

Delft Hydraulics, 2005. *Delft3D-FLOW user manual*, 2005.

Marc Dröge. *Cartesian Grid Methods for Turbulent Flow Simulation in Complex Geometries*. PhD thesis, Rijksuniversiteit Groningen, 2007.

J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer, 2002.

J. C. Fischenich. Channel erosion analysis and control. In W. Woessmer and D.F. Potts, editors, *Symposium Proceedings Headwaters Hydrology*. American Water Resources Association, 1989.

T. P. Flintham and P. A. Carling. The prediction of mean bed and wall boundary shear in uniform and compositely rough channels. pages 267–287. International Conference on River Regime, Wiley and Sons, 1988.

D. A. Ham. *On techniques for modelling coastal and ocean flow with unstructured meshes.* PhD thesis, TU Delft, 2006.

N. M. Hung, P. V. Ninh, M. Larson, and H. Hanson. Regional wave transformation and associated shoreline revolution in the Red River delta, Vietnam. Waves 2001, 4th international symposium on ocean waves measurements and analysis, 2001.

Hans Johansen and Phillip Colella. A Cartesian grid embedded boundary method for Poisson's equation on irregular domains. *J. Comput. Phys.*, 147:60–85, 1998.

J. P. Julian and R. Torres. Hydraulic erosion of cohesive riverbanks. *Geomorphology*, 76: 193–206, 2006.

K. M. T. Kleefsman. *Water impact loading on offshore structures - a numerical study.* PhD thesis, Rijksuniversiteit Groningen, 2005.

E. P. C. Koh, H. M. Tsai, and F. Liu. Euler solution using Cartesian grid with a gridless least-squares boundary treatment. *AIAA Journal*, 43(2), 2005.

E. W. Lane. Progress report on studies on the design of stable channels of the Bureau of Reclamation. *Proc. Am. Soc. Civ. Eng.*, 79:246–261, 1953.

E. W. Lane. Design of stable channels. *Transaction ASCE*, Vol. 120, 1955.

G. R. Lesser, J. A. Roelvink, J. A. T. M. van Kester, and G. S. Stelling. Development and validation of a three-dimensional morphological model. *Coastal Eng.*, 51:883–915, 2004.

C. G. Mingham. *Advanced Numerical Methods for Coastal Hydrodynamics*, pages 73–91. Elsevier, 2003.

E. Mosselman. *Mathematical modelling of morphological processes in rivers with erodible cohesive banks.* PhD thesis, TU Delft, 1992.

E. Mosselman. Bank erosion in Delft3D. Technical report, Delft Hydraulics, 2005.

Namdinh Sea Dike Service Department, 2001, 2001.

B. A. O'Connor. *Three-dimensional sediment transport model*, pages 199–214. Chapman and Hall, 1994.

J. O'Rourke. *Computational Geometry in C.* Cambridge University Press, 1994.

P. W. Hemker. Numerical hydrodynamics or computational fluid dynamics – Lecture notes. URL `citeseer.ist.psu.edu/484278.html`.

T. M. Parchure and A. J. Mehta. Erosion of soft cohesive sediment deposits. *J. Hydraul. Eng.*, 111(10):1308–1326, 1985.

E. Partheniades. Erosion and deposition of cohesive soils. *Journal of the Hydraulics Division, Proceedings ASCE*, 91(1):105 – 139, 1965.

Richard B. Pember, John B. Bell, Phillip Colella, William Y. Crutchfield, and Michael L. Welcome. An adaptive Cartesian grid method for unsteady compressible flow in irregular regions. Technical report, Lawrence Livermore National Laboratory, U.S., 1993.

James J. Quirk. A Cartesian grid approach with hierarchical refinement for compressible flows. Technical report, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, 1994.

J. A. Roelvink. Coastal morphodynamic evolution techniques. *Coastal Eng.*, 53:277 – 287, 2006.

G. Rosatti, D. Cesari, and L. Bonaventura. Semi-implicit, semi-Lagrangian modelling for environmental problems on staggered Cartesian grids with cut cells. *J. Comput. Phys.*, 204(1):353–377, 2005.

W. Shyy, H. S. Udaykumar, M. M. Rao, and R. W. Smith. *Computational Fluid Dynamics with Moving Boundaries*. Taylor & Francis, 1996.

G. S. Stelling and S. P. A. Duinmeijer. A staggered conservative scheme for every Froude number in rapidly varied shallow water flows. *Int. J. Numer. Meth. Fluids*, 43:1329–1354, 2003.

Dang Quang Thanh. Bank erosion and long-term morphodynamic equilibrium in alluvial estuaries. Master's thesis, UNESCO-IHE, 2006.

Y. H. Tseng and J. H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *J. Comput. Phys*, 192:593–623, 2003.

Shuang-Zhang Tu and Stephen M. Ruffin. Solution adaptive, unstructured Cartesian-grid methodology for chemically reacting flow, 2002.

R. J. R. van der Meûlen. The immersed boundary method for (2D) incompressible Navier-Stokes equations. Master's thesis, TU-Delft, 2003.

Leo C. van Rijn. *Principle of Sediment Transport in Rivers, Estuaries, Coastal Seas and Oceans*. North-Holland Publications, 1993.

T. T. Vinh, G. Kant, N. N. Huan, and Z. Pruszak. Sea dike erosion and coastal retreat at Nam Ha province, Vietnam. In *Coastal Engineering*, 1996.

J. Wackers. An adaptive-gridding solution method for the 2D unsteady euler equations. Technical report, CWI-Netherlands, 2003.

J. C. Winterwerp and W. G. M. van Kesteren. *Introduction to the physics of cohesive sediment in the marine environment.* Elsevier, 2004.

T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries. *J. Comput. Phys.*, 156:209–240, 1999.

Yonghui Zhu. *Breach growth in clay-dikes.* PhD thesis, TU Delft, 2006.