UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

PROGRAMA DE MAESTRÍA Y DOCTORADO EN INGENIERÍA
INGENIERÍA CIVIL – HIDRÁULICA

MODELACIÓN DE FLUJO EN AGUAS SOMERAS PARA LA EVALUACIÓN DE INUNDACIONES

TESIS
QUE PARA OPTAR POR EL GRADO DE:
DOCTOR EN INGENIERÍA

PRESENTA:
JASSY DANISSE RIVERA SOLIS

TUTOR PRINCIPAL
DR. RODOLFO SILVA CASARÍN, INSTITUTO DE INGENIERÍA
COMITÉ TUTOR
DR. RAMÓN DOMÍNGUEZ MORA, INSTITUTO DE INGENIERÍA
DR. OSCAR FUENTES MARILES, INSTITUTO DE INGENIERÍA
DR. EDGAR GERARDO MENDOZA BALDWIN, INSTITUTO DE INGENIERÍA
DR. JASSIEL VLADIMIR HERNÁNDEZ FONTES, INSTITUTO DE INGENIERÍA

CIUDAD DE MÉXICO, MARZO, 2020

JURADO ASIGNADO:

Presidente:	DR. RAMÓN DOMÍNGUEZ MORA
Secretario:	DR. OSCAR FUENTES MARILES
1 er. Vocal:	DR. RODOLFO SILVA CASARÍN
2 do. Vocal:	DR. EDGAR GERARDO MENDOZA BALDWIN
3 er. Vocal:	DR. JASSIEL VLADIMIR HERNÁNDEZ FONTES

Lugar donde se realizó la tesis: INSTITUTO DE INGENIERÍA, UNAM

TUTOR DE TESIS:

DR. RODOLFO SILVA CASARÍN

_____

FIRMA

(Segunda hoja)

Agradecimientos

A Dios en primer lugar y por sobre todo, por su amor, gracia y misericordia sobre mi vida.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo económico otorgado (406371) durante la realización de mis estudios de Doctorado.

Al Centro Mexicano de Innovación en Energía del Océano (CEMIE-Océano) por el apoyo ecónomico y técnico otorgado para la conclusión de esta tesis.

Al Programa Erasmus Mundus EURICA por la beca de movilidad otorgada para realizar una estancia de investigación en la Universidad de Groningen, Holanda.

Al Dr. Rodolfo Silva Casarin y al Dr. Edgar Mendoza Baldwin por haberme recibido en su grupo de investigación, aun en condiciones desfavorables para mi. Agradezco su guianza, sus consejos y las lecciones aprendidas de ambos. Para mí, siempre serán un ejemplo y un referente en la ingeniería de costas y puertos.

Mi agradecimiento especial a mi director de tesis, Dr. Rodolfo Silva Casarin, por su paciencia, confianza, enseñanzas y el apoyo que siempre recibí de su parte. ¡Toda mi admiración y cariño para usted!

A mi familia por su amor y su apoyo incondicional. Gracias por su comprensión, sus oraciones y palabras de fe. Siempre han sido y serán mi mayor motivación, en especial mi princesa Denisse, de quien me perdí sus primeros años de vida.

A mis amigos y compañeros del posgrado por los buenos y no tan buenos momentos compartidos. Gracias por sus palabras de ánimo cada vez que hacían falta, las tardes de café y té, los cumpleaños, etc. En especial a las niñas: Graciela, Itxaso, Pamela, Maday, Debora y Ana, las quiero y sepan que de cada una me llevo un recuerdo especial.

A mis compañeros de aventuras en el posgrado, Cesia y Joseph, les aprecio y les quiero demasiado. Gracias por su apoyo durante el tiempo que compartimos y aun ahora a distancia, a pesar de sus tantas ocupaciones. Podría decir muchas cosas pero solo les diré que gracias por aguantarme y por tanto que recibí de parte de ustedes. Atesoraré todos esos momentos como parte de las cosas bellas que viví durante el doctorado.

A mis amigos de Shalom México por su amistad incondicional, por siempre escucharme y unirse conmigo en oración ante el Padre. Les quiero y estaré por siempre agradecida por tanto amor de su parte.

A Aline y su mamá por su apoyo en el momento que más lo necesitaba. Gracias por la generosidad y afecto que me brindaron durante el corto tiempo que coincidimos.

Al Posgrado de Ingeniería y el Instituto de Ingeniería de la Universidad Nacional Autónoma de México por haberme dado la oportunidad de formar parte de su gran comunidad educativa.

Al pueblo de México por haberme acogido en este hermoso país de gente tan noble. ¡México lindo y querido, te amaré por siempre!

Contents

List of Figures

List of Tables

List of Symbols

Symbol	Description	Units
c	Wave celerity	$[m/s]$
$\widetilde{c}$	Mean wave celerity	$[m/s]$
c_f	Friction coefficient	
c_w	Wind drag coefficient	
$d,\ \eta$	Free surface elevation	$[m]$
$\bar{\eta}$	Mean free surface elevation	$[m]$
F_i	Body forces	
$\mathbf{F}$	Vector of fluxes on a global coordinate system	
$\widehat{\mathbf{F}}$	Vector of projected fluxes on a local coordinate system	
g	Acceleration of gravity	$[m/s^2]$
h	Water depth	$[m]$
$\widetilde{h}$	Mean water depth	$[m]$
$h_{_L}$	Water depth at the left side of the intercell boundary	$[m]$
$h_{_R}$	Water depth at the right side of the intercell boundary	$[m]$
$h_{_L}^{-}$	Water depth at the inner region for a negative middle wave	$[m]$
$h_{_R}^{+}$	Water depth at the inner region for a positive middle wave	$[m]$
$\overline{H}^{-}$	Rankine-Hugoniot jump condition for a negative middle wave	
$\overline{H}^{+}$	Rankine-Hugoniot jump condition for a positive middle wave	
$\widehat{\widehat{\mathrm{H}}}$	Averaged value of the water depth for the thrust calculation	$[m]$
ν	Kinematic viscosity	$[m^2/s]$

Symbol	Description	Units
p	Pressure	$[Pa]$
P_a	Atmospheric pressure	$[Pa]$
ρ	Water density	$[kg/m^3]$
ρ_a	Air density	$[kg/m^3]$
S_{ij}	Rate of strain tensor on a face whose outward normal vector points toward axis i, in the direction of axis j	$[1/s]$
$\overline{S_{ij}}$	Depth averaged rate of strain tensor	$[1/s]$
$\mathbf{S_n}$	Vector of source terms in the normal direction	
S_n	Components of the vector of source terms in the normal direction	
$\widehat{\mathbf{S}}$	Vector of projected source terms	
S_z	Thrust contribution to the source term	
S_τ	Bed shear stress contribution to the source term	
S_w	Wind stress contribution to the source term	
t	Time	$[s]$
τ_{b_i}	Shear stress at the bed	$[Pa]$
τ_{η_i}	Shear stress at the free surface	$[Pa]$
x, y, z	Length in the x-,y-,z- direction on a Cartesian coordinates system (it is also expressed as x_i in index notation)	$[m]$
u	Instantaneous velocity in the x-direction	$[m/s]$
U	Depth averaged velocity in the x-direction	$[m/s]$
$\widetilde{\mathbf{U}}$	Vector of conserved variables	
$\widehat{\mathbf{U}}$	Vector of projected conserved variables	
$\hat{u}$	Normal component of the depth averaged velocity	$[m/s]$
$\bar{u}$	Mean depth averaged velocity	$[m/s]$
$\widehat{u}$	Roe averaged velocity	
$\widehat{u}_{min}$	Minimum Roe averaged velocity	
$\widetilde{u}$	Cross section averaged velocity	$[m/s]$
u_{wn}	Normal component of the wind velocity	$[m/s]$

Symbol	Description	Units
u_w	Wind velocity	$[m/s]$
v	Instantaneous velocity in the y-direction	$[m/s]$
$\hat{v}$	Tangential component of the depth averaged velocity	$[m/s]$
V	Depth averaged velocity in the y-direction	$[m/s]$
$\mathbf{V}$	Control volume	$[m^3]$
w	Instantaneous velocity in the z-direction	$[m/s]$
w_1, w_2, w_3	Components of the vector of conserved variables	
z_b, b	Bed level elevation	$[m]$

Abstract

Every time turns out to be more evident the need to count on reliable tools to predict accurately the extent of inundations, as they have become more frequent and more extreme. This thesis presents the development of a numerical model based on the HLLCS solver with a novel multi quad-tree mesh for the study of floods assessment.

The meshing has the advantage to be suitable for rectangular domains as those are frequent in coastal environments. Apart from the mesh, a different formulation of the thrust term is tested together with the HLLCS solver. This expression for the thrust is based on the already existing divergence formulation, but it uses a different volume of integration given the fact that the solver includes the presence of the source term in the solution.

The resulting solver is assessed through different tests: theoretical, experimental and real test cases. The results show that the model predicts accurately the still water levels and the velocity field in those shallow flows where the reflection effects are not relevant. Moreover, the wetting and drying fronts are adequately tracked without negative implications of this new thrust term formulation.

Resumen

Cada vez resulta ser más evidente la necesidad de contar con herramientas que permitan predecir de forma precisa el alcance de las inundaciones, dado que estas se han vuelto cada vez más frecuentes y más extremas. Esta tesis presenta el desarrollo de un modelo numérico basado en el solver HLLCS, en conjunto con un nuevo mallado tipo multi quad-tree para el estudio de inundaciones.

El mallado tiene la ventaja de ser apropiado para dominios rectangulares como aquellos frecuentes en ambientes costeros. Además del mallado, una formulación diferente para el término del empuje se examina en conjunto con el solver HLLCS. Esta expresión para el empuje está basada en la ya existente formulación de la divergencia, sin embargo se utiliza un volumen de integración diferente dado que el solver incluye la presencia del término fuente en la solución.

El modelo resultante es evaluado apartir de diversos ensayos: casos de prueba teóricos, experimentales y reales. Los resultados demuestran que el modelo predice adecuadamente los niveles de superficie libre y campos de velocidad en flujos de aguas someras en los que los efectos de reflexión no son relevantes. Además, los frentes seco-mojado se localizan adecuadamente sin que esta nueva formulación del empuje tenga alguna repercusión.

Insights into the Solver

The solver here presented is based on the HLLCS scheme, developed by Murillo and García-Navarro (29) for the numerical solution of hyperbolic systems of partial differential equations. The HLLCS is a first order finite volume scheme that solves a Riemann Problem at every intercell boundary in order to find a solution.

The solution of the Riemann Problem is given by the superposition of left, right and middle waves, and in the case of the HLLCS, an extra stationary shock wave at the intercell boundary due to the presence of source terms. Thus the inner region is split into into three different states plus the two sates at the exterior sides of the left and right waves, respectively. The inner states are calculated, depending on the sign of the middle wave, by defining Rankine-Hugoniot conditions across the jumps.

The solver in this thesis differs from that of Murillo and García-Navarro (29) in how the geometric source term, thrust exerted over the bed, is defined. While Murillo and García-Navarro (29) calculates this term by assuming that it depends only on the free surface level at the lowest side of the interface, here the divergence formulation is used (45).

Finally, the time integration in the scheme is explicit thus the time step is subject to the well known Courant-Friedichs-Lewy condition.

Chapter 1

Introduction

The Navier-Stokes equations (NSEs) are a set of non-linear partial differential equations, which describe the motion of the fluids in $\Re^n$ ($n = 3$). Due to the complexity of such equations, for most engineering applications it is usually required and accepted to make simplifying assumptions that allow one to find a solution. One of these simplifications are the - so called - shallow water equations (SWEs), which allow to describe the motion of fluids in two dimensions for regions where the horizontal characteristic length is much larger than the vertical one.

The main assumption of the SWEs is that the vertical accelerations are negligible and so, the third equation of momentum conservation reduces to the hydrostatic law of pressures. Among the wide variety of phenomena that can be modelled by the SWEs, one could mention: hydrodynamics of lakes and estuaries (22), sediments transport (15, 25), atmospheric flows (21, 46), pollutant transport (3, 9, 26), river and coastal floods (2, 6, 22), etc.

Although the original set of three equations, for the conservation of momentum, is reduced to two equations, the SWEs still remain their non linearity and except for some ideal situations, it is usually only possible to find numerical solutions for them. Numerical modelling is, then, an essential tool in fluid mechanics which not just allows to find approximate solutions for many practical applications, but also make easier to analize complex and diverse scenarios without requiring of physical modelling, which is usually much more expensive than the numerical one.

1.1 Background

Two alternative approaches can be distinguished according to what kind of space coordinate system we adopt. The equations of motion can usually be obtained using either a **Lagrangian** frame of reference, material description of fluid flow, or an **Eulerian** frame of reference, space description of fluid flow (42).

1. The Lagrangian perspective follows the trajectory of each fluid particle within the domain, i.e. it uses a moving system of coordinates and its object of study is a given fluid particle.

The boundary of the domain is defined by the fluid particles and thus it is never crossed by them. Here the new space coordinates - Lagrangian coordinates - are functions of the independent variables: time and the initial position of all the particles.

2. The Eulerian perspective uses a fixed system of reference and thus its object of study is the flow at a given position or in a fixed region. This approach restricts the study of the flow to a certain control volume in which the individual position of the particles is not considered. Moreover the basic quantities of the flow are rather determined as a function of the independent variables: time and spatial coordinates.

Figure 1.1 shows an attempt to arrange the existing numerical methods. Although there is no a standard classification and there can be an extensive number of methods, it shows a good summary and it is useful to provide the reader a general background on the different methods.

In the same way the equations of motion can be obtained by two different coordinate systems, the methods can be sorted by their spatial discretization in Eulerian, Lagrangian or a combination of these two. If compared to the Eulerian methods, Lagrangian ones have been used with less frequency for the simulation of free surface flows (12), altough they have been widely used in other disciplines such as solid mechanics (13). According to Tennekes and Lumley (43), in the past only Eulerian measurements of the modelled systems were carried out and its transformation to Lagrangian coordinates is not trivial. In general, Lagrangian methods allow to solve precisely free surface and material boundaries, and they also provide and adequate treatment for rigid boundaries with an arbitrary shape (13). Nevertheless, they are yet bad for treating large distortions of the computational domain without recurring to frequent remeshing operations (16). Eulerian methods, even they have no difficulties due to mesh distortion, additional complexity is added for the treatment of the convective terms using of a fixed system of reference.

Since the Eulerian approach is used in this thesis, special attention is given to the methods in this category. The classical Eulerian methods can be grouped into Finite Differences, Finite Volume (FV) and Finite Element (FEM). Finite differences method uses Taylor series approximations of the strong form of the equations. The classic alternative methods (FV and FEM), on the opposite, make use of the weak form of the equations, and they often prove to be more versatile when dealing with multidimensional domains of complex geometry (31). Recently, another type such as the Spectral Methods and Boundary Element Methods have also emerged, but these are only mentioned here. As for the time discretization, schemes can be either explicit, implicit or the so called semi-implicit, which is a combination of the first two. Explicit schemes use the previous time step solution to determine all the flow quantities at the current time step. The implicit approach, on the other hand, gives rise to systems of equations that must be solved simultaneously since the solution depends not only on the previous time step, but also on the current time step itself. Although the explicit schemes require a lower computational cost, they are subject to a time limitation related to the rate of propagation of gravitational surface waves (7). This restriction, known as the Courant - Friedich - Lewy (CFL) condition, makes the explicit schemes to be stable only under certain combinations of time steps and spatial steps.

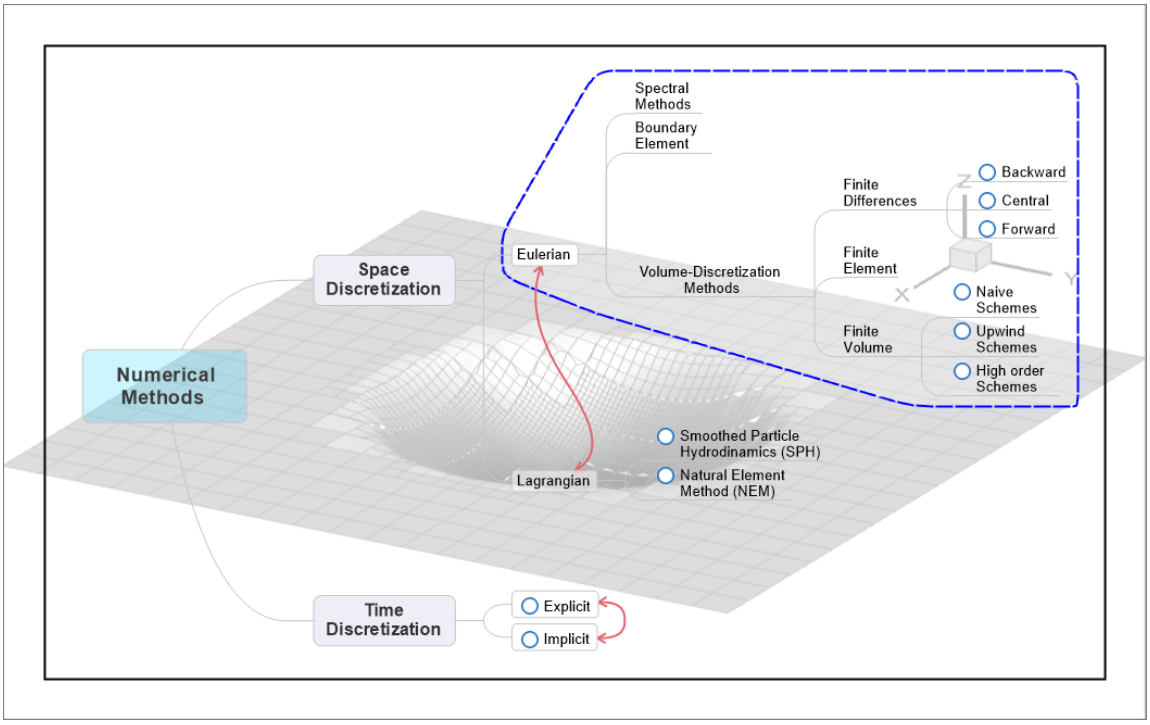


Figure 1.1: Classification of numerical methods by space and time discretization scheme

In this thesis an explicit finite volume scheme based on approximate Riemann solvers will be used for the solution of the SWEs and more specifically, for the assessment of inundations.

1.2 Problem Statement

The term inundation, i.e. flood, refers to the processes of rising and spreading water over lands that are supposed either to be dry or to have a lower water level. They can occur as a natural process such as in case of a storm, or as a consequence of man actions.

Inundations are a topic of major concern in hydraulic engineering. For instance, floods are one of the most frequent water-related disasters (see Fig. 1.2) and the one that causes more economic loss (30). In fact, according to a new study (20), the frequency of powerful cyclones in the US trebled over the last century, at the time they have become more intense. Inundations are also important in coastal environments such as an estuary, where the interaction between the sea and inland waters by means of tidal waves, gives place to very rich and productive ecosystems.

Against this background, it becomes necessary to count on practical tools to effectively assess inundations not only in their extent but also the characteristics of the flow itself. Two main aspects that influence the accuracy of a numerical solution are addressed in this thesis. These are the mesh

resolution and the performance of the solver as well. In the case of inundations, more complexity arises from tracking the wetting and drying fronts.

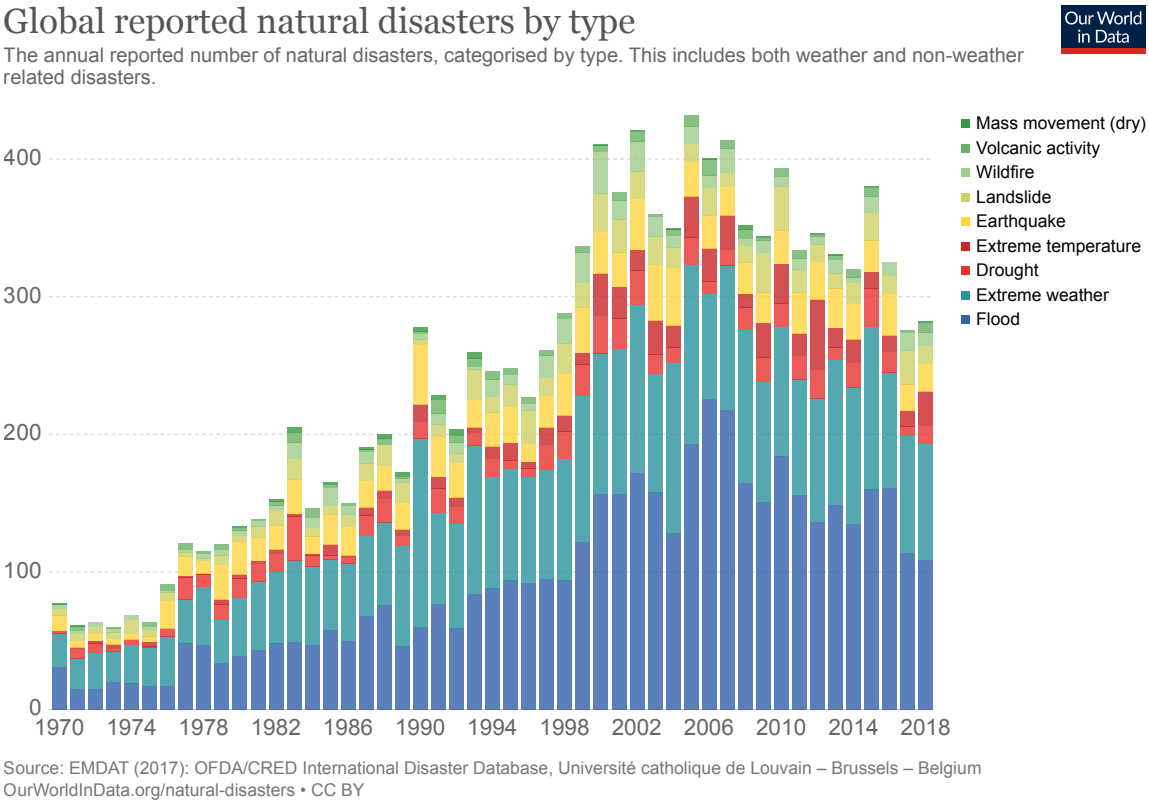


Figure 1.2: Global reported natural disasters by type from 1970 - 2018.
Source: Taken from Ritchie and Roser (33).

Although there are different numerical models that can resolve inundations, there is no general agreement on which model should be regarded as the best one given that every model has both strengths and weaknesses. While some models are commercial and thus they are not accessible to all, open source models have other disadvantages associated such as limited support or the absence of a graphical unit interface (GUI). In any case it is still required to continue developing efficient numerical tools that can be capable of modelling inundations without lose practicality.

1.3 Scope of the works

The main goal is to develop an integral numerical model for floods assessment which can be capable to accurately estimate the wetting and drying areas in regions of complex hydrodynamics where the influence of different source terms (e.g. wind stress, bed friction, Coriolis forces, etc.) can be effectively accounted.

1.3.1 Major Objective

The numerical model is to be achieved by implementing an already existing finite volume scheme, the HLLCS (29), in combination with a new proposal for creating a multi quad-tree type mesh and a different treatment for the bed slope source term.

1.3.2 Particular Objectives

- To develop an algorithm for generating the multi quad-tree type mesh. This algorithm must be capable of using the principles of recursive domain decomposition in rectangular areas and at the same time of providing interpolated values for the independent variables, i.e., Cartesian coordinates of the cell centers and forcing conditions, such as tide and wind.
- To perform laboratory experiments that can be useful for the validation process. This is particularly an important step given the difficulty to replicate the experiments reported in the literature.
- To implement the HLLCS scheme, in its original formulation (29), for solving theoretical, laboratory and practical test cases.
- To appropriately set the source terms definition for each particular case considered.
- To test the HLLCS scheme with divergence formulation of the bed slope source term (45).
- To test the HLLCS scheme, both in its original formulation as well as with the divergence formulation, together with the proposed multi quad-tree type mesh.

Chapter 2

Theoretical Framework

2.1 The Equations

The Navier-Stokes equations are a system of equations that governs the motion of Newtonian and incompressible fluids (23, 27, 48). They are given by:

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{2.1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}(2\nu S_{ij}) + F_i \tag{2.2}$$

where the indexes i and j range over 1,2 and 3. As usual ρ and ν are the density and the kinematic viscosity of the fluid, accordingly.

Equation 2.1 is the continuity equation which is but the principle of conservation of mass applied over an infinitesimal control volume. Equation 2.2, the well known NSEs, arise from the application of Newton's second law to the motion of fluids over this control volume. The instantaneous velocity field is described by u, p denotes the pressure, S_{ij} is the strain-rate tensor and t, the time. The last term of the right-hand side, F, accounts for the action of body forces (e.g. Coriolis), if any.

Both conservation laws together give rise to a system of four equations that describes the movement of fluids in three dimensions perfectly. The existence and uniqueness of an analytical solution of Eq. 2.2 remain a challenge of mathematics (11). Fortunately, however, a two-dimensional representation is usually sufficient for most practical applications. Therefore the utmost relevance of simplified versions of this system e.g. the Shallow Water Equations (SWEs).

The SWEs elucidate the motion of "shallow flows" in two dimensions (redundancy intended). The first property of a shallow flow is that the vertical length scale (h) is much smaller than the horizontal one (L) (Eq. 2.3). This situation is often encountered in many ocean, coastal and river

problems thus they are widely applicable. Some common shallow flows are the tide propagation in a coastal lagoon or the dam break problem.

$$\frac{h}{L} << 1 \tag{2.3}$$

As a consequence, it is possible to assume that the vertical accelerations are negligible and therefore the flow is mainly two dimensional, i.e., the primitive variables do not depend on the water depth, moreover the pressure has a hydrostatic distribution. This allows to simplify the vertical momentum equation and therefore the system is reduced to three equations instead of four. The steps required to derive the SWEs, from NSEs as a starting point, are summarized next:

- Neglect the vertical acceleration, i.e., assume a hydrostatic pressure distribution. The vertical momentum equation reduces to the hydrostatic pressure equation (Eq. 2.4). P_a stands for the atmospheric pressure.

$$\partial p/\partial z \approx -\rho g \implies p(z) = \rho g\,(\eta - z) + P_a \qquad for\ z_b \leq z \leq \eta \tag{2.4}$$

- Compute the horizontal pressure gradients with Eq. 2.4.

$$\frac{\partial p}{\partial x_i} = \rho g \frac{\partial \eta}{\partial x_i} + \frac{\partial P_a}{\partial x_i} \qquad (i = 1,2) \tag{2.5}$$

- Substitute Eq. 2.5 into the two remaining momentum equations (Eq. 2.2). This yields:

$$\begin{aligned} \frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) + \frac{\partial}{\partial z}(u_i w) = &- g\frac{\partial \eta}{\partial x_i} - \frac{1}{\rho}\frac{\partial P_a}{\partial x_i} + \frac{\partial}{\partial x_j}(2\nu S_{ij}) \\ &+ \frac{\partial}{\partial z}(2\nu S_{i3}) + F_i \qquad (i = 1,2) \end{aligned} \tag{2.6}$$

- Integrate the horizontal momentum equations and the continuity equation over the depth (Eqs. 2.1 and 2.6). The limits of integration are given by the bed vertical coordinate, z_b, and the free surface position, η, as depicted in Fig. 2.1.
- Apply the Leibnitz's rule for differentiation under the integral sign.
- Define the depth averaged velocities as:

$$U_i = \frac{1}{h}\int_b^\eta u_i\, dz \qquad (i = 1,2)$$

- Impose the boundary conditions (BCs): non slip at the bed, no relative normal flow at the free surface (kinematic BC) and at the bed.

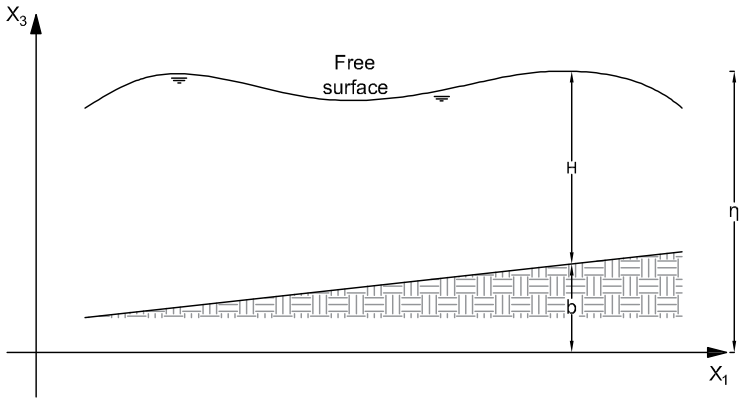


Figure 2.1: Definition of variables for the derivation of the SWEs. The free surface level is given by the sum of the bed elevation and the water head ($\eta = h + z_b$).

After the former assumptions and completion of the steps, the resulting non-linear shallow water equations are given by:

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x_i}(hU_i) = 0 \tag{2.7}$$

$$\frac{\partial}{\partial t}(hU_i) + \frac{\partial}{\partial x_j}(hU_iU_j) = -\frac{1}{2}g\frac{\partial h^2}{\partial x_i} - gh\frac{\partial z}{\partial x_i} - \frac{1}{\rho}\frac{\partial P_a}{\partial x_i} + \frac{\partial}{\partial x_j}\left(2\nu h\overline{S_{ij}}\right) + \frac{1}{\rho}(\tau_{\eta_i} - \tau_{b_i}) \tag{2.8}$$

where the tilde (-) indicates that the quantities are depth averaged and g is the gravitational constant, as usual. The last term in Eq. 2.8 accounts for the shear stress at the free surface and at the bottom, accordingly. A friction law, e.g. Manning or Chezy's law, is usually utilized to express the shear stress at the bottom, τ_{b_i}. While the shear stress at the free surface, τ_{η_i}, is equal to the value imposed by the wind, because of the dynamic boundary condition, and it is usually modelled using semiempirical formulas based on the module of the wind velocity.

Notice that Eq. 2.8 is only valid for non-stratified and constant density flows. Otherwise the pressure might not follow a hydrostatic behaviour, which is the main assumption in the derivation process. Although there is a lot of rigorous math involved to obtain the SWEs, it is important for engineers to have knowledge of the process since this allows to understand the limitations of the model. The complete derivation of the SWEs can be found in Appendix A.

2.2 Other Forms of the Equations

As for the SWE, also referred to hereinafter as "the equations", two more assumptions are required before going into details:

- Atmospheric pressure gradients can be neglected.

$$\frac{\partial P_a}{\partial x_i} \approx 0 \qquad (i = 1,2)$$

- Advection process is dominant over diffusion. Thus the latter can be neglected.

$$\frac{\partial}{\partial x_j}(hU_iU_j) >> \frac{\partial}{\partial x_j}\left(2\nu h\overline{S_{ij}}\right) \qquad (i = 1,2)$$

With these conditions, the SWE can be rewritten as:

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial}{\partial x_i}(hU_i) &= 0 \\ \frac{\partial}{\partial t}(hU_i) + \frac{\partial}{\partial x_j}(hU_iU_j) &= -\frac{1}{2}g\frac{\partial h^2}{\partial x_i} - gh\frac{\partial z}{\partial x_i} + \frac{1}{\rho}(\tau_{\eta_i} - \tau_{b_i}) \end{aligned} \tag{2.9}$$

By following the usual conventions of index notation (summation over repeated indexes), Eq. 2.9 can be expanded as a system of three equations. In a Cartesian coordinate system (x,y,z) with a vertical z-axis and two horizontal axes, x and y, these are given by:

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hU) + \frac{\partial}{\partial y}(hV) = 0 \tag{2.10a}$$

$$\frac{\partial}{\partial t}(hU) + \frac{\partial}{\partial x}\left(hU^2\right) + \frac{\partial}{\partial y}(hUV) = -\frac{1}{2}g\frac{\partial h^2}{\partial x} - gh\frac{\partial z}{\partial x} + \frac{1}{\rho}(\tau_{\eta_x} - \tau_{b_x}) \tag{2.10b}$$

$$\frac{\partial}{\partial t}(hV) + \frac{\partial}{\partial y}\left(hV^2\right) + \frac{\partial}{\partial x}(hVU) = -\frac{1}{2}g\frac{\partial h^2}{\partial y} - gh\frac{\partial z}{\partial y} + \frac{1}{\rho}(\tau_{\eta_y} - \tau_{b_y}) \tag{2.10c}$$

The SWE however can be written in several ways, e.g. conservative or non-conservative form, differential or integral form. Although they are all equivalent, regardless of the form they are written in, this is no longer valid in their numerical discretized form. Two forms of Eq. 2.10 are presented in this section, since they are both essential for the development of the numerical methods used in this thesis.

2.2.1 Conservation-Law Form

Equation 2.10 can be also written in compact matrix form, i.e., as a system of conservation laws. In differential form, they now read:

$$\begin{gathered}\frac{\partial \widetilde{\mathbf{U}}}{\partial t}+\frac{\partial \mathbf{F}(\widetilde{\mathbf{U}})}{\partial x}+\frac{\partial \mathbf{G}(\widetilde{\mathbf{U}})}{\partial y}=\mathbf{S}(\widetilde{\mathbf{U}}) \\ \text{Or} \\ \widetilde{\mathbf{U}}_t+\mathbf{F}(\widetilde{\mathbf{U}})_x+\mathbf{G}(\widetilde{\mathbf{U}})_y=\mathbf{S}(\widetilde{\mathbf{U}})\end{gathered} \tag{2.11}$$

with

$$\widetilde{\mathbf{U}}=\begin{bmatrix} h \\ hU \\ hV \end{bmatrix} \qquad \textbf{Vector of conserved variables} \tag{2.12a}$$

$$\mathbf{F}=\begin{bmatrix} hU \\ hU^2+\frac{1}{2}gh^2 \\ hUV \end{bmatrix} \quad \mathbf{G}=\begin{bmatrix} hV \\ hVU \\ hV^2+\frac{1}{2}gh^2 \end{bmatrix} \qquad \textbf{Flux vectors} \tag{2.12b}$$

$$\mathbf{S}=\left[0,\ -gh\frac{\partial z}{\partial x}-ghS_{f,x}\ ,\ -gh\frac{\partial z}{\partial y}-ghS_{f,y}\right] \qquad \textbf{Source terms vector} \tag{2.12c}$$

where $S_{f,x}$ and $S_{f,y}$ are the friction slopes in the x and y direction, which are used to model the shear stresses at the bottom. Notice that the flux vectors, $\mathbf{F}(\widetilde{\mathbf{U}})$ and $\mathbf{G}(\widetilde{\mathbf{U}})$, are both function of the vector of conserved variables. Unless otherwise indicated, the usage of subscripts to denote derivatives is introduced hereon.

The source terms vector (Eq. 2.12c), also referred to as forcing terms, may account for the effect of body forces such as gravity. For many engineering applications this vector will have additional terms to represent other effects e.g. vegetation stress, wind forces, Coriolis force, mass or momentum injection. In the particular case of the SWEs, $\mathbf{S}(\widetilde{\mathbf{U}})$ is also function of some of the flow variables.

2.2.2 Quasi-linear Form

The non-linearity of Eq. 2.11 is due to the fact that the product of the unknowns appear in the flux equations (see Eq. 2.12b). Differentiating between linear and non-linear systems of partial diferential equations (PDE's) is also important when developing any numerical method. Most of the theory for the solution of PDE's systems was first developed for those linear, therefore the

need to present here the quasi-linear form of the equations which also defines the structure of the solution.

As per the chain rule, and since $\mathbf{F}$ and $\mathbf{G}$ are both function of $\widetilde{\mathbf{U}}$, Eq. 2.11 can be written in quasi-linear form as:

$$\widetilde{\mathbf{U}}_t + \mathbf{A}(\widetilde{\mathbf{U}})\widetilde{\mathbf{U}}_x + \mathbf{B}(\widetilde{\mathbf{U}})\widetilde{\mathbf{U}}_y = \mathbf{S}(\widetilde{\mathbf{U}}) \tag{2.13}$$

where $\mathbf{A}(\widetilde{\mathbf{U}})$ and $\mathbf{B}(\widetilde{\mathbf{U}})$ are the *Jacobian matrices* corresponding to the flux vectors. The entries of these matrices are the partial derivatives of the elements f_i of $\mathbf{F}(\widetilde{\mathbf{U}})$ and g_i of $\mathbf{G}(\widetilde{\mathbf{U}})$ with respect to the components w_j of the vector $\widetilde{\mathbf{U}}$. Here the indexes i and j range over 1, 2 and 3, hence the matrices $\mathbf{A}(\widetilde{\mathbf{U}})$ and $\mathbf{B}(\widetilde{\mathbf{U}})$ read:

$$\begin{aligned}
\mathbf{A}(\widetilde{\mathbf{U}}) &= \frac{\partial \mathbf{F}}{\partial \widetilde{\mathbf{U}}} = \begin{bmatrix} \partial f_1/\partial w_1 & \partial f_1/\partial w_2 & \partial f_1/\partial w_3 \\ \partial f_2/\partial w_1 & \partial f_2/\partial w_2 & \partial f_2/\partial w_3 \\ \partial f_3/\partial w_1 & \partial f_3/\partial w_2 & \partial f_3/\partial w_3 \end{bmatrix} \\
\mathbf{B}(\widetilde{\mathbf{U}}) &= \frac{\partial \mathbf{G}}{\partial \widetilde{\mathbf{U}}} = \begin{bmatrix} \partial g_1/\partial w_1 & \partial g_1/\partial w_2 & \partial g_1/\partial w_3 \\ \partial g_2/\partial w_1 & \partial g_2/\partial w_2 & \partial g_2/\partial w_3 \\ \partial g_3/\partial w_1 & \partial g_3/\partial w_2 & \partial g_3/\partial w_3 \end{bmatrix}
\end{aligned} \tag{2.14}$$

In order to compute the Jacobian matrices, it is necessary to explicitly express the flux vectors in terms of the conserved variables, i.e.:

$$\left.\begin{aligned}
\widetilde{\mathbf{U}} &= \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} h \\ hU \\ hV \end{bmatrix} \\
\mathbf{F}(\widetilde{\mathbf{U}}) &= \begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = \begin{bmatrix} hU \\ hU^2 + \frac{1}{2}gh^2 \\ hUV \end{bmatrix} = \begin{bmatrix} w_2 \\ w_2^2/w_1 + \frac{1}{2}gw_1^2 \\ w_2 w_3/w_1 \end{bmatrix} \\
\mathbf{G}(\widetilde{\mathbf{U}}) &= \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix} = \begin{bmatrix} hV \\ hVU \\ hV^2 + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} w_3 \\ w_3 w_2/w_1 \\ w_3^2/w_1 + \frac{1}{2}gw_1^2 \end{bmatrix}
\end{aligned}\right\} \tag{2.15}$$

The Jacobian matrices then are given by:

$$\mathbf{A}(\widetilde{\mathbf{U}}) = \begin{bmatrix} 0 & 1 & 0 \\ c^2 - U^2 & 2U & 0 \\ -UV & V & U \end{bmatrix} \quad \mathbf{B}(\widetilde{\mathbf{U}}) = \begin{bmatrix} 0 & 0 & 1 \\ -UV & V & U \\ c^2 - V^2 & 0 & 2V \end{bmatrix} \tag{2.16}$$

Here we introduce the concept of celerity, c, which is given by:

$$c = \sqrt{gh} \tag{2.17}$$

This concept in turn allows to define the Froude number, $F = \widetilde{u}/c$, a dimensionless parameter which enables for a classification into three type of free surface flow regimes: critical ($Fr = 1$), subcritical ($Fr < 1$) and supercritical ($Fr > 1$). Here, $\widetilde{u}$ is the cross section averaged velocity along the main flow direction.

2.3 Basic Mathematical Concepts

Although the mathematical theory for PDE's is quite extensive, we only present here some of the basic concepts required in the frame of Finite Volume methods (FV).

Definition 2.3.1 (Eigenvalues). To compute the eigenvalues λ_i of a matrix $\mathbf{M}$, one needs to solve the characteristic polynomial defined by:

$$|\mathbf{M} - \lambda\mathbf{I}| = \det(\mathbf{M} - \lambda\mathbf{I}) = 0$$

where $\mathbf{I}$ is the identity matrix. This is, the eigenvalues are the roots of this polynomial.

Definition 2.3.2 (Eigenvectors). A left eigenvector $\mathbf{L}^p = [l_1^p, l_2^p, ..., l_m^p]$ of a matrix $\mathbf{M}$ corresponding to an eigenvalue λ^p is a vector satisfying:

$$\mathbf{L}^{(p)}\mathbf{M} = \lambda^p \mathbf{L}^{(p)}$$

Instead, this vector can be a right eigenvector of $\mathbf{M}$ if it satisfyes:

$$\mathbf{M}\mathbf{L}^{(p)} = \lambda^p \mathbf{L}^{(p)}$$

Here the superscripts have been used to indicate the relationship between the p-eigenvalue and its corresponding eigenvector. While the subscritps have been used to denote the m elements of the left and right eigenvectors, accordingly.

Definition 2.3.3 (Diagonalizable System). In turn this matrix $\mathbf{M}$ is said to be diagonalizable if can be expressed in terms of a diagonal matrix Λ and a matrix $\mathbf{R}$, i.e.:

$$\mathbf{M} = \widetilde{\mathbf{R}}\Lambda\widetilde{\mathbf{R}}^{-1} \;\; or \;\; \Lambda = \widetilde{\mathbf{R}}^{-1}\mathbf{M}\widetilde{\mathbf{R}} \tag{2.18}$$

Such that the diagonal elements of Λ are the eigenvalues λ^p of $\mathbf{M}$ and the m columns of $\widetilde{\mathbf{R}}$ are the right eigenvectors of $\mathbf{M}$ corresponding to every eigenvalue λ^p, thus:

$$\Lambda = \begin{bmatrix} \lambda^1 & \dots & 0 \\ 0 & \dots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \dots & \lambda^m \end{bmatrix}, \ \widetilde{\mathbf{R}} = \begin{bmatrix} \mathbf{R}^{(1)}, \dots, \mathbf{R}^m \end{bmatrix} \tag{2.19}$$

in agreement with Def. 2.3.2, it follows that $\mathbf{M}\mathbf{R}^p = \lambda^p \mathbf{R}^{(p)}$.

Definition 2.3.4 (The Riemann Problem). A Riemann problem (RP) is a special initial value problem (IVP) consisting of a conservation law together with piecewise constant initial data having a single discontinuity. The dam break problem is a simple example of a RP in which the water depth at the left of the dam is different from that at the right of this one, at the initial time, i.e., the time when the break occurs.

In a FV context, RPs arise intrinsically as a consequence of the discretization of the physical domain. Here we consider the SWEs (Eqs. 2.11 and 2.12) with piecewise constant initial data, along the x axis, given by:

$$\overset{\circ}{h}(x,0) = \begin{cases} h_l & \text{if} \quad x < 0, \quad \overset{\circ}{U}(x,0) = 0 \\ h_r & \text{if} \quad x > 0, \quad \overset{\circ}{V}(x,0) = 0 \end{cases} \tag{2.20}$$

where initial data is denoted by a circle above the variable, e.g., $\overset{\circ}{h}$, rather than utilizing subscripts. Notice that the initial jump discontinuity in Eq. 2.20 occurs at $x = 0$, $y = 0$.

2.4 Some properties of the Shallow Water Equations

This section is devoted to describe the main mathematical properties of the SWE, since they play a fundamental role when implementing numerical methods for solving any set of PDEs. Consequently, the necessity of summarizing here the main mathematical aspects is clearly well justified.

There are essentially three types of PDEs, namely parabolic, hyperbolic and elliptic PDEs. This classification gives rise to important differences between one type and another. For instance, parabolic PDEs arise from the mathematical analysis of time dependent diffusion processes, while elliptic ones are typically related to steady state - equilibrium - processes.

The SWEs, as presented in Eq. 2.10, are a first order hyperbolic system of PDEs. Hyperbolic equations physically represent the rate of change of a physical quantity with respect to time and space variables, i.e., the transport of such quantity. They exhibit wave-like solutions, indeed the "wave equation" is the simplest model of a hyperbolic PDE.

Mathematically, a system of PDEs is hyperbolic if the Jacobian matrices are diagonalizable and they have only real eigenvalues. Moreover, if the eigenvalues are all real and distinct, the system is strictly hyperbolic. An interesting property of the Jacobian is that is diagonalizable, thus the system becomes a set of decoupled equations. The eigenvalues of the Jacobian matrices for the SWEs, Eq. 2.16, are given by:

$$\lambda_1 = U - c, \qquad \lambda_2 = U, \qquad \lambda_3 = U + c, \tag{2.21}$$

and

$$\lambda_1 = V - c, \qquad \lambda_2 = V, \qquad \lambda_3 = V + c \tag{2.22}$$

where c is the wave celerity, as defined in Eq. 2.17. Physically, these represent the speeds at which information is propagated. Since the information is propagated at finite speeds, disturbances have a domain of influence and not every point in the physical domain feels the disturbance at once.

It is worth to mention that the Jacobian matrices of the SWEs, **A** and **B** (Eq. 2.16), can not be diagonalizable simultaneously, and mainly for this reason most of the schemes are based on a normal flux formulation (17). Such normal flux formulation is also possible given that the SWEs suffice the rotational invariance property (18, 29), i.e., they remain invariant under rotations (Eq. 2.23).

$$\mathbf{F}n_1 + \mathbf{G}n_2 = \mathbf{T}^{-1}\mathbf{F}(\mathbf{T}\widetilde{\mathbf{U}}) \tag{2.23}$$

where $\mathbf{T}$ is the rotational matrix and $\mathbf{T^{-1}}$, its corresponding inverse matrix.

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & n_1 & n_2 \\ 0 & -n_2 & n_1 \end{bmatrix} \qquad \mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & n_1 & -n_2 \\ 0 & n_2 & n_1 \end{bmatrix} \tag{2.24}$$

2.5 From a Continuous to a Discrete Model of the SWEs

So far we have presented the equations that govern the Hydrodynamics of shallow flows, i.e., the SWEs. These equations have been obtained from applying two conservation laws, namely Mass and Momentum Conservation to an infinitesimal control volume of fluid that it is assumed to be continuous. However, equations in their continuous form don't allow to obtain a numerical solution. The process of transforming a continuous equation into a form that admits numerical solutions is called "discretization" and such resulting form is known as "discrete equation".

The discretization process includes not only to obtain a system of algebraic equations that it is equivalent to the continuous form, but also to split the domain into a finite number of points where the solution will be available at. For this purpose the FV method was selected because of its well known suitability for conservation laws. Such method requires the domain to be splitted into cells,

rather than points. Later, once the domain is divided into cells, the conservation laws are applied over each of them. In turns every cell encloses a "finite volume", thus the name of the method.

In broad terms the numerical algorithm includes the following steps:

- Divide the domain into a finite number of cells. In practice, solution domains in real engineering problems are too extensive and thus they may have very diverse properties. Given that the flow quantities are assumed to be piecewise constant and the solution is only available at the discrete locations, the size of every cell has a direct influence on the accuracy of the results. However, the smaller cell sizes are, the higher the computational cost is. The resulting mesh from this step is not a trivial part of the solution process.

- Integration of the governing equations, in this case the SWEs, over all the finite control volumes of the domain. The integral form of the equations is also known as "weak form" in the sense that they weaken the differentiability requirements, at the time they admit discontinuous solutions. To obtain this integral form, one needs to integrate Eq. 2.12 term by term over the control volume and then to apply the Gauss's Divergence Theorem to some of the resulting terms. The corresponding integral form of Eq. 2.12 is given by:

$$\frac{d}{dt}\int\int\int_{\mathbf{V}} \widetilde{\mathbf{U}}\, dV + \int\int_{A} \boldsymbol{\psi}\cdot d\mathbf{A} = 0 \tag{2.25}$$

  where $\mathbf{V}$ is a control volume, A is the boundary of V, $\boldsymbol{\psi} = (\mathbf{F}, \mathbf{G}, \mathbf{S})$ is the tensor of fluxes and $\mathbf{n}$ is the outward unit vector normal to the boundary surface A (see Fig. 2.2). Equation 2.25 may be interpreted as saying that the time rate of change of the conserved quantities within the fixed control volume depends solely on the total flux through the boundary surface, i.e., the balance between the different processes that can increase or decrease every quantity.

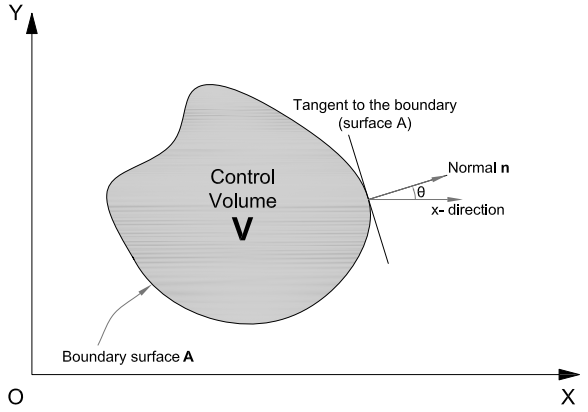


Figure 2.2: X-Y plane view of control volume $\mathbf{V}$ and its boundary, $\mathbf{A}$.

A non-trivial task that needs to be performed then, is how the fluxes leaving or entering the domain are computed.

- The resulting integral equations need to be transformed into a set of of algebraic equations. As it was indicated above, this is the so called discretization of the equations and it will allow to calculate the fluxes across the boundary surface.
- Solution of the system of algebraic equations. At this stage, the solution at the next time step is updated with calculated fluxes. Depending on the numerical scheme, it may be required to use or not an iterative method.

2.6 Boundary Conditions

A boundary value problem (BVP) is a differential equation subject to specific constraints along the entire boundary of the domain. In contrast to a IVP, in a BVP the constraints need to be specified at every time stage and not just at an initial time. These constraints are called boundary conditions (BCs) and they must also satisfy the differential equation at the boundary. Thus weak form of the equations weaken the restrictions for BCs as well.

In many hydraulic engineering problems one deals with finite domains. The description of these problems in such a physical domain requires certain knowledge of the flow variables at the boundaries, besides the initial status of all of them. A problem like this is known as Initial Boundary Value Problem (IBVP), i.e., a differential equation subject to initial conditions (ICs) and BCs. This is the type of problems that we are concerned with in this thesis.

Depending on the type of value specified, BCs can be classified into:

- **Dirichlet or First-type**: the value prescribed is a flow variable itself, e.g., $\mathbf{U}_x|_1 = 0$. Here, point 1 is an arbitrary point of the boundary.
- **Neumann or Second-type**: the value prescribed is a derivative of a flow variable, e.g., $\mathbf{U}_x|_1 = 0$.
- **Robin or Third-type**: it is a weighted combination of Dirichlet and Neumann BCs.

It must be pointed out that a correct numerical implementation of BCs must be consistent with the physical BCs as well. As for the physics, they can be divided into:

- **Reflecting**: any wave reaching the boundary will be reflected back inside the computational domain.
- **Absorbing**: a wave reaching the boundary will leave the domain without perturbing the flow field inside this one. This condition is commonly used to represent open boundaries.
- **Mixed**: a wave reaching the boundary will be partially absorbed and therefore partially reflected, accordingly. For instance, at the shore the waves are partially reflected.

Chapter 3

"Multi-quadtree" mesh implementation

Building a mesh is a primary step when numerically solving a partial differential equation by finite volume methods. It is actually one of the most time consuming tasks, considering that the quality of the results depends to a large extent on the mesh. A mesh is a network of vertices, edges and faces that can have almost any geometrical shape to represent a given 2D object. Three dimensional representations are also possible, but they are out of the scope of this work.

Triangles and quadrilaterals are among the most common cell shapes, although triangular ones are probably preferred within a finite element framework. According to the type of connectivity, a mesh can be either structured, unstructured or a combination of these two (hybrid). The highly irregular shapes usually found along coastlines or around water bodies can make difficult to adequately discretize such a domain. Unstructured meshes, those having irregular connectivity, therefore are more suitable when solving coastal engineering problems.

The term quadtree is a class of hierarchical tree-type data structure based on the recursive decomposition of space. The first applications were in the field of computer image processing (38) and have been later extended to other disciplines (5, 41, 47, 49). It has the advantage of allowing different mesh densities, i.e. local refinement, within the numerical domain. In a quadtree mesh each node has four children and, if required, each of these children can have similarly four children. The mesh strategy presented here is based on that of Posada-Vanegas (32), although slightly different from it. The traditional quad-tree approach is usually not feasible for coastal environments, because the dimension along the coastline tends to be greater than the dimension across the beach. Posada-Vanegas (32) overcame this issue by dividing the computational domain into square subdomains and implementing the traditional quad-tree algorithm to each one of them.

In this work, the computational domain is covered with a regular mesh and each of these square cells are refined to the same level according to the criteria defined by the user. This meshing, hereinafter referred to as modified multi-quadtree (MMQ), reduces the number of verifications necessary to ensure that none of the neighbouring cells dimensions differ by more than a factor of two. The steps required to generate the mesh with the MMQ algorithm can be summarized as follows:

1. Define the computational domain.
2. Discretize the computational domain into uniform square cells. Since this will be the largest cell of the mesh in some areas, it is important to take into account the physics and the geometry of the problem in order to set its size. An oversized cell in this step can result in loss of relevant details in the solution.
3. Define the "seed points" and their level of refinement. These are the coordinates "x" and "y" of the points used to indicate at which level must be refined each cell.
4. Verify that the difference in size of every two neighbor cells does not exceed a factor of 2 and, if necessary, reduce the level of refinement at those cells where this criteria is not fulfilled.
5. It is possible that the quantities, e.g. the velocity, the water depth, etc., may not be available at certain locations of the new grid. Therefore, the last step is to find the quantities at each cell center by means of a bilinear interpolation.

3.1 Mesh generation

This section presents a detailed description of the algorithm and all the steps required to generate a mesh by the "MMQ" method. The procedure is better explained using a practical example. In Fig. 3.1 the blue line represents an arbitrary computational domain and the black lines are the subdivision of this domain into regular square cells. Each one of the cells in Fig. 3.1 is a "mother cell", but the number of "children" of each mother cell is given by the location of the "seed points". Each cell is identified by a number ID and a locator, the number of node in the **x** and **y** direction, respectively.

In a second step, each seed point has to be related to a mother cell. Equation 3.1 finds the indexes **i** and **j** of the mother cell to which a seed point, whose coordinates are **x** and **y**, belongs to.

$$i = \left\lfloor \frac{x - x_{min}}{\delta_x} \right\rfloor + 1 \quad , \quad j = \left\lfloor \frac{y - y_{min}}{\delta_y} \right\rfloor + 1 \tag{3.1}$$

Where:
x_{min}, y_{min} = are the minimum **x** and **y** coordinates, i.e., the lowest limit at the left boundary of the computational domain (see Fig. 3.1).
δ_x, δ_y = are the grid spacing along the **x** and **y** direction, respectively. For the meshing presented in this thesis, the cells are squares thus $\delta_x = \delta_y = \delta$.

This allows to define the level of refinement required at each mother cell. When there are seed points with different level of refinement at the same mother cell, their maximum value is selected for that cell. The grid size of the refined cell is calculated by eq. 3.2, where **k** is the level of

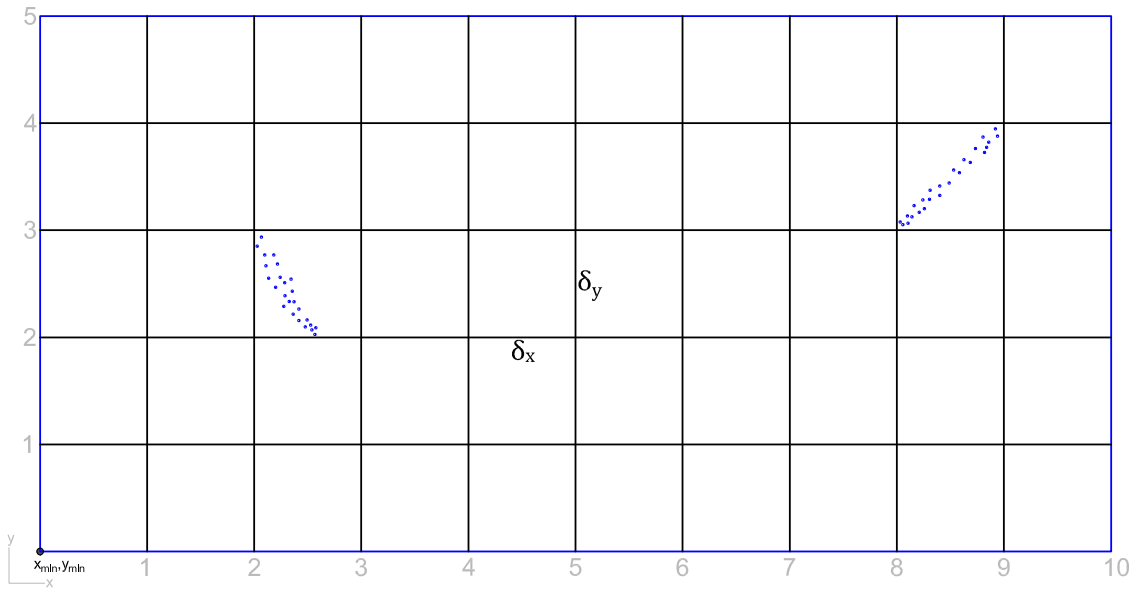
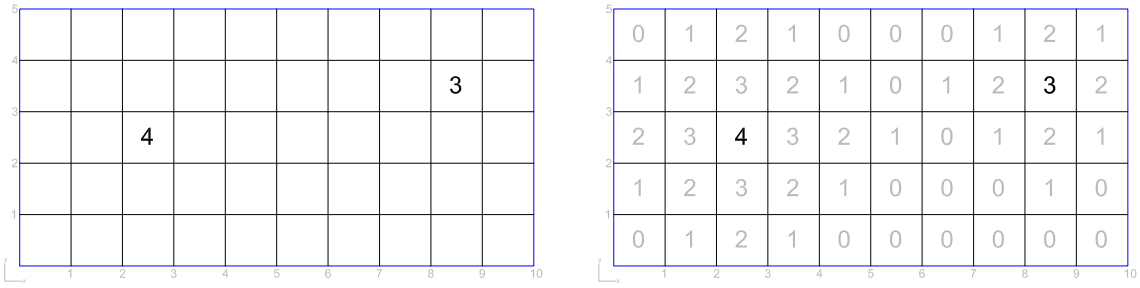


Figure 3.1: Regular grid over the computational domain.

refinement and $\delta_{\mathbf{ref}}$ is the grid size of the children cells. In the example (Fig. 3.1) eq. 3.1 will result in two nodes of the regular mesh, which are (2,2) and (8,3).

The assignation of the refinement level is performed, at a first stage, only in consistency with



(a) Initial stage. **(b)** After regularization.

Figure 3.2: Assignation of refinement level.

the location of the seed points (see Fig. 3.2a). At the next stage, the level of refinement must be regularized in the entire computational domain. Regularization is the process of adjusting the level of refinement of the direct neighbors of those cells whose level was modified during the first stage and, in turn, the neighbors of these neighbors. The traditional condition 2:1 is adopted here, i.e. the ratio between the linear dimension of every two adjacent cells can not be greater than 2. One of the difficulties when discretizing partial differential equations over a quadtree mesh, are the loose

nodes that arise across the interface between two cells of different size. The number of loose nodes should be minimized (36, 50); therefore, the relevance of ensuring the 2:1 ratio. Although it may be relevant for some specific cases, the diagonal neighbors are not regularized in this meshing. Figure 3.2b shows the refinement level for every mother cell, once the regularization is carried out.

$$\delta_{ref} = \frac{\delta}{2^k} \tag{3.2}$$

Figure 3.3 shows the mesh after refinement, in agreement with the levels in Figure 3.2b. This meshing has the advantage that the process of verifying the 2:1 ratio needs to be performed only over the mother cells and not to every child, since the entire mother cell has the same level of refinement. Also it is practical in terms of handling information and storage, considering that the unique ID of every cell is only a consecutive number while the coordinates of its centroid are stored in a separate variable. Numbering is carried out from left to right and from the bottom to the top, as depicted in Figure 3.4. One of the implications of this numbering system is that it is necessary

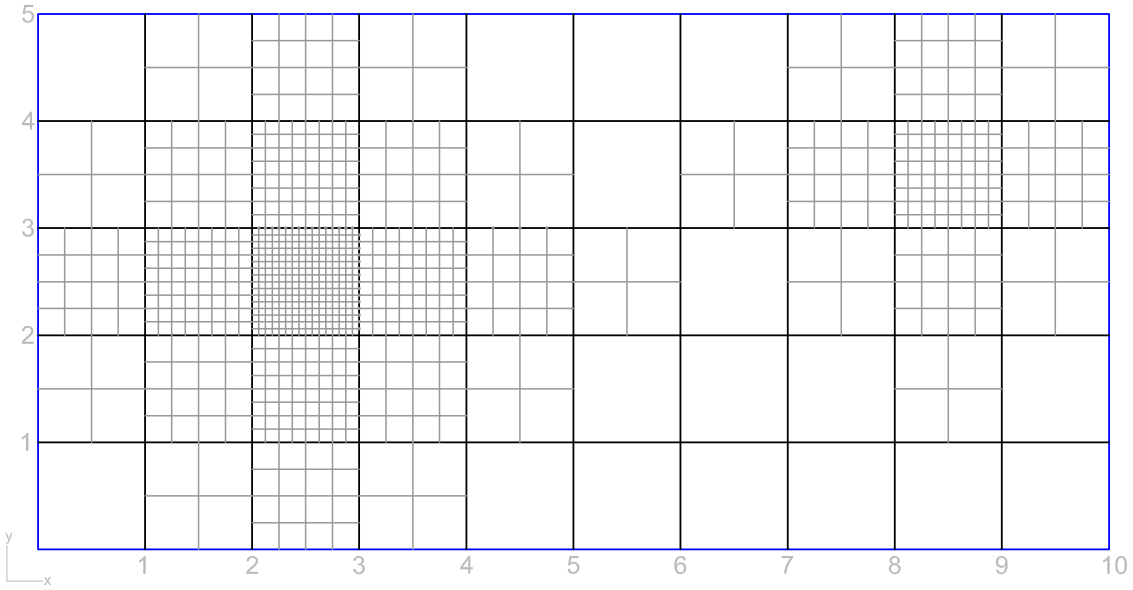


Figure 3.3: Mesh after the regularization

to create individual variables to name the neighbors of an arbitrary cell. Figure 3.5a depicts the convention of neighbors adopted in this thesis. Every cell, regardless of being or not a child, is assigned a value for each of the neighbors. Negative numbers are usually associated to situations where such neighbor does not exist, e.g. a cell at the boundary of the computational domain. This convention is used to name the neighbors of an arbitrary cell, when both are at the same level (Fig.

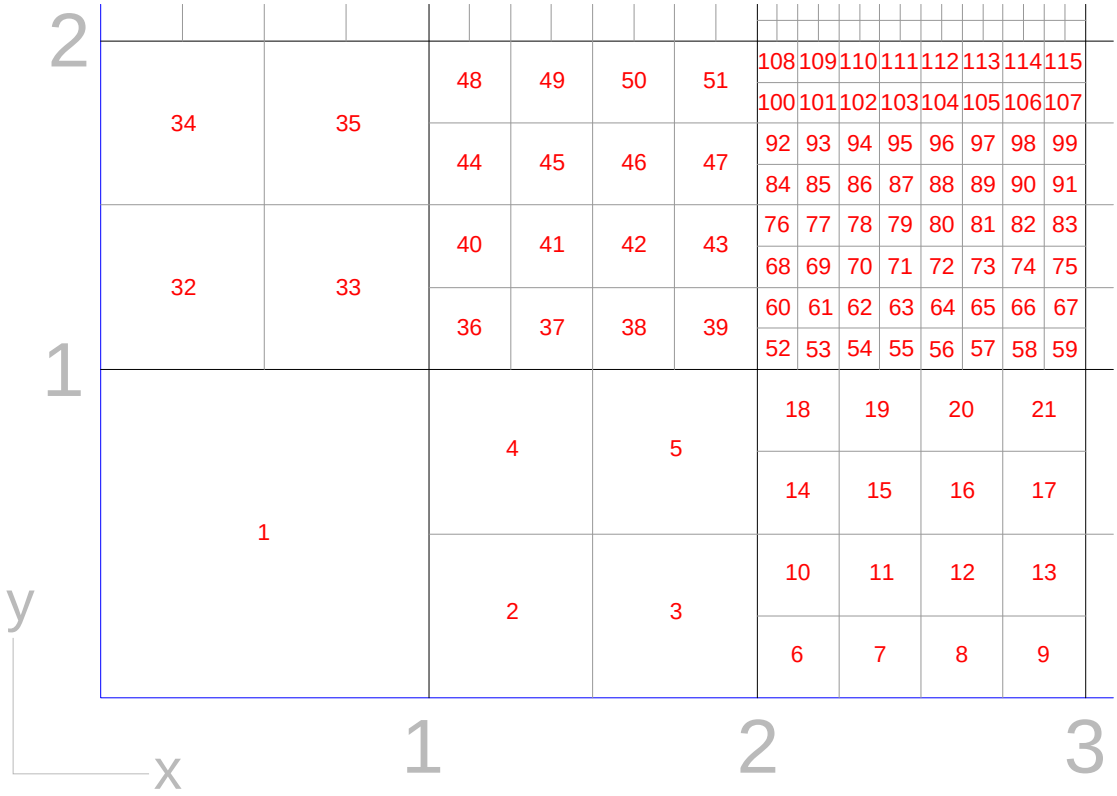


Figure 3.4: Neighbors

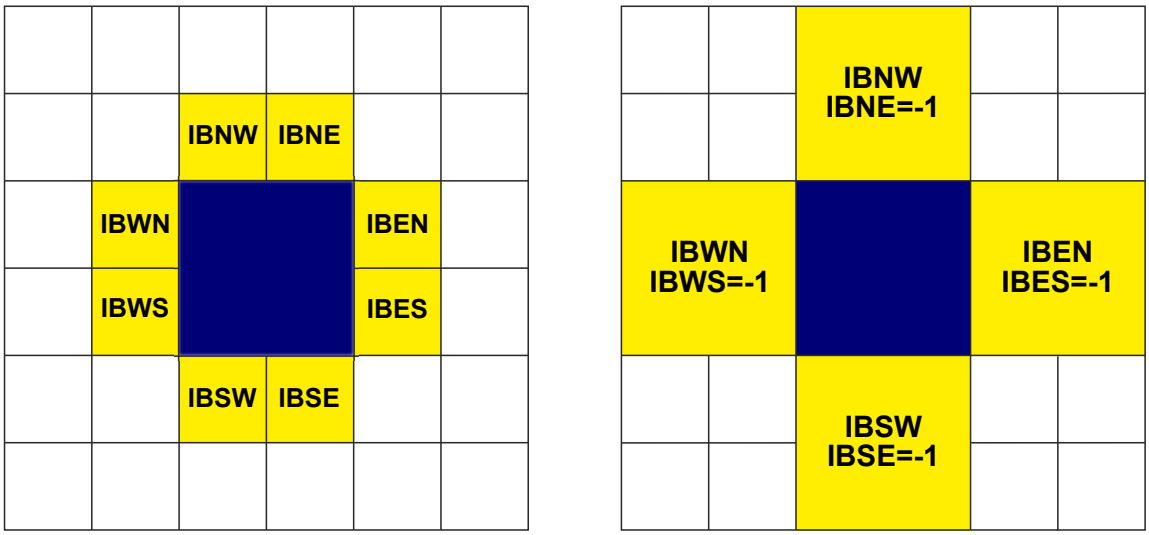
3.5b). In this case, the cell is considered to have a neighbor at the North-West direction and a negative value of -1 for the neighbor at North-East direction.

A neighbor, in an arbitrary direction, takes a value of -2 when it has a level of refinement greater than that of the cell at the other side of the interface. The remaining cases are illustrated in Figure 3.6. Finally, the neighbor takes a value of -3 when a cell is located at one of the boundaries.

3.2 Primitive variables interpolation

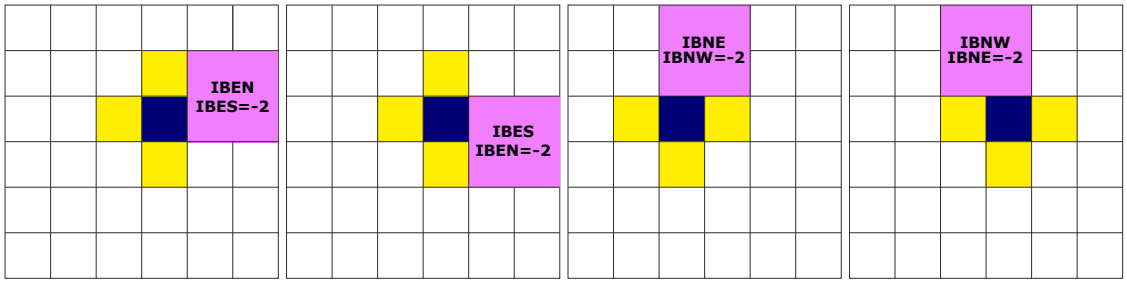
It should be mentioned that the grid is collocated, i.e. all the primitive variables are stored at the cell center. As for the velocities, they are usually initialized with zero values and no interpolation is required. But for other variables data may not be available at the center of every cell and they must be interpolated. The two specific cases of bathymetry and wind are covered in the next paragraphs.

When it comes to the bathymetry, the algorithm identifies the limits of a rectangle that covers the area where data are required at every refinement level. These limits are used to generate

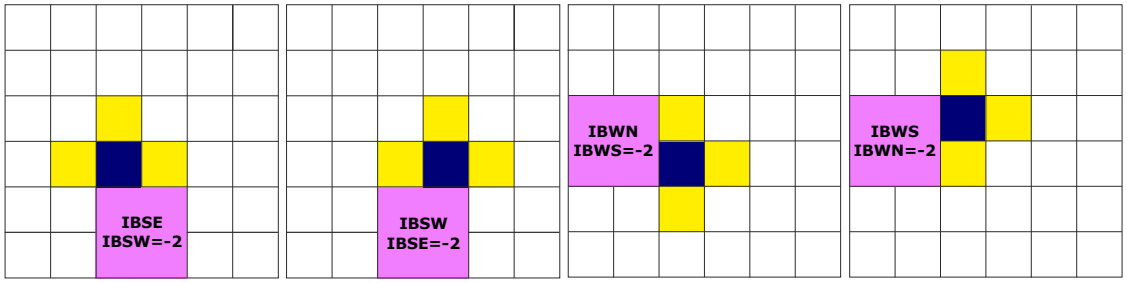


(a) Convention of neighbors for an arbitrary cell. **(b)** Arbitrary cell with neighbors of equal size.

Figure 3.5: Convention of neighbors.



(a) East-North neighbor. **(b)** East-South neighbor. **(c)** North-East neighbor. **(d)** North-West neighbor.



(e) South-East neighbor. **(f)** South-West neighbor. **(g)** North-West neighbor. **(h)** West-South neighbor.

Figure 3.6: Convention of neighbors with a different level of refinement.

a **.GRD** file extension in text format, one for each refinement level. The **.GRD** file is generated with software dedicated to the processing of survey data, therefore provided with a wide range of interpolators. In such a way, data are available at every cell and coding interpolation is avoided. This characteristic makes adaptive mesh refinement, at least at present, not to be an option since these files are created outside the mesh algorithm. In the example (Fig. 3.2), the nodes of such limits are presented below for every level of refinement (Table 3.1).

Table 3.1: Limits for every level of refinement.

k	Minimum	Maximum	Grid size
4	(2,2)	(3,3)	$\delta_4 = \frac{\delta}{2^4}$
3	(1,1)	(9,4)	$\delta_3 = \frac{\delta}{2^3}$
2	(0,0)	(10,5)	$\delta_2 = \frac{\delta}{2^2}$
1	(0,0)	(10,5)	$\delta_1 = \frac{\delta}{2^1}$

For instance, for a refinement level of 4, the **.GRD** file should be created from the coordinates of node (2,2) to the coordinates of node (3,3), while the spacing of such grid should be equal to $\frac{\delta}{16}$ and so on, repeating the process for every refinement level. The algorithm will assign those depths at every cell of the refined mesh.

As for the wind, the algorithm uses bilinear interpolation to find the value of the velocity components at every node and linear interpolation to find the variation of these data over time.

3.3 Neighbors finding

One of the most important aspects is how practical is retrieving information of the mesh. Therefore this section is dedicated to the expressions used to find such information, i.e. the position of the center of every cell and its neighbors.

$$S_{(i,j)} = \sum_{(p,q)=(1,1)}^{(i,j)} \left(2^{2k_{(p,q)}}\right) - 2^{2k_{i,j}} \tag{3.3}$$

Where:
(p,q) = are the number of nodes in the x and y direction, respectively.
$S_{(p,q)}$ = is the summation of the number of cells, including their children, located up to the cell of

interest.
$k_{(p,q)}$ = is the level of refinement as previously defined in Section 3.1.

Each cell is numbered, from left to right and from the bottom to the top, according to Equation 3.4 and the position of its center is calculated with Equations 3.5 and 3.6.

$$P = S_{(i,j)} + (jj-1) * 2^{k_{(i,j)}} + ii \tag{3.4}$$

$$X_p = X_A + \left(i - \frac{3}{2}\right) * \delta_{(i,j)} + \left(ii - \frac{1}{2}\right) * \delta_{ref_{(i,j)}} \tag{3.5}$$

$$Y_p = Y_A + \left(j - \frac{3}{2}\right) * \delta_{(i,j)} + \left(jj - \frac{1}{2}\right) * \delta_{ref_{(i,j)}} \tag{3.6}$$

Where:
P = is a unique identifier for each cell. It is necessary to retrieve all the properties of a given cell.
(i, j) = counter in the x and y direction, respectively, for an arbitrary mother cell. The maximum value of i is m, while n is the maximum value of j.
(ii, jj) = counter in the x and y direction, respectively, for the children of an arbitrary cell. They both have a maximum value of 2^k.
X_P = coordinate in the x direction of the center of cell p.
Y_P = coordinate in the y direction of the center of cell p.
X_A = coordinate in the x direction of the center of cell No. 1.
Y_A = coordinate in the y direction of the center of cell No. 1.

Table 3.2 summarizes the expressions derived for finding the neighbors in all the relevant directions of an arbitrary cell P. Depending on the location of the cell, this may have either all neighbors or only neighbors at certain direction. The different possibilities are listed as an individual case in Table 3.2. All the expressions derived for finding the neighbors have a subscript "P", which is not shown for the sake of simplicity. For instance IBWN$_P$ is only expressed as IBWN, on the understanding that is the neighbor located at the West-North direction of cell P.

Table 3.2: Expressions for neighbors finding.

CASE				NEIGHBORS
i = 1 & ii = 1				IBWN = null IBWS = null
i = m & ii = 2^k				IBEN = null IBES = null
i > 1	ii = 1	*If* $k_{(i-1,j)} = k_{(i,j)}$		IBWN = $S_{(i-1,j)} + jj * 2^{k_{(i-1,j)}}$ IBWS = null
		If $k_{(i-1,j)} > k_{(i,j)}$		IBWN = $S_{(i-1,j)} + 2jj * 2^{k_{(i-1,j)}}$ IBWS = $S_{(i-1,j)} + (2jj-1) * 2^{k_{(i-1,j)}}$
		If $k_{(i-1,j)} < k_{(i,j)}$	*If* $\text{jj}_{2\mathbb{N}} \Rightarrow$	IBWN = $S_{(i-1,j)} + (jj/2) * 2^{k_{(i-1,j)}}$ IBWS = null
			If $\text{jj}_{2\mathbb{N}+1} \Rightarrow$	IBWN = null IBWS = $S_{(i-1,j)} + (jj+1)/2 * 2^{k_{(i-1,j)}}$
	ii $\neq$ 1			IBWN = $C_p - 1$ IBWS = null
i < m	ii = 2^k	*If* $k_{(i+1,j)} = k_{(i,j)}$		IBEN = $S_{(i+1,j)} + (jj-1) * 2^{k_{(i+1,j)}} + 1$ IBES = null
		If $k_{(i+1,j)} > k_{(i,j)}$		IBEN = $S_{(i+1,j)} + (2jj-1) * 2^{k_{(i+1,j)}} + 1$ IBES = $S_{(i+1,j)} + 2(jj-1) * 2^{k_{(i+1,j)}} + 1$
		If $k_{(i+1,j)} < k_{(i,j)}$	*If* $\text{jj}_{2\mathbb{N}} \Rightarrow$	IBEN = $S_{(i+1,j)} + (jj-2)/2 * 2^{k_{(i+1,j)}} + 1$ IBES = null
			If $\text{jj}_{2\mathbb{N}+1} \Rightarrow$	IBEN = null IBES = $S_{(i+1,j)} + (jj-1)/2 * 2^{k_{(i+1,j)}} + 1$
	ii $\neq$ 1			IBEN = $C_p + 1$ IBES = null

Continued on next page

Table 3.2 – *Continued from previous page*

CASE		NEIGHBORS		
j = 1 & jj = 1		IBSW= null IBSE = null		
j = n & jj = 2^k		IBNW = null IBNE = null		
j > 1	jj = 1	*If* $k_{(i,j-1)} = k_{(i,j)}$	IBSW = $S_{(i,j-1)} + 2^{k_{(i,j-1)}} * \left(2^{k_{(i,j-1)}} - 1\right) + ii$ IBSE = null	
		If $k_{(i,j-1)} > k_{(i,j)}$	IBSW = $S_{(i,j-1)} + 2^{k_{(i,j-1)}} * \left(2^{k_{(i,j-1)}} - 1\right) + 2ii - 1$ IBSE = $S_{(i,j-1)} + 2^{k_{(i,j-1)}} * \left(2^{k_{(i,j-1)}} - 1\right) + 2ii$	
		If $k_{(i,j-1)} < k_{(i,j)}$	*If* $\text{ii}_{2\mathbb{N}} \Rightarrow$	IBSW = $S_{(i,j-1)} + 2^{k_{(i,j-1)}} * \left(2^{k_{(i,j-1)}} - 1\right) + ii/2$ IBSE = null
			If $\text{ii}_{2\mathbb{N}+1} \Rightarrow$	IBSW = null IBSE = $S_{(i,j-1)} + 2^{k_{(i,j-1)}} * \left(2^{k_{(i,j-1)}} - 1\right) + (ii+1)/2$
	jj $\neq 1$	IBSW = $C_p - 2^{k_{(i,j)}}$ IBSE = null		
j < n	jj = 2^k	*If* $k_{(i,j+1)} = k_{(i,j)}$	IBNW = $S_{(i,j+1)} + ii$ IBNE = null	
		If $k_{(i,j+1)} > k_{(i,j)}$	IBNW = $S_{(i,j+1)} + 2ii - 1$ IBNE = $S_{(i,j+1)} + 2ii$	
		If $k_{(i,j+1)} < k_{(i,j)}$	*If* $\text{ii}_{2\mathbb{N}} \Rightarrow$	IBNW = $S_{(i,j+1)} + ii/2$ IBNE = null
			If $\text{ii}_{2\mathbb{N}+1} \Rightarrow$	IBNW = null IBNE = $S_{(i,j+1)} + (ii+1)/2$
	jj $\neq 1$	IBNW = $C_p - 2^{k_{(i,j)}}$ IBNE = null		

Chapter 4

The HLLCS Riemann Solver

The HLLCS solver is an approximate Riemann Solver for the solution of the SWEs in the presence of source terms ($S \neq 0$). This solver was proposed by Murillo and García-Navarro (29) and later implemented by Salazar Carrillo (37) for the simulation of ocean circulation patterns in the Gulf of Mexico. To the best knowledge of the author, that is the only application of the HLLCS solver that has been reported in the literature.

The Conservation-Law form of the equations, as presented in Section 2.2.1, is particularly suitable for the introduction of the solver (Eqs. 4.1 and 4.2). For the sake of clarity, the components of the source term are replaced by a unique term that includes all the contributions. Therefore they read:

$$\frac{\partial \widetilde{\mathbf{U}}}{\partial t} + \frac{\partial \mathbf{F}(\widetilde{\mathbf{U}})}{\partial x} + \frac{\partial \mathbf{G}(\widetilde{\mathbf{U}})}{\partial y} = \mathbf{S}(\widetilde{\mathbf{U}}) \tag{4.1}$$

with

$$\widetilde{\mathbf{U}} = \begin{bmatrix} h \\ hU \\ hV \end{bmatrix} \qquad \textbf{Vector of conserved variables} \tag{4.2a}$$

$$\mathbf{F} = \begin{bmatrix} hU \\ hU^2 + \frac{1}{2}gh^2 \\ hUV \end{bmatrix} \quad \mathbf{G} = \begin{bmatrix} hV \\ hVU \\ hV^2 + \frac{1}{2}gh^2 \end{bmatrix} \qquad \textbf{Flux vectors} \tag{4.2b}$$

$$\mathbf{S} = \begin{bmatrix} 0 \\ S_x \\ S_y \end{bmatrix} \qquad \textbf{Source terms vector} \tag{4.2c}$$

Godunov's method (19) is the essential starting point for methods for non-linear conservation laws, such as the SWEs. The basis of Godunov's method is that the solution of the conservation law is obtained by solving the RP defined at the intercell boundary ($x_{i+1/2}$) as depicted in Fig. 4.1. The

initial conditions at the boundary $\left(x = i + \frac{1}{2}\right)$ are given by:

$$U(x,0) = \begin{cases} U_i & if \quad x < 0 \\ U_{i+1} & if \quad x > 0 \end{cases} \tag{4.3}$$

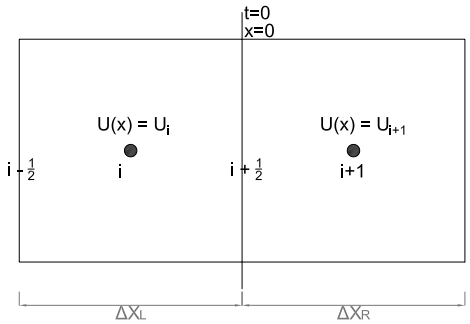


Figure 4.1: Spatial discretization in 1D and Riemann problem defined by two piecewise constant initial states.

Although Godunov's method (19) is based upon exact solutions of the RP, it set the basis for a family of schemes that later used approximate solutions instead (1, 35, 44) and thus are now called "Godunov-type" methods. In the particular case of hyperbolic systems it is also important the sign of the characteristics speeds and from that, the well-known term "upwind" to describe the class of methods that take this into account.

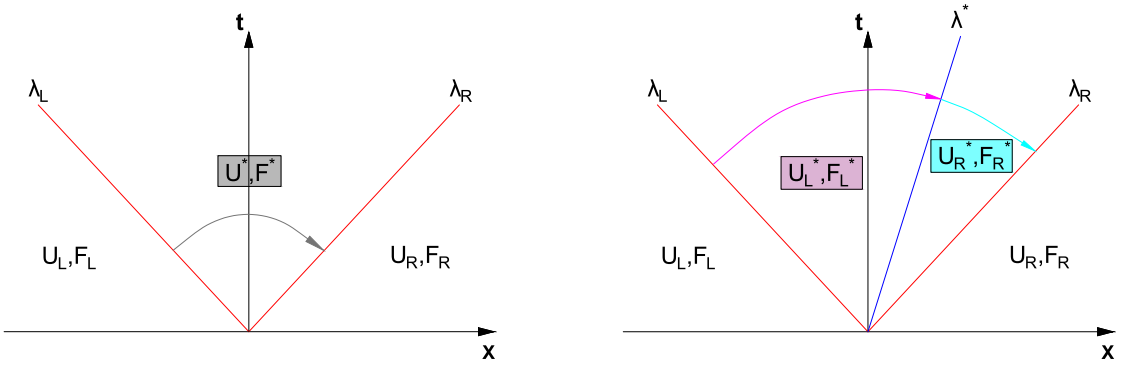
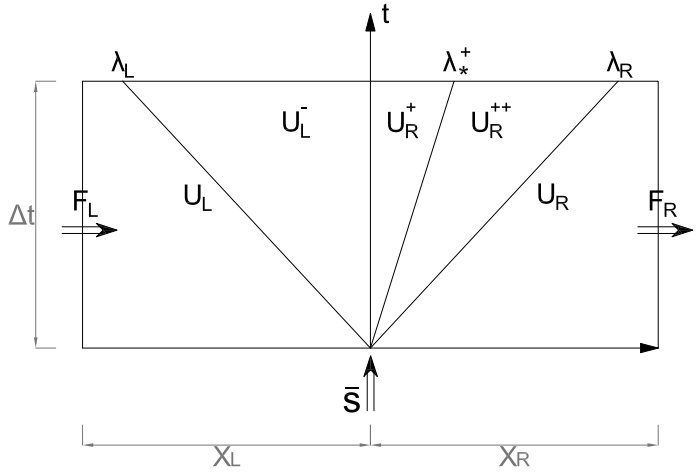


Figure 4.2: Structure of the solution as defined by the HLL (left) and the HLLC (right) schemes, respectively.

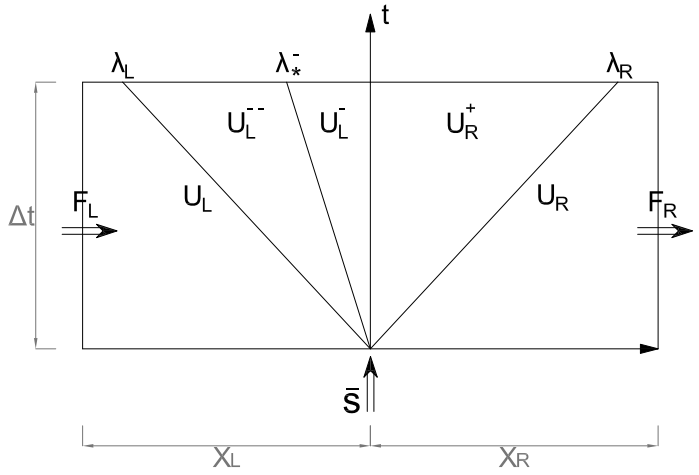
The HLLCS solver is a Godunov-type upwind scheme which is an augmented version of its predecessors, the HLL (1) and the HLLC (44) solver. HLL stands for Harten-Lax-van Leer while Toro et al. (44) added a "C" for the restored contact wave. Figure 4.2 shows a comparison of the

structure of the solution in both cases. While the HLL scheme (1) provides a solution based on a two waves structure, the HLLC (44) scheme added a contact wave that splits the solution in four regions with two intermediate states. However, the source terms had to be treated separately since these schemes resolve only the homogeneous SWEs ($S = 0$).

Murillo and García-Navarro (29) overcame this difficulty by adding an extra stationary wave located at the intercell boundary. The solution is given by a left and a right wave, separated by a contact wave and an stationary shock wave due to the presence of the source term. Thus the solution is split into five regions with three intermediate states. There are two possibilities depending on the sign of the contact wave (see Fig. 4.3), which can not be known a priori.



(a) Positive middle wave



(b) Negative middle wave

Figure 4.3: Structure of the solution of a RP as provided by the HLLCS scheme (29).

Since the scheme is explicit, the solution at the current time step, $\widetilde{\mathbf{U}}^{n+1}$, depends entirely on the solution at the previous time step, n. The one-dimensional FV scheme reads:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \sum_{k=1}^{\mathrm{NE}} \left(\mathbf{T}^{-1}\mathbf{F}\right)_{i,k}^{-} \frac{l_k \Delta t}{A_i} \tag{4.4}$$

where:
Δt = time step restricted by the CFL condition.
i = the ID of the cell where the solution is being calculated.
A = the enclosed area within the cell.
NE = the number of edges of the cell, i.e., the number of RPs solved for every cell.
l = the length of edge k.
T^{-1} = inverse of the rotation matrix (Eq. 2.24).
$\mathbf{F}$ = the flux at the intercell boundary between cells i and k.

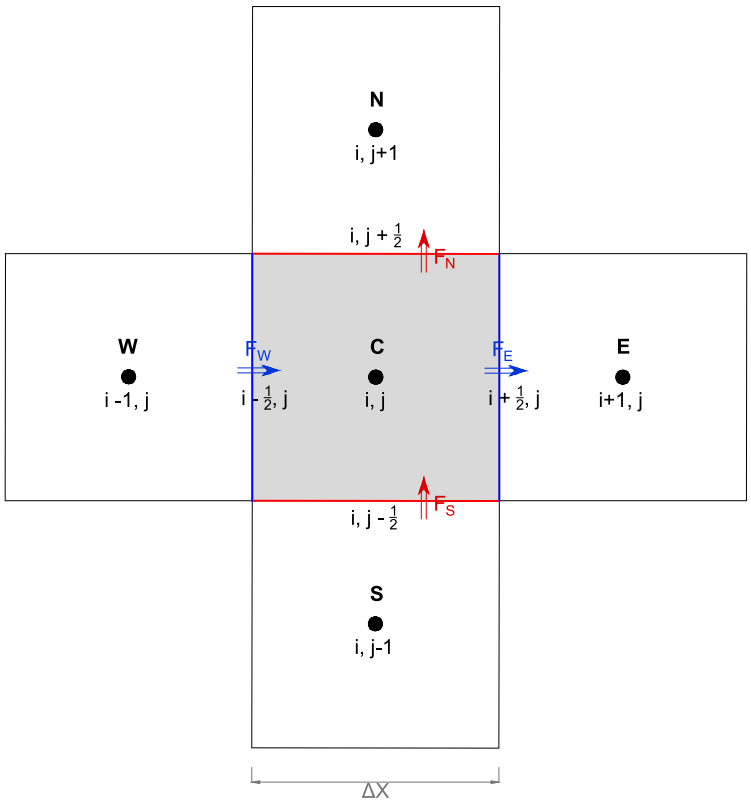


Figure 4.4: Finite volume discretization in a collocated grid. All the flow variables are located in the center of the cell, thus the fluxes at the intercell boundaries must be calculated. Here, every edge has a length of Δx and the area A of cell C is the shaded area.

For instance, Fig. 4.4 shows a 2D FV discretization where cell **C** has four neighbors: cells **N**, **S**, **E** and **W**, respectively. In order to update the solution for the flow variables in cell **C**, one needs to calculate the fluxes ($\mathbf{F_N}, \mathbf{F_S}, \mathbf{F_E}$, and $\mathbf{F_W}$) at its four intercell boundaries.

As previously stated, by the rotational invariance property (Eq. 2.23), the 2D case is based upon the superposition of independents 1D RPs projected onto the normal direction to every cell side. In Fig. 4.4 NE is equal to four, therefore one needs to solve two RPs in the horizontal direction and two RPs in the vertical one. The steps required to calculate the fluxes by the HLLCS solver (29) with this approach are listed below.

1. Project the conserved variables, fluxes and source terms onto the normal direction to each cell edge. The normal flux is calculated by:

$$\mathbf{F}n_1 + \mathbf{G}n_2 \tag{4.5}$$

Notice that Eq. 4.5 is equal to the left-hand side of Eq. 2.23, which is the rotational invariance property. Therefore it is possible to calculate the vector of projected conserved variables, $\widehat{\mathbf{U}} = \mathbf{T}\widetilde{\mathbf{U}}$, and the projected flux vector, $\widehat{\mathbf{F}} = \mathbf{F}(\widehat{\mathbf{U}})$:

$$\widehat{\mathbf{U}} = \begin{bmatrix} h \\ h\hat{u} \\ h\hat{v} \end{bmatrix} \qquad \textbf{Projected conserved variables} \tag{4.6a}$$

$$\widehat{\mathbf{F}} = \begin{bmatrix} h\hat{u} \\ h\hat{u}^2 + \frac{1}{2}gh^2 \\ h\hat{u}\hat{v} \end{bmatrix} \qquad \textbf{Projected Flux} \tag{4.6b}$$

with the normal velocity, $\hat{u}$, and the tangential velocity, $\hat{v}$, defined by:

$$\begin{aligned} \hat{u} &= Un_1 + Vn_2 \\ \hat{v} &= -Un_2 + Vn_1 \end{aligned} \tag{4.7}$$

The source term vector, in Eq. 4.2c, is expressed in the normal direction as follows:

$$\mathbf{S_n} = \begin{bmatrix} 0 \\ S_n n_1 \\ S_n n_2 \\ 0 \end{bmatrix} \tag{4.8}$$

where S_n is the source term in the normal direction to the edge considered and it is different from $\mathbf{S_n}$. The projected source term vector, $\widehat{\mathbf{S}}_\mathbf{n}$, is obtained by applying the rotation matrix

(Eq. 2.24) to the source term in Eq. 4.8. The equivalent source term vector reads:

$$\widehat{\mathbf{S}} = \mathbf{T}\mathbf{S}_{\mathbf{n}} = \begin{bmatrix} 0 \\ S_n \\ 0 \\ 0 \end{bmatrix} \tag{4.9}$$

In consequence, the two-dimensional SWEs (Eq. 4.1) can be written in a one-dimensional form, on the basis of the projected variables:

$$\frac{\partial \widehat{\mathbf{U}}}{\partial t} + \frac{\partial \widehat{\mathbf{F}}}{\partial x} = \widehat{\mathbf{S}} \tag{4.10}$$

and the numerical scheme is written as presented in Eq. 4.4.

2. Calculate the right, $\widehat{\mathbf{U}}_R$, and left, $\widehat{\mathbf{U}}_L$, states, using Eq. 4.6a. This is actually the vector of conserved variables, the "initial condition" that defines a RP at the intercell boundary.

   Unless otherwise specified, the "p" index takes values of "R" or "L", that stands for the right and left states, respectively.

$$\widehat{\mathbf{U}}_p = \begin{bmatrix} h \\ h\hat{u} \\ h\hat{v} \end{bmatrix}_p \tag{4.11}$$

   Similarly, calculate the right, $\widehat{\mathbf{F}}_R$, and the left, $\widehat{\mathbf{F}}_L$, fluxes with Eq. 4.6b:

$$\widehat{\mathbf{F}}_p = \begin{bmatrix} h\hat{u} \\ h\hat{u}^2 + \frac{1}{2}gh^2 \\ h\hat{u}\hat{v} \end{bmatrix}_p \tag{4.12}$$

3. Calculate the source terms in correspondence with the physics of the problem. Three types of source term are considered: thrust exerted over the bed (S_z), bed friction (S_τ) and wind stress (S_w). The second component of the source term, $\widehat{\mathbf{S}}_2$, is given by the sum of the three contributions (Eq. 4.13). Notice that, unlike the fluxes and the initial states, the source term needs to be calculated only at the intercell boundary.

$$\widehat{\mathbf{S}}_2 = \left(S_z + S_\tau + S_w\right)_{i+1/2} \tag{4.13}$$

$$\widehat{\mathbf{S}} = \begin{bmatrix} 0 \\ S_z + S_\tau \Delta x + S_w \Delta x \\ 0 \end{bmatrix} \tag{4.14}$$

The three contributions appearing in Eqs. 4.13 and 4.14, are described in the following paragraphs.

- **Thrust exerted over the bed**
  This term is actually the result of decomposing the pressure term, while deducting the equations, into a flux contribution and a source term contribution. This decomposition makes possible to keep the hyperbolic character of the equations. The flux for the convective part must be balanced with the contribution of this source term. Therefore some difficulties arise from this term, specially under the presence of very irregular beds.

  Murillo and García-Navarro (29) proposed to calculate the thrust under the assumption that the pressure head over the bed follows a hydrostatic distribution and that it depends only on the water surface level at the lowest side of the discontinuity. Thus the thrust term, T, is written as:

$$T = g\rho \left(h_j - \frac{|\delta z'|}{2} \right) \delta z' \tag{4.15}$$

  with

$$j = \begin{cases} L & \text{if } \delta z \geq 0 \\ R & \text{if } \delta z < 0 \end{cases} \qquad \delta z' = \begin{cases} h_L & \text{if } \delta z \geq 0 \text{ and } d_L < d_R \\ h_R & \text{if } \delta z < 0 \text{ and } d_R < d_L \\ \delta z & \text{otherwise} \end{cases} \tag{4.16}$$

  where $\delta z = z_R - z_L$, $d = h + z$ and $\delta d = d_R - d_L$.
  To avoid overestimation in cases of wet and dry fronts, where the pressure head is far of following a hydrostatic distribution, the following restriction is applied:

$$S_{z,2} = \begin{cases} T_{max} & \text{if } \delta d\, \delta z \geq 0 \text{ and } \widetilde{u}\, \delta z > 0 \\ -\frac{T}{\rho} & \text{otherwise} \end{cases} \tag{4.17}$$

  with

$$T_{max} = \begin{cases} -g\widetilde{h}(\delta z) & \text{if } |g\widetilde{h}(\delta z)| \geq |T| \\ -\frac{T}{\rho} & \text{otherwise} \end{cases} \tag{4.18}$$

The mean water depth, $\widetilde{h} = 1/2(h_R + h_L)$, is just the average of both depths, while the mean velocity, $\widetilde{u}$, is calculated as a Roe average:

$$\widetilde{u} = \frac{(U_L\sqrt{h_L} + U_R\sqrt{h_R})}{\sqrt{h_L} + \sqrt{h_R}} \tag{4.19}$$

- **Bed Friction**
  This term is usually modelled by a friction law that relates the shear stress at the bottom, τ_b, with the flow mean velocity, $\bar{u}$, and a friction coefficient, c_f. In practice, this coefficient depends on the material at the bed and the geometry of the section. It is also used to account for turbulence effects in depth averaged models. The friction component of the source term is given by:

$$S_\tau = -\Delta x \left(\frac{\tau_b}{\rho}\right)_{i+1/2} = -\Delta x \left(c_f \bar{u}|\bar{u}|\right)_{i+1/2} \tag{4.20}$$

  This estimation should be restricted to the minimum available kinetic energy, thus the following verification is applied:

$$\left|\frac{\tau_b}{\rho g \widetilde{h}}\Delta x\right|_{i+1/2} = \min\left(\left|\frac{\tau_b}{\rho g \widetilde{h}}\Delta x\right|, \left|\frac{\bar{u}|\hat{u}_{min}|}{2g}\right|\right)_{i+1/2} \tag{4.21}$$

  with, $|\hat{u}_{min}| = \min(|\hat{u}_R|, |\hat{u}_L|)$, the minimum velocity.
  Figure 4.5 shows a typical cell "C" with four neighbors of different refinement level. The variable Δx in Eqs. 4.20 and 4.21 is the distance between the center of one cell to the center of its neighbor, in the direction of the outward normal to the edge considered. For instance, in Fig. 4.6, the edge in every case is the red line and the blue rectangle shows the length considered for every 1D RP. In both cases the normal distance is given by the sum of half times the length of the two cells considered.

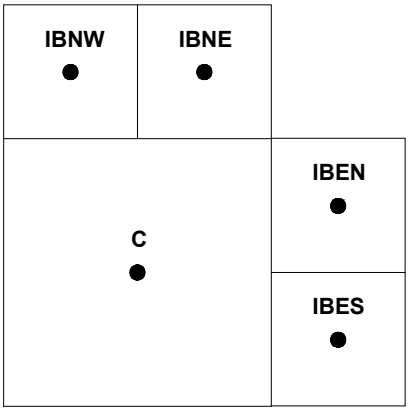
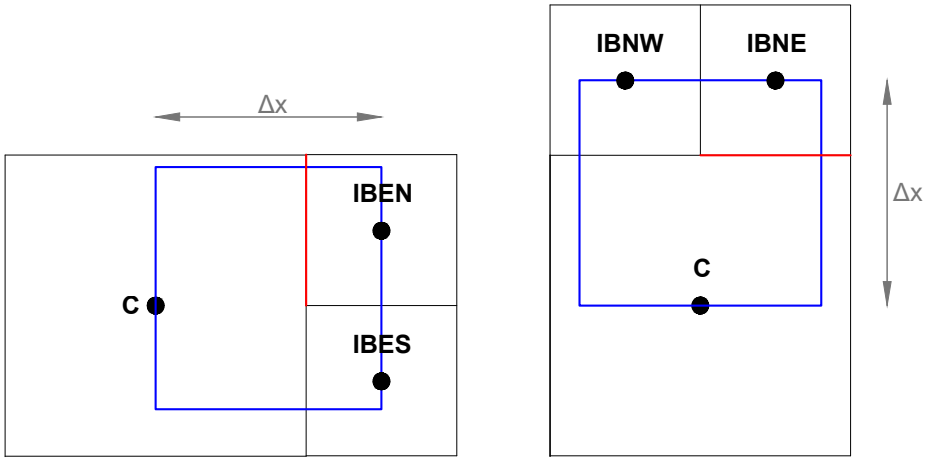


Figure 4.5: Arbitrary cell "C" with two neighbors at its North edge and two neighbors at its East edge, all of them having a lower refinement level.

Last but not least, the mean velocity here is just the average of the normal component of the velocity at both sides of the discontinuity:

$$\bar{u} = \frac{1}{2}(\hat{u}_L + \hat{u}_R) \tag{4.22}$$



(a) Delta x calculation for the RP in the horizontal direction.

(b) Delta x calculation for the RP in the vertical direction.

Figure 4.6: Calculation of the normal distance to an arbitrary edge in the horizontal (Fig. 4.6a) and the vertical direction (Fig. 4.6b).

- **Wind Stress**
  The wind stress is the shear stress exerted by the wind at the interface wind-water, i.e., at the free surface. The projected wind stress is expressed as:

$$\tau_w = c_w \frac{\rho_a}{\rho} u_{wn} |u_w| \tag{4.23}$$

  where c_w is the wind drag coefficient, ρ_a is the air density, u_{wn} is the normal component of the wind velocity and u_w is the wind velocity. In the same way as for the bed friction case, Δx in Eq. 4.14 is calculated as depicted in Fig. 4.6.

- **The divergence form of the thrust term**
  This formulation of the bed slope source term was presented by Valiani and Begnudelli (45) and it has been successfully implemented with different Godunov-type solvers (39, 40). Moreover, the authors claim that this discretization is independent of the convective flux solver.
  In this thesis we test the divergence form of the thrust term with a solver that it is not based on resolving the homogeneous case without considering the presence of source terms, which is the HLLCS. To do so, some modifications are required to the formulation proposed by Valiani and Begnudelli (45). For instance, since they combined this source term discretization with a solver that it is based on resolving the homogeneous case, the length of integration differs in the case of the HLLCS.

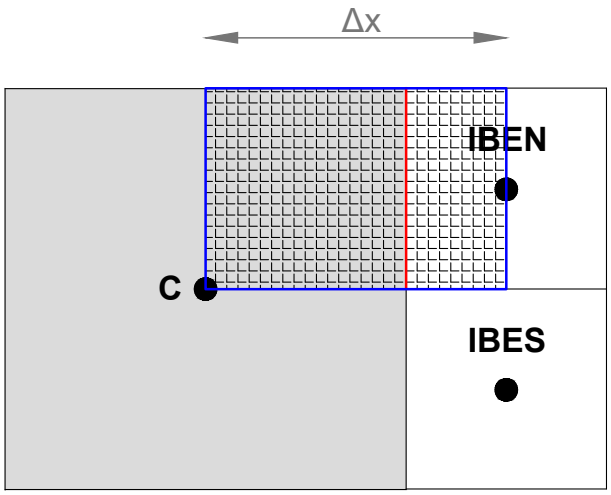


Figure 4.7: Length of integration, Δx, for the calculation of the divergence form of the bed slope source term.

Fig. 4.7 shows the comparison of the area in both cases. The shaded area should be used if implemented as in (45), while the blue rectangle is the area required to be im-

plemented with the HLLCS solver. The thurst term with this formulation is calculated by:

$$S_z = \frac{1}{2} g \left(\hat{\bar{\mathrm{H}}}^2|_R - \hat{\bar{\mathrm{H}}}^2|_L \right) \tag{4.24}$$

with

$$\hat{\bar{\mathrm{H}}}_p = \bar{\eta} - z_p \tag{4.25a}$$

$$\bar{\eta} = \eta_R + \eta_L \tag{4.25b}$$

where $\hat{\bar{\mathrm{H}}}$ is an averaged value of the water depth that is evaluated with Eq. 4.25b. This slightly differs from the evaluation of $\hat{\bar{\mathrm{H}}}$ proposed by Valiani and Begnudelli (45). While they used the free surface elevation at the cell center as the mean value in the control volume, we need to calculate a mean value because the volume of integration contains two different cells. On the other hand, Valiani and Begnudelli (45) had to obtain interpolated values for the bottom elevation and here we use directly the value at the center of both cells.

4. Calculate the wave speeds, for the left, the right and the middle waves.

$$\lambda_L = \begin{cases} \min\left(\widetilde{\lambda}_1, \hat{u}_L - c_L, \hat{u}_R - c_R\right) & \text{if } |\widehat{S}_2| = 0 \\ \widetilde{\lambda}_1 & \text{if } |\widehat{S}_2| \neq 0 \end{cases}$$

$$\lambda_R = \begin{cases} \min\left(\widetilde{\lambda}_2, \hat{u}_L + c_L, \hat{u}_R + c_R\right) & \text{if } |\widehat{S}_2| = 0 \\ \widetilde{\lambda}_2 & \text{if } |\widehat{S}_2| \neq 0 \end{cases} \tag{4.26}$$

with $\widetilde{\lambda}_1 = \widetilde{u} - \widetilde{c}$, $\widetilde{\lambda}_2 = \widetilde{u} + \widetilde{c}$ and $c_p = \sqrt{g h_p}$. The mean celerity, $\widetilde{c}$, is obtained when H_p is replaced by the mean depth ($\widetilde{h}$).

Initially, the sign of the middle wave (λ_*) is unknown. Thus, both the positive and negative middle waves must be calculated. The middle waves are approximated by:

$$\lambda_*^+ = \frac{\lambda_L h_R (\hat{u}_R - \lambda_R) - \lambda_R h_L (\hat{u}_L - \lambda_L) + \lambda_R \lambda_L \overline{H}_1^+}{h_R (\hat{u}_R - \lambda_R) - h_L (\hat{u}_L - \lambda_L) + \lambda_L \overline{H}_1^+} \tag{4.27}$$

$$\lambda_*^- = \frac{\lambda_L h_R (\hat{u}_R - \lambda_R) - \lambda_R h_L (\hat{u}_L - \lambda_L) + \lambda_R \lambda_L \overline{H}_1^-}{h_R (\hat{u}_R - \lambda_R) - h_L (\hat{u}_L - \lambda_L) + \lambda_R \overline{H}_1^-} \tag{4.28}$$

with the following Rankine-Hugoniot relations:

$$\overline{H}^{+} = -\frac{\widehat{S}_2}{(\widetilde{u}-\widetilde{c})(\widetilde{u}+\widetilde{c})}\begin{pmatrix} 1 \\ 0 \\ \hat{v}_L \end{pmatrix} \tag{4.29a}$$

$$\overline{H}^{-} = -\frac{\widehat{S}_2}{(\widetilde{u}-\widetilde{c})(\widetilde{u}+\widetilde{c})}\begin{pmatrix} 1 \\ 0 \\ \hat{v}_R \end{pmatrix} \tag{4.29b}$$

When computing Eqs. 4.27 and 4.28, it may be possible to obtain positive or negative solutions for both estimates, so the following redefinition is required:

$$\lambda_*^+ = \frac{\lambda_*^+ + |\lambda_*^+|}{2} \quad \lambda_*^- = \frac{\lambda_*^- - |\lambda_*^-|}{2} \tag{4.30}$$

If after redefining the middle waves in Eq. 4.30 both estimates are not zero, the middle wave estimate of Roe, $\widetilde{u}$, is considered by the following condition:

$$\lambda_*^+ = \begin{cases} \lambda_*^+ & \text{if } \lambda_*^+\widetilde{u} > 0 \\ 0 & \text{otherwise} \end{cases} \quad \lambda_*^- = \begin{cases} \lambda_*^- & \text{if } \lambda_*^-\widetilde{u} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{4.31}$$

5. Once all the waves are calculated, it is necessary to verify that the inner states are positive. The inner states read:

$$h_R^+ = \left(\frac{h_L(\hat{u}_L - \lambda_L) - \lambda_L \overline{H}_1^+}{\lambda_*^+ - \lambda_L}\right) \tag{4.32}$$

$$h_L^- = \left(\frac{h_R(\hat{u}_R - \lambda_R) - \lambda_R \overline{H}_1^-}{\lambda_*^- - \lambda_R}\right) \tag{4.33}$$

In case that h_R^+ becomes negative when computed using Eq. 4.32 , the source term is reduced to:

$$\widehat{S}_2 = \widetilde{\lambda}_1\widetilde{\lambda}_2\frac{h_R(\hat{u}_R - \lambda_R) + h_L(\lambda_L - \hat{u}_L)}{\lambda_L} \tag{4.34}$$

In case that h_L^- becomes negative when computed using Eq. 4.33, the source term is reduced to:

$$\widehat{S}_2 = \widetilde{\lambda}_1\widetilde{\lambda}_2\frac{h_R(\hat{u}_R - \lambda_R) + h_L(\lambda_L - \hat{u}_L)}{\lambda_R} \tag{4.35}$$

In the presence of wet/dry RP with discontinuous water level surface defined by:

$$h_L + z_L < z_R,\ h_R = 0 \tag{4.36}$$

or

$$h_R + z_R < z_L,\ h_L = 0 \tag{4.37}$$

the normal velocity at the involved cells are set to zero, $\hat{u}_R = \hat{u}_L = 0$. In this way negative depths at the inner regions are avoided, at the time an accurate tracking of the wetting/drying fronts is provided.

6. Calculate the intermediate state vectors as:

$$\widehat{\mathbf{U}}_R^+ = h_R^+ \begin{pmatrix} 1 \\ \lambda_*^+ \\ \hat{v}_L \end{pmatrix} \tag{4.38}$$

$$\widehat{\mathbf{U}}_L^- = h_L^- \begin{pmatrix} 1 \\ \lambda_*^- \\ \hat{v}_R \end{pmatrix} \tag{4.39}$$

7. Estimate the intercell boundary fluxes with the HLLCS solver.

   - If $0 \leq \lambda_L$ then

$$\begin{aligned} \mathbf{F}_{i+1/2}^{\text{HLLCS-}} &= \widehat{\mathbf{F}}_L \\ \mathbf{F}_{i+1/2}^{\text{HLLCS+}} &= \widehat{\mathbf{F}}_L + \widehat{\mathbf{S}}_{i+1/2} \end{aligned} \tag{4.40}$$

   - If $0 \geq \lambda_R$ then

$$\begin{aligned} \mathbf{F}_{i+1/2}^{\text{HLLCS-}} &= \widehat{\mathbf{F}}_R - \widehat{\mathbf{S}}_{i+1/2} \\ \mathbf{F}_{i+1/2}^{\text{HLLCS+}} &= \widehat{\mathbf{F}}_R \end{aligned} \tag{4.41}$$

- If $\lambda_L \leq 0 \leq \lambda_R$ then

  If $\lambda_*^+ > 0$
  Compute $\widehat{\mathbf{U}}_R^+$ using Eq. 4.38, $\widehat{\mathbf{U}}_L^-$ as $\widehat{\mathbf{U}}_L^- = \widehat{\mathbf{U}}_R^+ - \overline{\mathbf{H}}_{i+1/2}^+$ and define the fluxes as:

$$\begin{aligned} \mathbf{F}_{i+1/2}^{\text{HLLCS-}} &= \widehat{\mathbf{F}}_L + \lambda_L \left(\widehat{\mathbf{U}}_L^- - \widehat{\mathbf{U}}_L \right) \\ \mathbf{F}_{i+1/2}^{\text{HLLCS+}} &= \mathbf{F}_{i+1/2}^{\text{HLLCS-}} + \widehat{\mathbf{S}}_{i+1/2} \end{aligned} \tag{4.42}$$

  If $\lambda_*^- < 0$
  Compute $\widehat{\mathbf{U}}_L^-$ using Eq. 4.39, $\widehat{\mathbf{U}}_R^+$ as $\widehat{\mathbf{U}}_R^+ = \widehat{\mathbf{U}}_L^- + \overline{\mathbf{H}}_{i+1/2}^+$ and define the fluxes as:

$$\begin{aligned} \mathbf{F}_{i+1/2}^{\text{HLLCS+}} &= \widehat{\mathbf{F}}_R + \lambda_R \left(\widehat{\mathbf{U}}_R^+ - \widehat{\mathbf{U}}_R \right) \\ \mathbf{F}_{i+1/2}^{\text{HLLCS-}} &= \mathbf{F}_{i+1/2}^{\text{HLLCS+}} - \widehat{\mathbf{S}}_{i+1/2} \end{aligned} \tag{4.43}$$

8. Calculate the time step for the time integration. As mentioned above, the time step is restricted by the CFL condition, thus is calculated by:

$$\Delta t \leq \text{CFL} \left(\frac{\Delta x_{min}}{\max(|\lambda_L|, |\lambda_R|)} \right) \tag{4.44}$$

   with CFL ≤ 1 for the one dimensional case and CFL ≤ 0.5 for the two dimensional case.

   In a quadtree mesh, Δx_{min} is given by $min\,(X_R, X_L)$, where X_R and X_L are the length of the two cells at every side of the discontinuity (see Fig. 4.3), accordingly.

Finally, it remains to update the solution with Eq. 4.4 based on the projected fluxes and variables:

$$\widehat{\mathbf{U}}_i^{n+1} = \widehat{\mathbf{U}}_i^n - \sum_{k=1}^{\text{NE}} \left(\mathbf{T}^{-1} \widehat{\mathbf{F}} \right)_{i,k}^- \frac{l_k \Delta t}{A_i} \tag{4.45}$$

Chapter 5

Test Cases and Validation

This chapter presents the numerical results of implementing the solver for the solution of different hydraulic engineering problems. The chapter is divided in three main sections that cover analytical test cases, laboratory test cases and a real test case, respectively.

5.1 Analytical Test Cases

In this section the solver is implemented for the solution of different classical problems, i.e. theoretical problems with available analytical solutions, that are commonly used to assess the performance of numerical schemes.

5.1.1 Lake at rest with an emerged bump

This test case allows to verify that the numerical scheme satisfies the exact conservation property (4), also known as exact C-property, over a non-flat bottom. Here the difficulty arises from the geometrical source term due to the variable bottom topography. In the absence of any additional source term or perturbation, the initial rest state must be preserved over time.

In the one-dimensional case, the topography is a parabola and in the two-dimensional case it is a paraboloid, which is described by the equation:

$$z(x,y) = \mathbf{max}\left[0,\, 0.25 - 0.004\left((x-5)^2 + (y-5)^2\right)\right] \text{ m} \tag{5.1}$$

as depicted in Fig. 5.1. We consider the 2D-SWEs subject to the following ICs:

$$H(x,y,0) = \mathbf{max}(0.1 - z,\, 0) \text{ m} \;\text{ and }\; U(x,y,0) = V(x,y,0) = 0 \text{ m/s}$$

and the BCs:

$$\begin{cases} H = 0.1 \text{ m} \\ q_x = q_y = 0.0 \text{ m}^2/\text{s} \end{cases} \tag{5.2}$$

where $q_x = HU$ and $q_y = HV$ are the unit discharge, i.e., the flow rate per unit width in each direction.

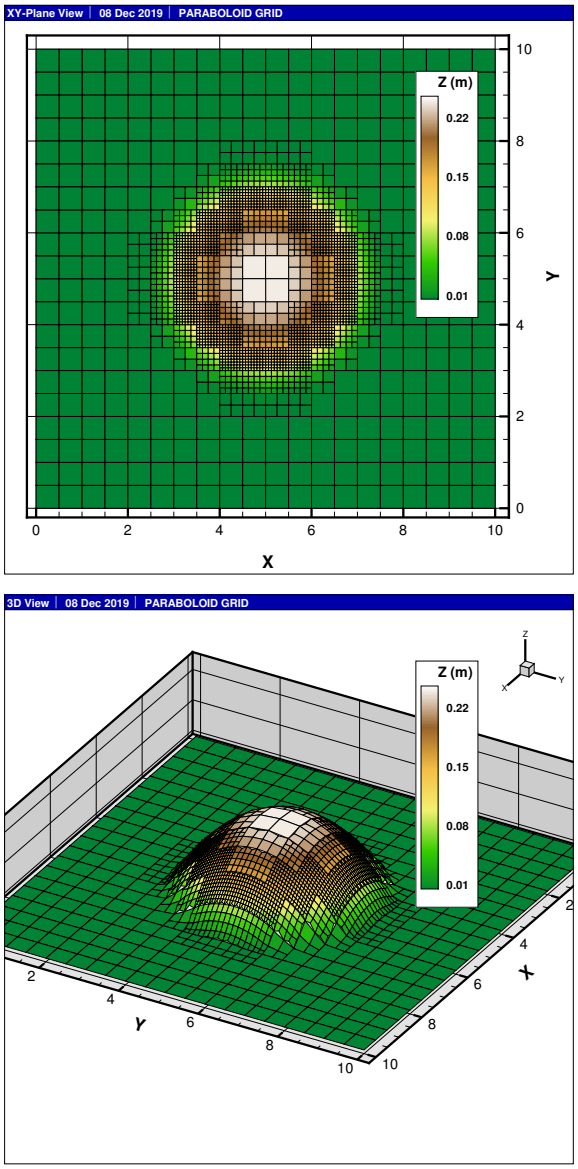


Figure 5.1: XY-Plane view (top) and 3D view (bottom) of a lake with an irregular bottom profile.

For the emerged bump, the initial water depth is lower than the highest point of the bump. The initial condition and, also, the solution to this test case is shown in Fig. 5.2. After 1000 seconds of simulation the rest state is preserved and maintained over time.

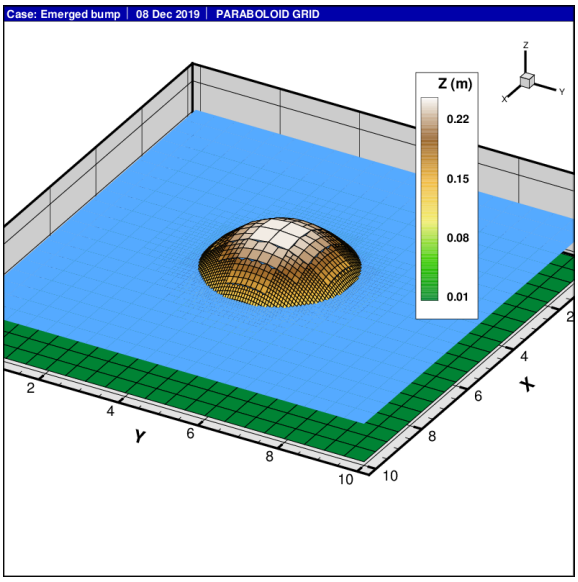


Figure 5.2: Lake at rest with an emerged bump.

5.1.2 Lake at rest with an immersed bump

The bottom profile of this problem is given by a paraboloid as depicted in Fig. 5.1. However in this case the initial water depth is higher than the crest of the bump, thus the bump is totally immersed in the lake. Again the initial rest state is preserved over time.

5.1.3 Steady flow over a hump

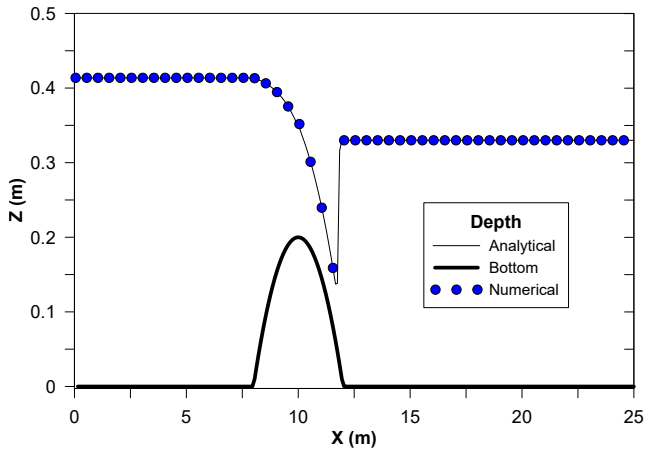
The channel has a length of 25 meters and the bed level is given by:

$$z(x) = \begin{cases} 0.2 - 0.05(x-10)^2 \text{ m} & \text{if } 8 \leq x \leq 12 \\ 0 \text{ m} & \text{otherwise} \end{cases} \tag{5.3}$$

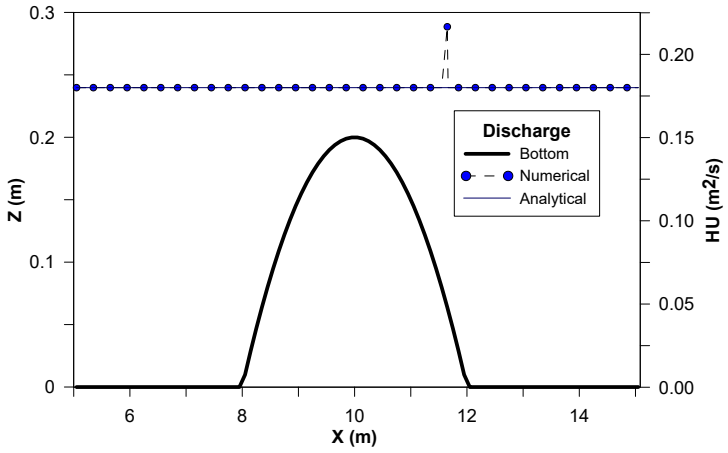
Depending on the boundary conditions, there can be different flow regimes. Only the transcritical flow with a shock is considered here, since it is the one presenting more difficulties.

The discharge, q_x = 0.18 m^2/s, is imposed upstream and the water depth, $H = 0.33$ m, is imposed downstream. The initial conditions are taken as:

$$H(x,0) = 0.5 - z(x) \text{ m} \quad \text{and} \quad U(x,0) = 0\text{m/s} \tag{5.4}$$



(a) Water depth level, analytical solution (-) vs numerical solution (-•-).



(b) Unit discharge, analytical solution (-) vs numerical solution (-•-).

Figure 5.3: Steady state of a transcritical flow with a shock. Numerical solution for the water depth level (Fig. 5.3a) and the unit discharge (Fig. 5.3b), $\Delta x = 0.10$ m.

The analytical solution is provided by Delestre et al. (14). The water depth is computed by solving the following system of equations:

$$\begin{cases} H(x)^3 + \left(z(x) - \dfrac{q_0^2}{2gH_M^2} - H_M - z_M \right) H(x)^2 + \dfrac{q_0^2}{2g} = 0 & \text{if } x \leq x_{shock} \quad (5.5a) \\ H(x)^3 + \left(z(x) - \dfrac{q_0^2}{2gH_L^2} - H_L \right) H(x)^2 + \dfrac{q_0^2}{2g} = 0 & \text{if } x > x_{shock} \quad (5.5b) \\ q_0^2 \left(\dfrac{1}{H_1} - \dfrac{1}{H_2} \right) + \dfrac{1}{2} g \left(H_1^2 - H_2^2 \right) = 0 & (5.5c) \end{cases}$$

In Eq. 5.5, $z_M = \max_{x \in [0,L]} z$, H_M is the corresponding water height. $H_L = H(x = L)$ and $H_1 = H(x_{shock}^-)$, $H_2 = H(x_{shock}^+)$ are the water heights upstream and downstream, respectively. The water height is determined with Eqs. 5.5a and 5.5b, while the shock location is determined with Eq. 5.5c.

Figure 5.3 shows the numerical solution using 250 cells for the water level surface (Fig. 5.3a) and the unit discharge (Fig. 5.3b). For the water depth, there is an improvement of accuracy since the error in L_1 norm is reduced by a factor of 0.5, when the results are compared with the numerical solution obtained by Murillo and García-Navarro (29). Nevertheless, the computed unit discharge is more accurate when using the source term formulation in (29). Broadly speaking, the overall accuracy in this test case is improved with the present formulation, except at the shock location ($x = 11.6655$ m).

5.1.4 Open channel flow including friction

The performance of the HLLCS solver with the divergence form of the thrust term is assesed for the calculation of the steady state solution of a subcritical flow with friction in a straight channel. The results are compared with the analytical solution derived by MacDonald et al. (28). The channel has a length of 1000 m and a constant width $B = 10$ M. The discharge The longitudinal bed slope is given by:

$$S_0(x) = \left(1 - \frac{Q^2 B}{g(h(x)B)^3} \right) h_x(x) + \frac{Q^2 n^2 (B + 2h(x))^{4/3}}{(h(x)B)^{10/3}} \quad (5.6)$$

The discharge, $Q = 20\,\mathrm{m}^3/\mathrm{s}$, is imposed upstream. Thus by Eq. 5.6, with $n = 0.03\,\mathrm{s\,m}^{-1/3}$, the longitudinal bed slope reads:

$$S_0(x) = \left[1 - \frac{4}{gh(x)^3} \right] h_x(x) + 0.36 \frac{[10 + 2h(x)]^{4/3}}{[10h(x)]^{10/3}} \quad (5.7)$$

where

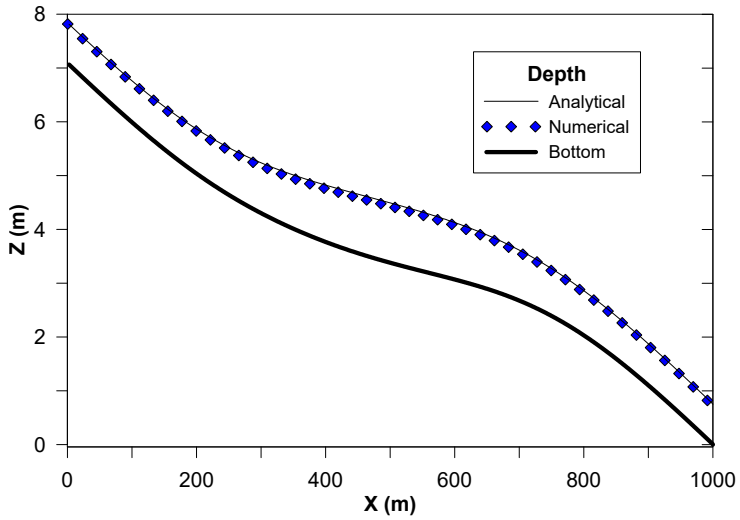
$$h(x) = \left(\frac{4}{g}\right)^{1/3} \left\{1 + \frac{1}{2}\exp\left[-16\left(\frac{x}{1000} - \frac{1}{2}\right)^2\right]\right\} \tag{5.8a}$$

$$h_x(x) = -\left(\frac{4}{g}\right)^{1/3} \frac{2}{125}\left(\frac{x}{1000} - \frac{1}{2}\right)\exp\left[-16\left(\frac{x}{1000} - \frac{1}{2}\right)^2\right] \tag{5.8b}$$

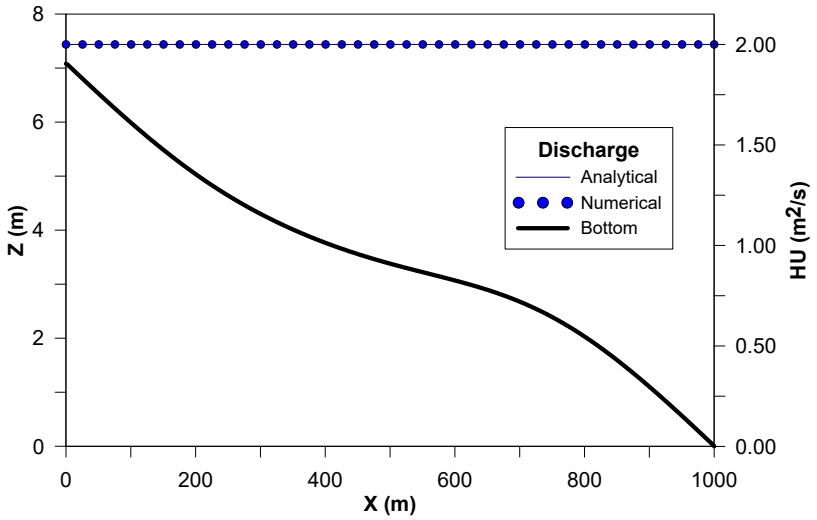
Friction loss is computed using the Glaucker-Manning formula, $cf = gn^2/R_h^{1/3}$, where R_h is the mean hydraulic radius. The flow is subcritical at both the inflow and the outflow, the later with an imposed water depth of 0.748409 m.

The numerical solution is compared with the exact solution given by Eq. 5.8a, as depicted in Fig. 5.4. Both, the water depth and the unit discharge are in correspondence with the analytical solution derived by MacDonald et al. (28). The water depth is accurately predicted (Fig. 5.4a), except at the boundaries where small discrepancies appear. The unit discharge is predicted without any error (see Fig. 5.4b).

Although not presented here, the solution was also computed for smaller Δx grids without observing an improvement of accuracy. The water depth only seems to be more accurate as the roughness coefficient decreases, thus probably the errors are due to overestimation of the friction loss.



(a) Water depth level, analytical solution (-) vs numerical solution (-•-).



(b) Unit discharge, analytical solution (-) vs numerical solution (-•-).

Figure 5.4: Steady state of a subcritical flow in a frictional straight channel. Numerical solution for the water depth level (Fig. 5.4a) and the unit discharge (Fig. 5.4b), $\Delta x = 1.0$ m.

5.2 Laboratory Test Cases

This section presents a comparison of laboratory experimental results and numerical solutions obtained with the solver. The experiments, described in detail in Appendix B, broadly consisted of measuring the run up reached by two different waves patterns, i.e. a solitary wave and a train of regular waves, onto a slope covered with different materials.

Here, one of the boundaries is located at the position of sensor 1 and so the water depth records provided by this sensor are imposed there (see the layout of the experiments in Fig. B.3). The other one is an open boundary located at the extreme of the ramp.

5.2.1 Solitary Wave

Figure 5.5 depicts the water level computed at the locations of sensor 3 (top) and sensor 7 (bottom) for the run up of a solitary wave onto a slope covered with acrylic material. The model predicts correctly the water depth at the flat part of the channel. At the slope, while a good estimation of the water depth is obtained, the solution profile is smoother than the laboratory one. This smoothing is

a result of the first order accuracy of the scheme. As for the run up, the model predicts it with an average error of $\pm$ 10 %.

The results obtained for the slope covered with acrylic material are similar to those obtained for the slope covered with sand paper of different roughness. The profile at the same locations was compared and it is also observed a smoothing of the water depth profile at sensor 7, but the error for the run up estimation decreases to $\pm$ 5 %. This is also a result of the dissipative behaviour of first order schemes. However only the results for the case of acrylic material coating are presented, given the similarity with the two other materials.

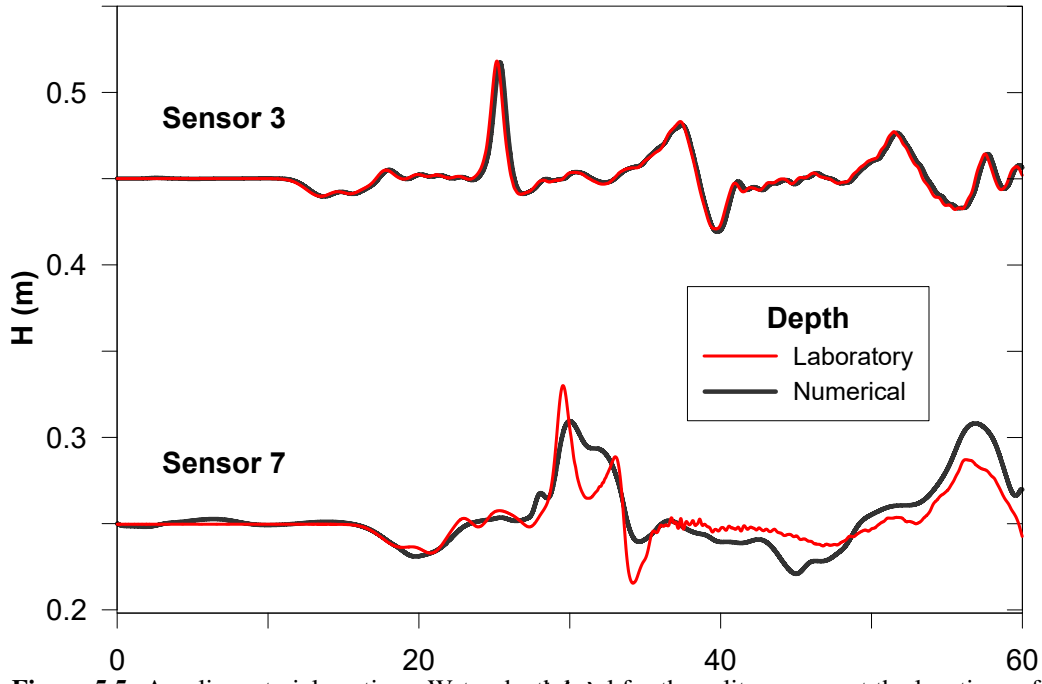
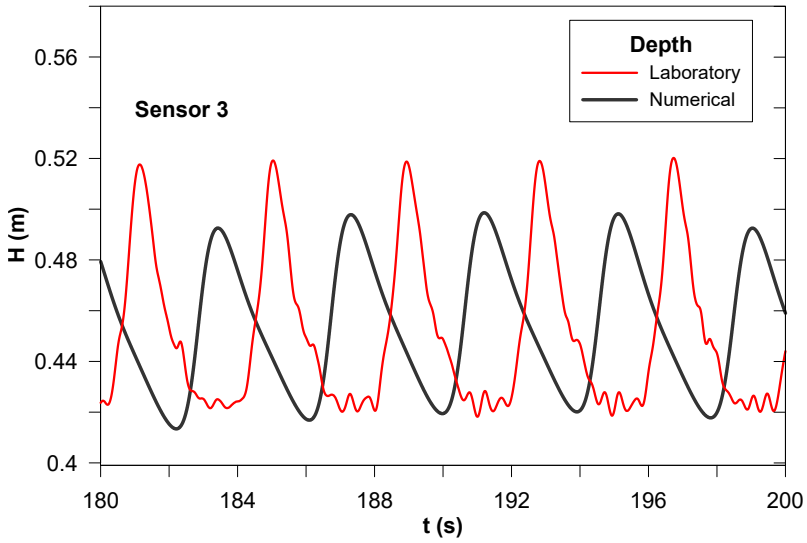


Figure 5.5: Acrylic material coating - Water depth level for the solitary wave at the locations of sensor 3 (top) and sensor 7 (bottom). In every case the red line is obtained directly from laboratory measurements.

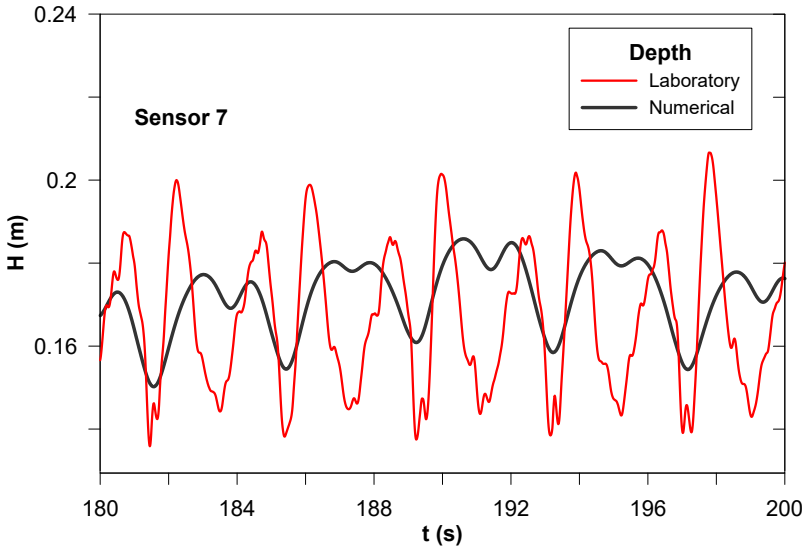
5.2.2 Train of Regular Waves

Again, only the slope covered with acrylic material is shown since the results are very similar for the three materials tested. Section 5.2.2 depicts the water depth level at the locations of sensor 3 and sensor 7, respectively. Smoothing is also observed in this case and moreover, the wave profile at sensor 3 is slightly lagged with respect to the measurements profile as a consequence of the reflection present in this experiment.

As expected, the error for the estimation of the run up also increases to a $\pm$ 15 in the acrylic material coating case and $\pm$ 10 in the sand paper cases.



(a) Water depth level at location of sensor 3.



(b) Water depth level at location of sensor 7.

Figure 5.6: Acrylic material coating - Water depth level at location of sensor 3 and sensor 7 for a train of regular waves. In every case the red line is obtained directly from laboratory measurements.

5.3 Two-Dimensional Test Cases

This is the last section of this chapter and it includes a 2D problem with analytical solution and a real test case in which the numerical solution obtained with the solver is compared with field measurements.

5.3.1 Wind induced circulation in a circular basin

The physical domain is a circular basin where motion is generated by the action of an uniform wind stress applied to the free surface. The radius of the circular basin, R_o, is 193.2 m and the water depth contours (Figure 5.7) are given by

$$h(x) = \frac{1}{1.3}\left(\frac{1}{2} + \sqrt{\frac{1}{2} - \frac{r_b}{R_o}}\right) \tag{5.9}$$

where r_b is the radial distance from the center of the basin to the point of interest. This case was selected since the results can be compared to the analytical solution provided by Kranenburg (24) and the numerical simulations carried out by Rogers et al. (36), who also used the same test case.

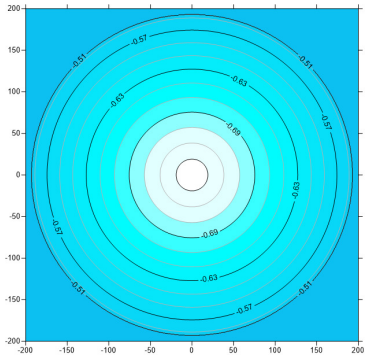


Figure 5.7: Still water depth contours for the circular basin (m).

At the initial time, the water is at rest. The uniform wind stress applied, from west to east, is given by:

$$\begin{cases} \tau_w = \tau_{ss}\left(-2\left(\frac{t}{T_r}\right)^2 + 3\left(\frac{t}{T_r}\right)^2\right) & t < T_r \\ \tau_w = \tau_{ss} & t \geq T_r \end{cases} \tag{5.10}$$

where t is time and T_r is the time required for the flow to be developed. Here T_r is set equal to 1000 s and the maximum wind stress, τ_{ss}, is set to 0.02 N m^{-2}.

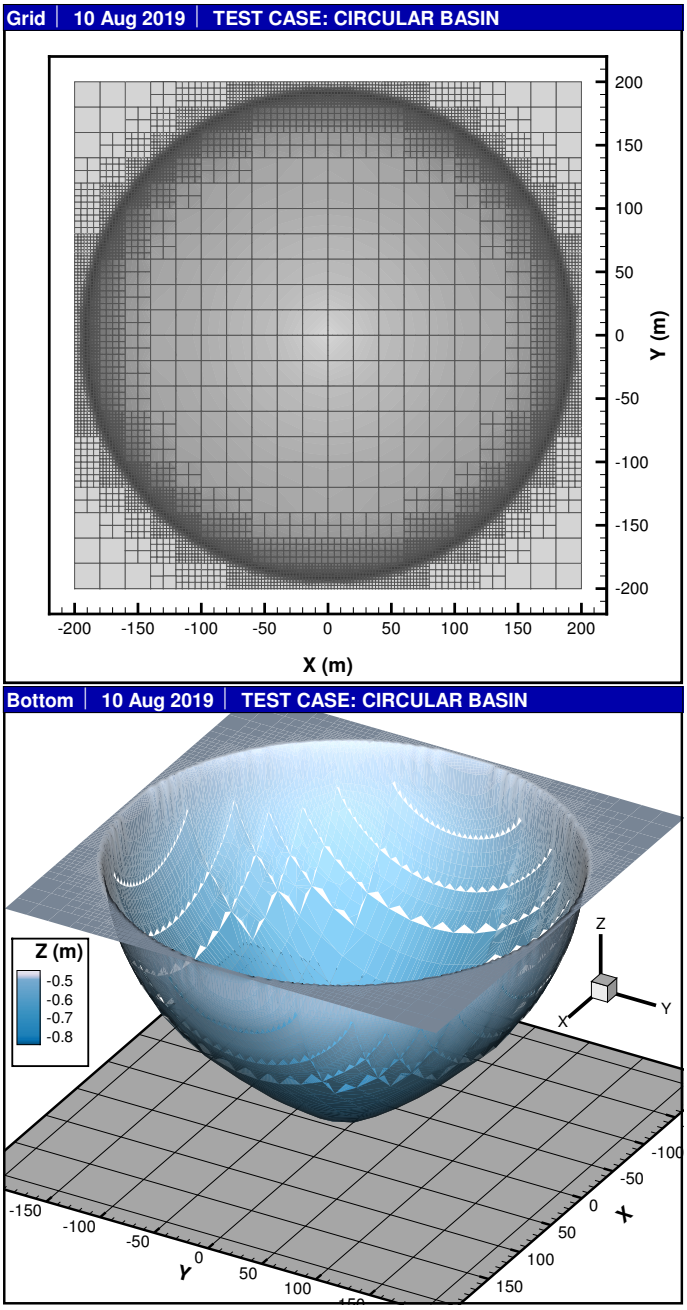


Figure 5.8: Multi-Quadtree grid for the circular basin case (top) and 3D-View of the still water level contours (bottom).

This test case is useful to show the application and versatility of the proposed multi-quadtree grid. It can be observed (Fig. 5.8) that the circular geometry can be represented thanks to the quadtree approach.

Here the mesh was generated from a regular mesh of $\Delta x = 20$m. At an initial step only the regular mesh is considered, with the points located on the boundary being used as "seed points". At a second step, all the mother cells which contain at least one seed point, are refined to the desired level. Then, regularization takes place, i.e., a cell can not be larger than two times the size of its neighbors. Figure 5.9 shows the boundary represented with a red line.Also at these points, the no-slip BCs were imposed. The shaded cells in Fig. 5.9 are outside the physical domain and therefore they are outside of the numerical domain as well, i.e., they play no role in the calculation.

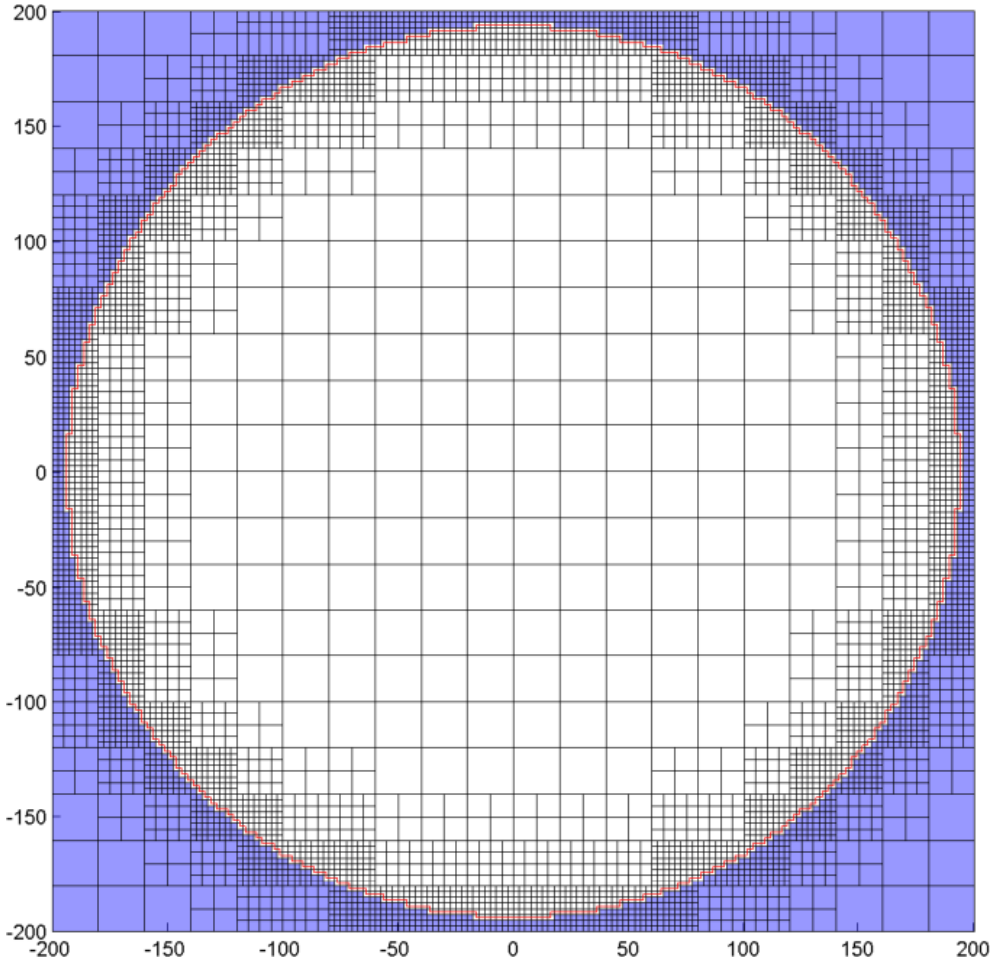


Figure 5.9: Mesh with boundary delineation. The shaded cells are not considered in the calculation, since the no-slip condition is imposed at the boundary.

The solution for the velocity field is depicted in Fig. 5.10. A pair of counter-rotating gyres can be discerned, with the flow directed against the direction of the wind at the deepest part of the basin. The flow patterns are accurately predicted and they are coincident with the analytical solution presented by Kranenburg (24).

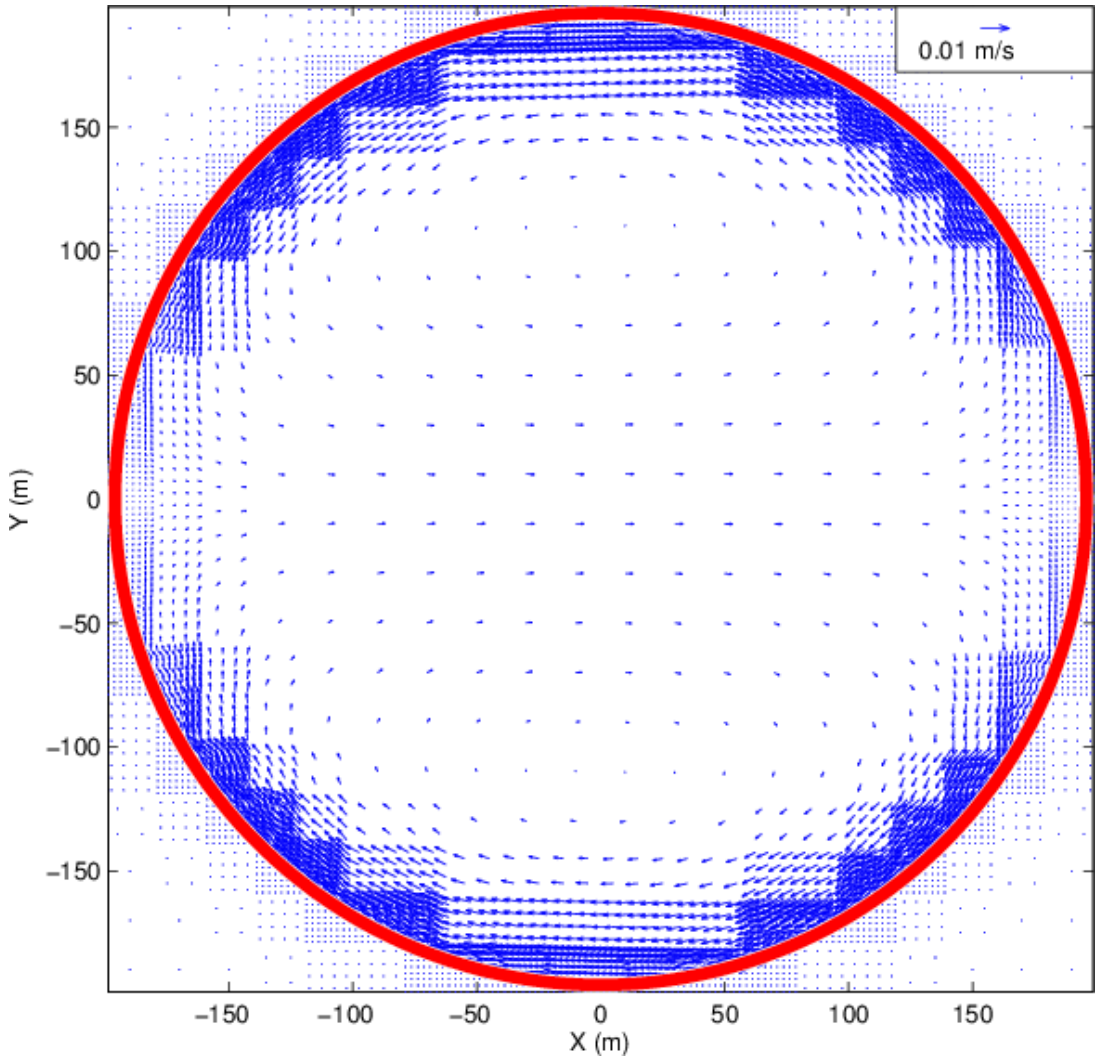
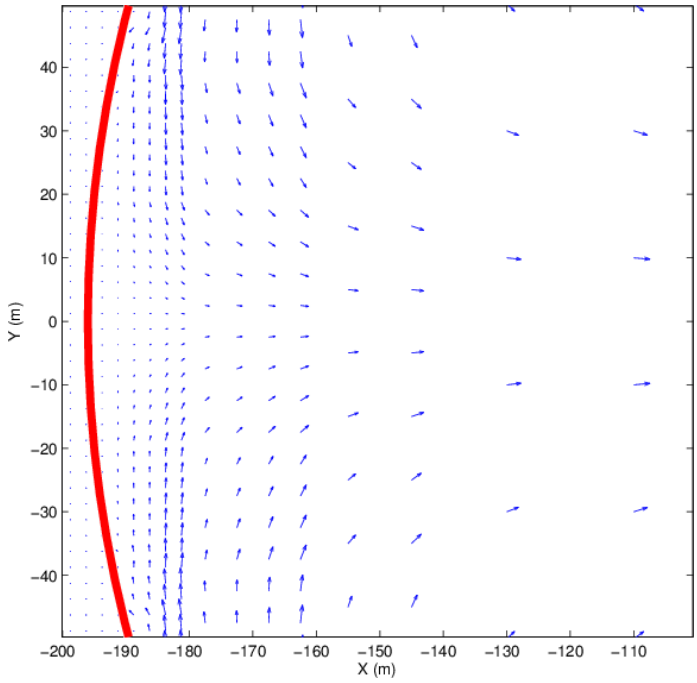
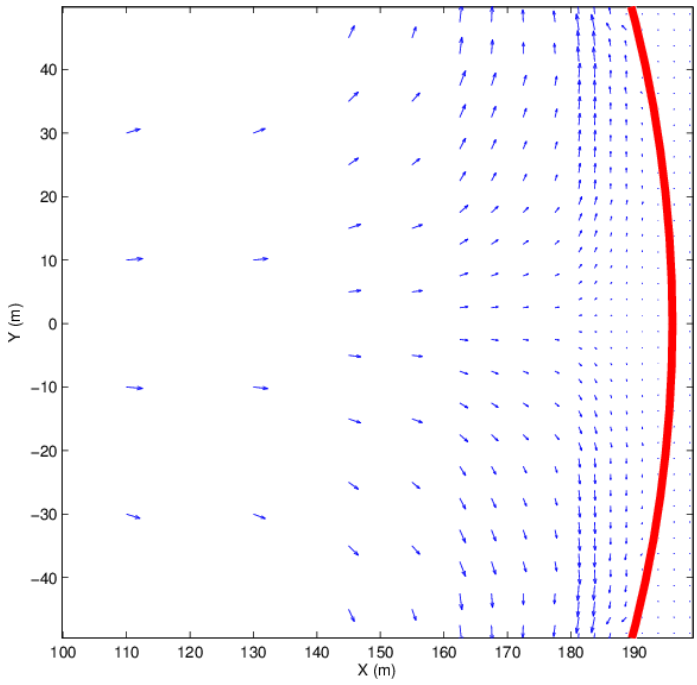


Figure 5.10: Velocity field for the wind induced circulation in a circular basin. The solution is a pair of counter-rotating gyres with the flow across the middle of the basin directed against the direction of the wind (24).

This is better visualized in Fig. 5.11, which shows the velocity field at the West and the East cost of the circular basin as depicted in Figs. 5.11a and 5.11b, respectively.



(a) Vector field at the West coast of the circular basin.



(b) Vector field at the East coast of the circular basin.

Figure 5.11: Vector field at two portions of the circular basin.

The results are also coincident with those obtained by Rogers et al. (36), since no kink was observed at the center of the basin either. Moreover, the velocity field obtained is smaller in magnitude than the one determined by the analytical solution, which is due to the first order accuracy of the scheme.

5.3.2 Real test case: "La Mancha" lagoon

The last test case is the tidal flow in "La Mancha" coastal lagoon which is located in Veracruz, Mexico (see Fig. 5.12). This test case allows to show the applicability of the MQT-mesh and the ability of the solver to treat wetting-drying fronts as well.

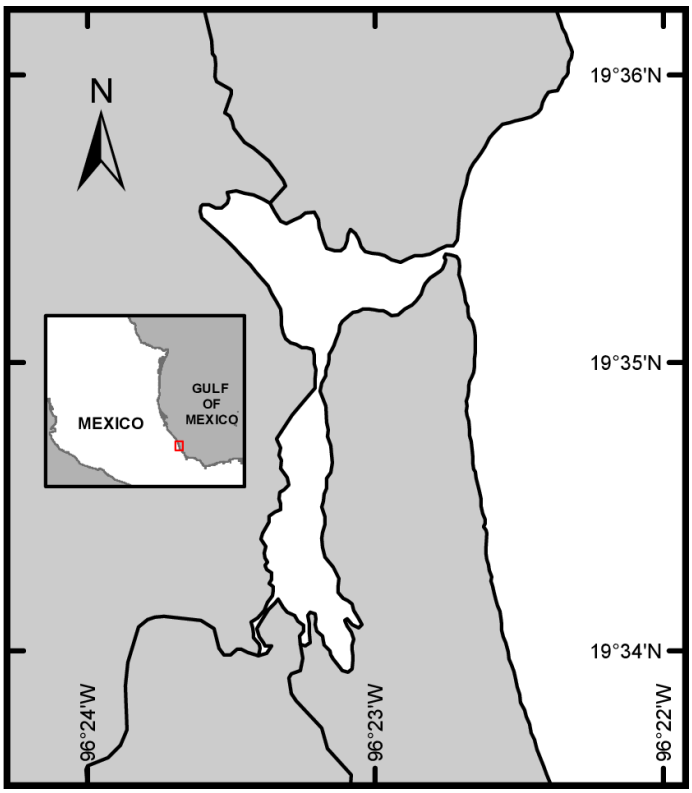


Figure 5.12: Location of La Mancha Lagoon, Veracruz.
Source: Taken from Rivera et al. (34).

The bathymetry and field measurements were obtained from site surveys carried out by Chávez et al. (8) in 2014. The model itself needs to be calibrated, therefore the necessity to count on such field measurements. The selected data, free surface level measurements and discharges, belong to a survey carried out during November 2014.

Tide is used as forcing at the East boundary. The tide data are obtained from the predictions provided by CICESE (10), which in turn are obtained from hourly historical see levels records. The discharge measured on-site from Caño Gallegos basin, Q = 0.45 $m^3 s^{-1}$, is imposed at the south boundary.

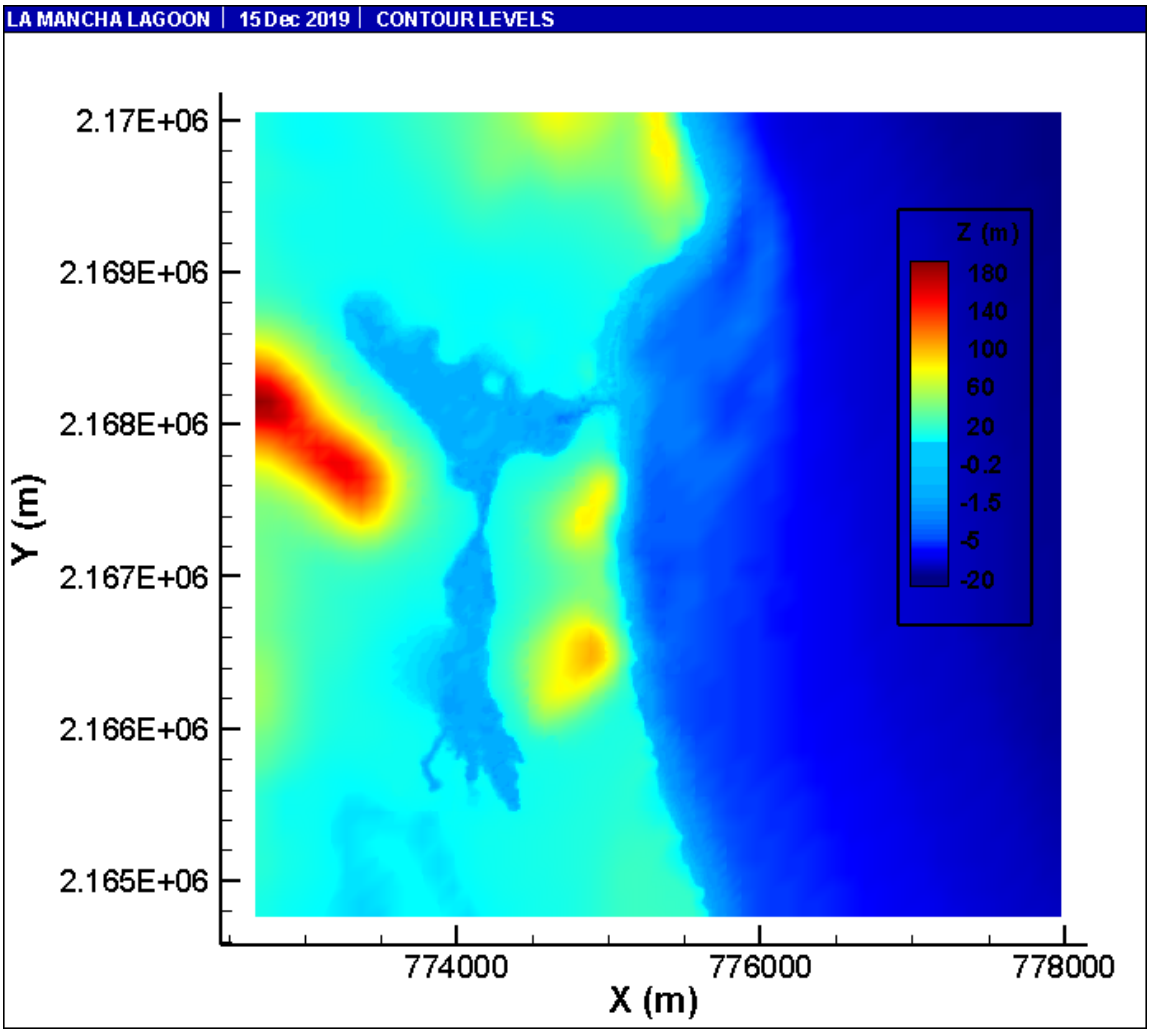


Figure 5.13: Bathymetry in the region of La Mancha Lagoon.

A regular mesh with $\Delta x = 100$ m is generated at an initial stage. The points belonging to the zero elevation contour are used as seed points. Here, the relative refinement level is set to 3. Figure 5.14 depicts the resulting mesh, with cell sizes ranging from 100 m to 12.5 m. Fig. 5.15 depicts a zoomed in view of the grid at the upper part of the lagoon, in order to provide a better illustration of the cells arrangement.

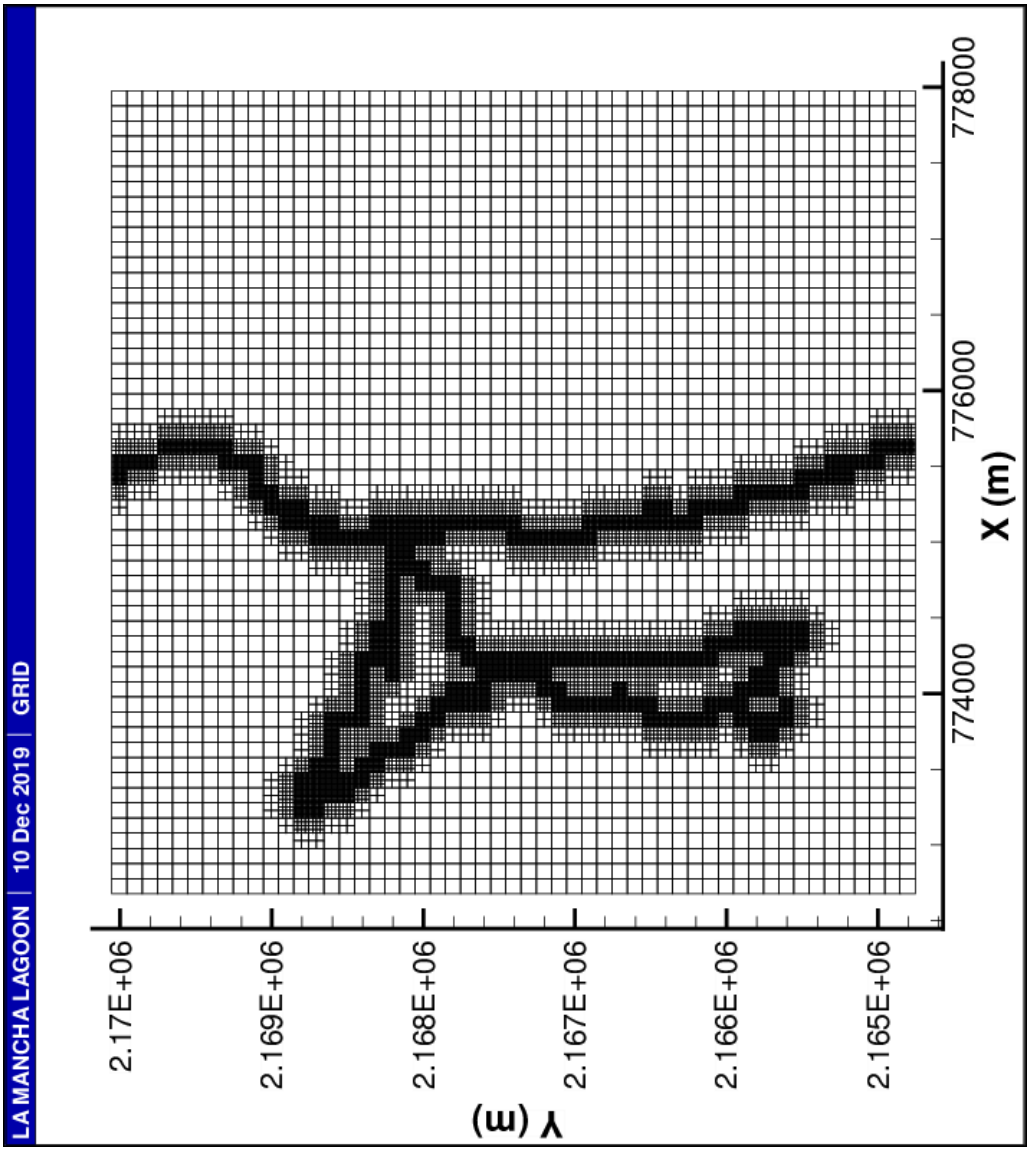


Figure 5.14: MQT-Mesh for La Mancha Lagoon. The maximum level of refinement is 3, which is a figure relative to the initial grid size.

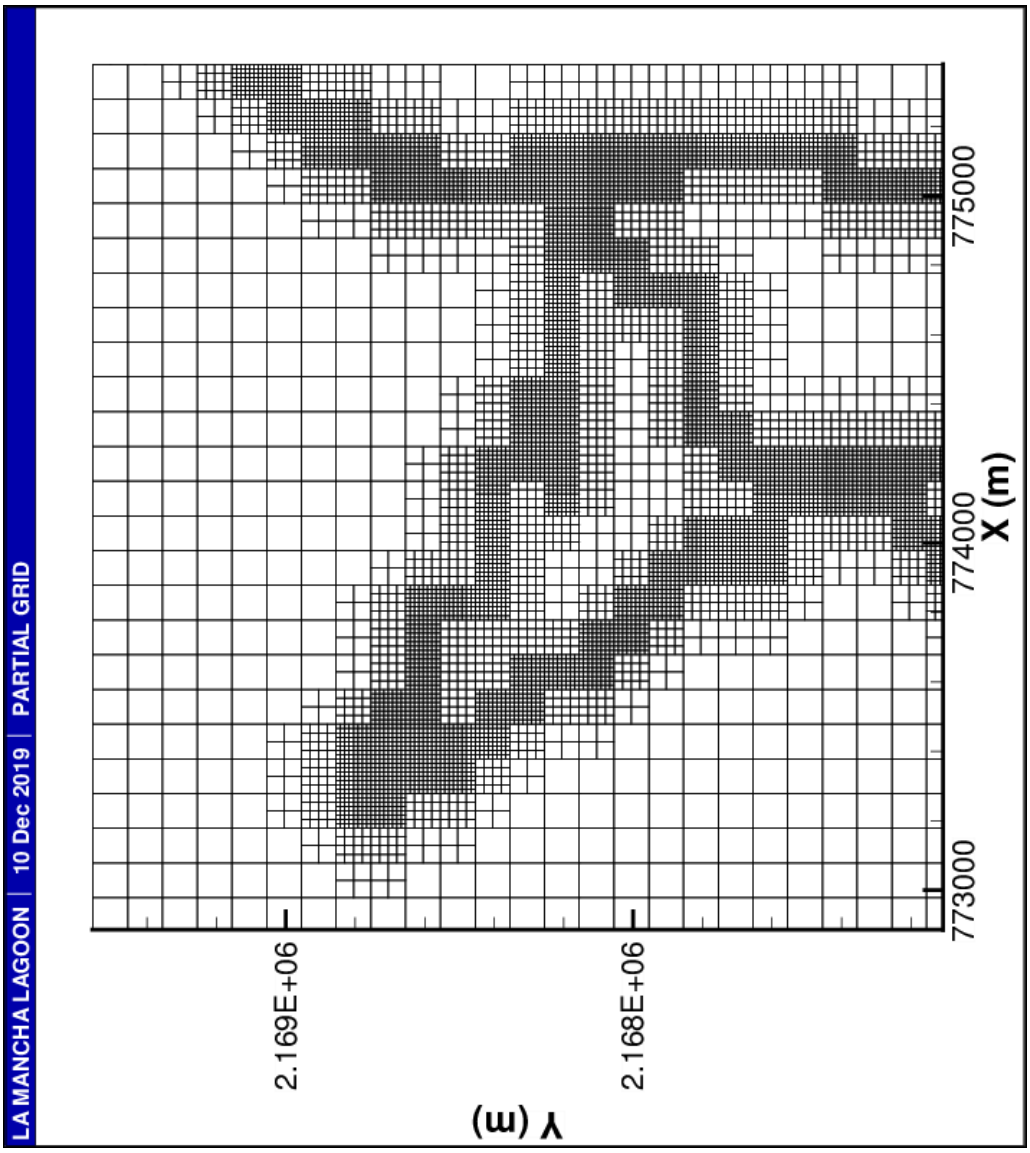


Figure 5.15: Zoomed in view of the MQT-mesh at the upper part of La Mancha Lagoon.

The free surface level predicted by the model is compared with the field measurements in Fig. 5.16. Other than a slight smoothing in the profile, it is coincident with the measurements.

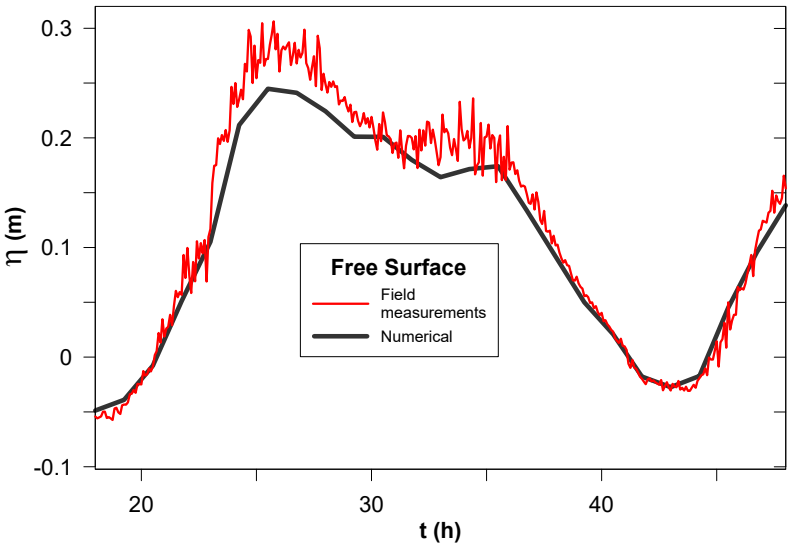


Figure 5.16: Comparisson of the numerical free surface level and field measurements at the inlet of La Mancha Lagoon during an ebb cycle - November 14, 2014.

Figure 5.17 shows an instant picture of the velocity field during and ebb cycle in La Mancha Lagoon. In general the magnitude of the velocities is low, which is coincident with the measurements. The only points where they seem to increase are at the inlet and at the narrow part of the lagoon, also in agreement with the field data.Figure 5.18 and Fig. 5.19 depict a zoomed view of the instant velocity field at both the inlet and the narrow part of the lagoon, respectively.

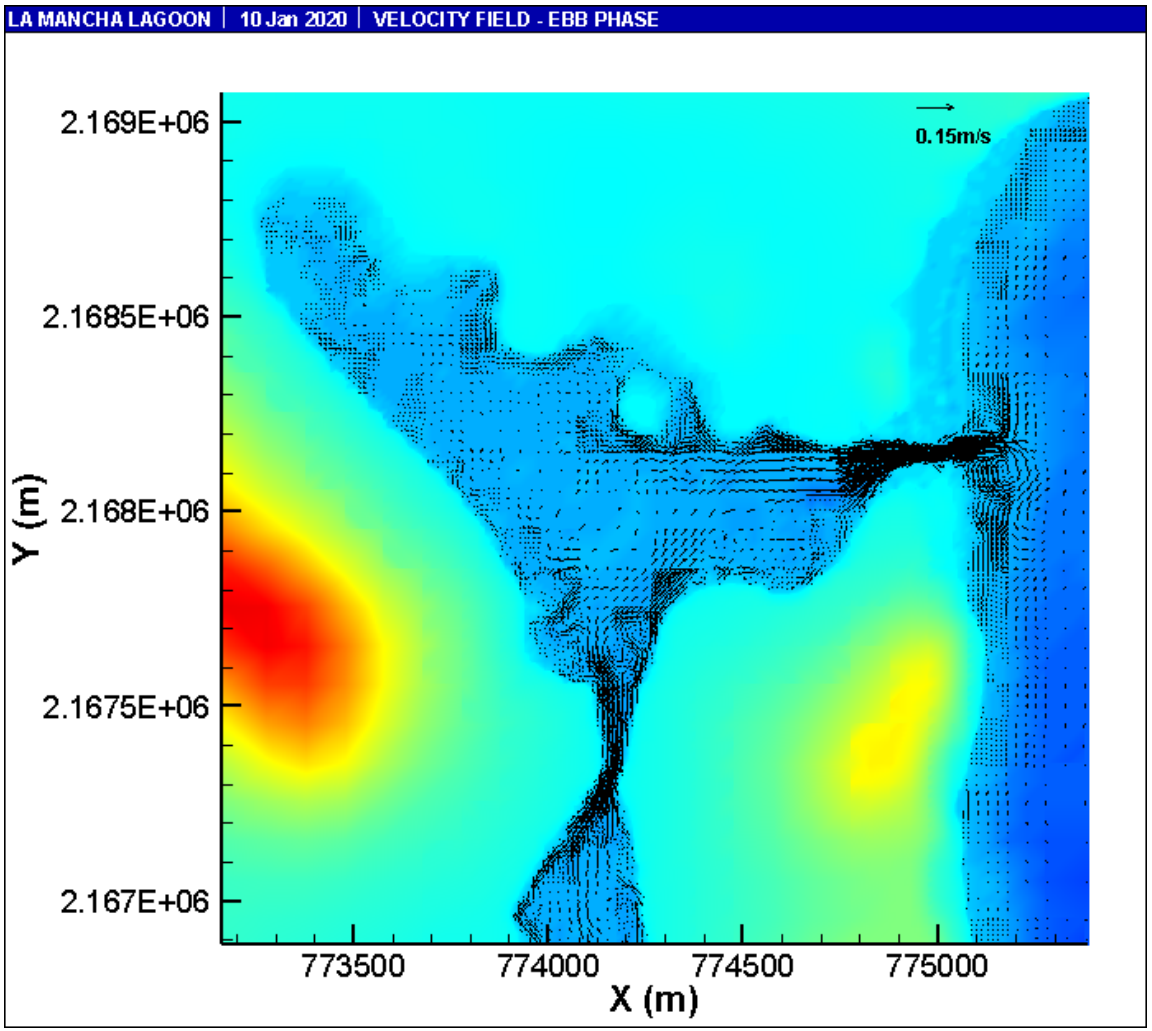


Figure 5.17: Instant velocity field during an ebb cycle in La Mancha Lagoon.

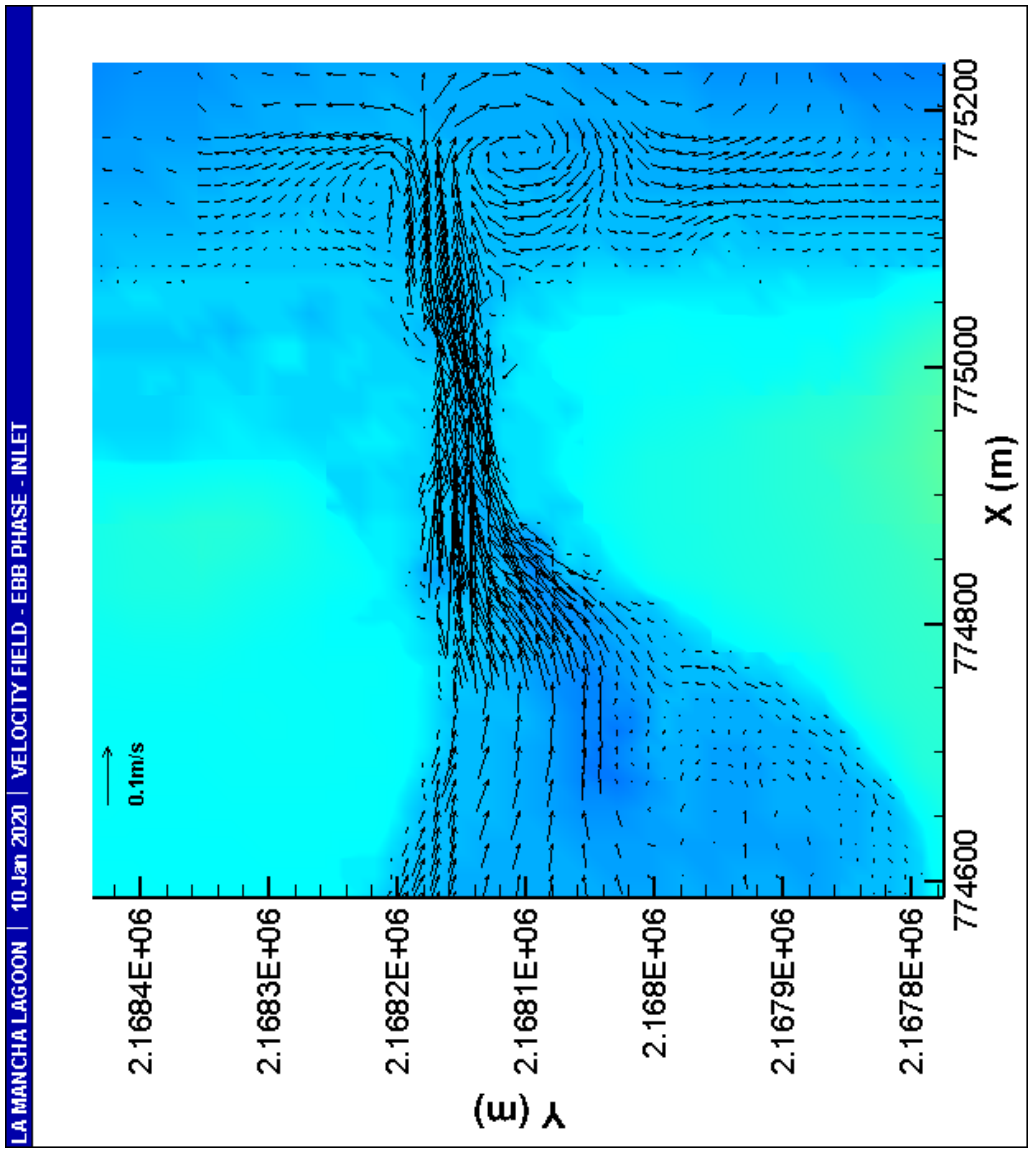


Figure 5.18: Zoomed in view of the velocity field at the inlet of La Mancha Lagoon.

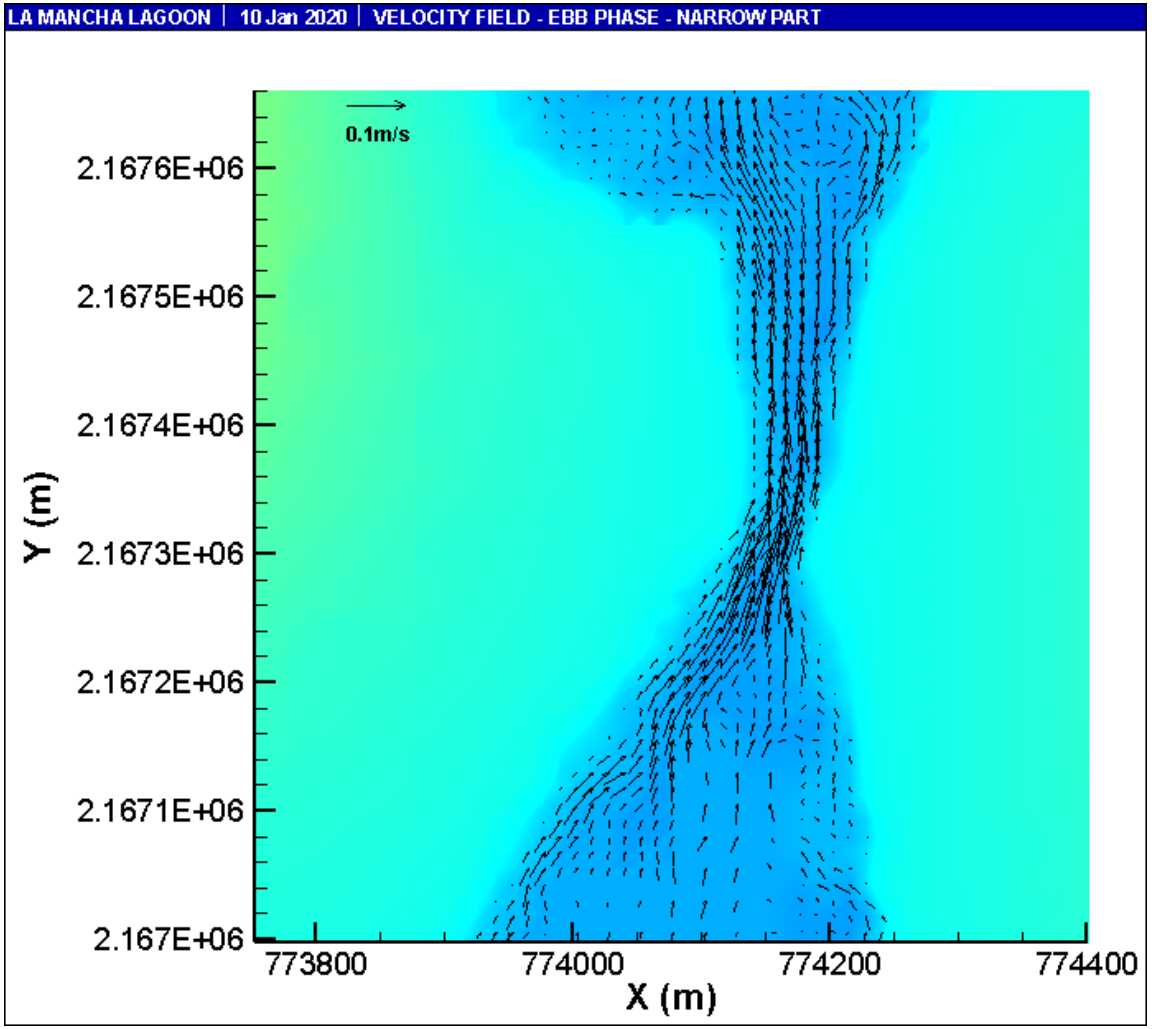


Figure 5.19: Zoomed in view of the velocity field at the narrow part of La Mancha Lagoon, i.e., at the transition zone between the North and the South part of the lagoon.

Chapter 6

Conclusions

Two of the most important aspects of numerical modelling were addressed in this thesis: the mesh generation and the implementation of a well-balanced solver. Therefore the first section of this chapter summarizes the main aspects of the mesh algorithm developed, while the second one the main aspects of the numerical solver with the implementation of a different approach for the thrust term. The last section is devoted to the limitations and possible future lines of research.

6.1 As for the Mesh generation

A new strategy based on the quad-tree approach was proposed and tested. The proposed mesh, which was referred to as multi-quadtree mesh, showed to be effective when dealing with complex geometries such as those found in coastal environments. The pros of this meshing strategy are listed next:

- The resulting mesh allows to discretize very irregular and extensive domains. It is possible to vary from large cell sizes, of the order of a number of kilometers, to small cell sizes.
- The mesh algorithm is straightforward and easy to implement. Unlike the traditional quad-tree algorithm, the position of the cell centers is simply found with basic mathematical operations.
- Since the inputs for the mesh generation are "**.GRD**" files, it is possible to avoid manual interpolations when finding the terrain elevation at the locations required by the subdivision of the domain.
- The mesh algorithm can be applied in rectangular domains without requiring to implement many times the quad-tree approach, as other choices previously proposed.

6.2 As for the Solver

The HLLCS solver with a different formulation for the thrust term, i.e. the divergence form of the thrust term, was tested and validated. The resulting scheme showed to be well-balanced and accurate for the cases tested. The pros of the resulting scheme are listed next:

- The scheme implemented with the multi quad-tree mesh is an effective and practical tool to study floodings and their extent, in shallow environments.
- The divergence form of the thrust term works properly with the HLLCS solver. This formulation is easier to implement than its former version included in the solver (29).
- The computing time is reduced around 15 to 20 percent with this formulation of the source term.

6.3 Limitations and Future lines of research

- Although the divergence form of the thrust term is second order accurate (45), the resulting scheme when combined with the HLLCS solver is only of first order. Even though this is enough to obtain good solutions in many cases, it is recommended to improve at least at a second order of accuracy. However, developing a second order of this scheme when implemented on a quad-tree mesh is not a trivial task.
- Despite of the practicality of using "**.GRD**" files as inputs for the mesh algorithm, this restricts the possibilities to obtain an adaptive version of the meshing. Adaptive meshing is always an interesting choice, since it provides the possibility to reduce the number of cells when they are not required.
- The multi quad-tree mesh proposed does not seek to obtain separate cells for different seed points, i.e., a child cell may contain more than one seed point. Instead, the meshing only aims to achieve a desired resolution. This may or may not be a limitation, depending on the needs of the user.
- The scheme is not suitable for problems where the reflection effects are important, such as in the case of a train of regular waves previously presented.
- The wetting and drying technique does not consider partially dry cells. This is a potential area where future research might be required.
- The algorithm is not parallelized. Parallel computing is always necessary when dealing with large domains.

Appendices

Appendix A

Derivation of the Shallow Water Equations

We take the system of equations given by Eqs. 2.1 and 2.6 as a starting point for this chapter. After assuming a hydrostatic pressure distribution, they read:

$$\frac{\partial u_i}{\partial x_i} + \frac{\partial w}{z} = 0 \tag{A.1a}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) + \frac{\partial}{\partial z}(u_i w) = -g\frac{\partial \eta}{\partial x_i} - \frac{1}{\rho}\frac{\partial P_a}{\partial x_i} + \frac{\partial}{\partial x_j}(2\nu S_{ij}) + \frac{\partial}{\partial z}(2\nu S_{i3}) + F_i \tag{A.1b}$$

with i = (1,2).

A.1 Integration of the Continuity Equation

The depth integration of the continuity equation is an essential step in the derivation process. This allows to obtain a simple equation to compute the free surface. Integrating Eq. A.1a over the depth yields:

$$\int_b^\eta \left(\frac{\partial u_i}{\partial x_i} + \frac{\partial w}{z}\right) dz = 0 \tag{A.2}$$

Through the Leibniz's rule, it is possible to change the order of integration. By doing so and after expanding the index notation, Eq. A.1a reads:

$$\frac{\partial}{\partial x}\int_b^\eta u\,dz + u|_b\frac{\partial z_b}{\partial x} - u|_{z_\eta}\frac{\partial \eta}{x} +$$
$$+\frac{\partial}{\partial y}\int_b^\eta v\,dz + v|_b\frac{\partial z_b}{\partial x} - v|_{z_\eta}\frac{\partial \eta}{x} +$$

$$+w|_\eta - w|_b = 0 \tag{A.3}$$

Both the free surface and the bed are considered impermeable boundaries, i.e., there is no fluid flow across such boundaries. The kinematic boundary conditions express that at an impermeable boundary, the flow of fluid relative to the boundary must be tangential to it, this is, the normal component of the velocity is null.

Such kinematic BC, that relates the motion of the fluid relative to the boundary of the free interface to the fluid velocities at that interface, simply reads:

- **At the free surface:**

$$w|_\eta = \frac{\partial \eta}{\partial t} + u|_\eta \frac{\partial \eta}{\partial x} + v|_\eta \frac{\partial \eta}{\partial y} \tag{A.4a}$$

- **At the bed:**

$$w|_b = \frac{\partial z_b}{\partial t} + u|_b \frac{\partial z_b}{\partial x} + v|_b \frac{\partial z_b}{\partial y} \tag{A.4b}$$

Substituting Eqs. A.4a and A.4b into Eq. A.1a gives:

$$\left(\frac{\partial \eta}{\partial t} + \frac{\partial z_b}{\partial t}\right) + \frac{\partial}{\partial x}\int_b^\eta u\,dz + \frac{\partial}{\partial y}\int_b^\eta v\,dz = 0 \tag{A.5}$$

Considering that the depth averaged velocities, Eq. 2.7, are given by:

$$U = \frac{1}{h}\int_b^\eta u\,dz \quad \therefore \quad \int_b^\eta u\,dz = hU \tag{A.6a}$$

$$V = \frac{1}{h}\int_b^\eta v\,dz \quad \therefore \quad \int_b^\eta v\,dz = hV \tag{A.6b}$$

and that the water depth can be expressed in terms of both the free surface elevation and the bed level, $h = \eta - z_b$, the depth integrated continuity equation finally reads:

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hU) + \frac{\partial}{\partial y}(hV) = 0 \tag{A.7}$$

wich is consistent with Eq. 2.10a, presented in Chapter 2.

A.2 Integration of the Momentum Equation

By integrating Eq. A.3 over the depth, we get:

$$\int_b^\eta \left(\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) + \frac{\partial}{\partial z}(u_i w) \right) dz = \int_b^\eta \left(-g\frac{\partial \eta}{\partial x_i} - \frac{1}{\rho}\frac{\partial P_a}{\partial x_i} + \frac{\partial}{\partial x_j}(2\nu S_{ij}) + \frac{\partial}{\partial z}(2\nu S_{i3}) + F_i \right) dz \quad \text{(A.8)}$$

For the sake of clarity, the depth integration of the momentum equation is performed on a term by term basis. Starting with the left-hand side of Eq. A.8 and, again, for the Leibniz's rule we get:

$$\int_b^\eta \frac{\partial u_i}{\partial t} dz = \frac{\partial}{\partial t}\int_b^\eta u_i\, dz + u_i|_b \frac{\partial z_b}{\partial t} - u_i|_\eta \frac{\partial \eta}{\partial t} \quad \text{(A.9a)}$$

$$\int_b^\eta \frac{\partial}{\partial x_j}(u_i u_j)\, dz = \frac{\partial}{\partial x_j}\int_b^\eta u_i u_j\, dz + u_i|_b u_j|_b \frac{\partial z_b}{\partial x_j} - u_i|_\eta u_j|_\eta \frac{\partial \eta}{\partial x_j} \quad \text{(A.9b)}$$

$$\int_b^\eta (u_i w)\, dz = u_i|_\eta w|_\eta - u_i|_b w|_b \quad \text{(A.9c)}$$

Once again, we use the kinematic BCs (Eq. A.4). By inserting them into Eq. A.9c, we get:

$$\int_b^\eta (u_i w)\, dz = u_i|_\eta \frac{\partial \eta}{\partial t} + u_i|_\eta u_j|_\eta \frac{\partial \eta}{\partial x_j} - u_i|_b \frac{\partial z_b}{\partial t} - u_i|_b u_j|_b \frac{\partial z_b}{\partial x_j} \quad \text{(A.9d)}$$

After reducing similar terms, the left-hand side of Eq. A.8 reads:

$$\int_b^\eta \left(\frac{\partial u_i}{\partial t} + \frac{\partial}{\partial x_j}(u_i u_j) + \frac{\partial}{\partial z}(u_i w) \right) dz = \frac{\partial}{\partial t}\int_b^\eta u_i\, dz + \frac{\partial}{\partial x_j}\int_b^\eta u_i u_j\, dz = \frac{\partial}{\partial t}(HU_i) + \frac{\partial}{\partial x_j}(HU_iU_j) \quad \text{(A.10)}$$

with

$$\frac{\partial}{\partial x_j}\int_b^\eta (u_i u_j)\, dz = \frac{\partial}{\partial x_j} H\,(U_iU_j + \cancel{D_{ij}}) \quad \text{(A.11)}$$

The expression D_{ij} in Eq. A.11 arises from depth averaging the non-linear convective terms. Such term is responsible for the dispersion-diffusion processes and it is usually neglected.

Additionally, the right-hand side of Eq. A.8 reads:

$$\int_b^\eta \left(-g\frac{\partial \eta}{\partial x_i} - \frac{1}{\rho}\cancel{\frac{\partial P_a}{\partial x_i}} + \frac{\partial}{\partial x_j}(2\nu S_{ij}) + \frac{\partial}{\partial z}(2\nu S_{i3}) + \cancel{F_i} \right) dz \tag{A.12}$$

where some terms are cancelled out. A short explanation about these terms is provided next.

- The atmospheric pressure term, the second expression, is usually considered more or less constant. Unless in case of an extreme event such as a hurricane, this assumption is not valid.
- The last expression represents any external forces, for instance the Coriolis force. Any of the test cases presented in this thesis includes such type of forces, so the term is no longer useful.

By integrating the remaining terms in Eq. A.12, we get:

$$\begin{aligned} \int_b^\eta \left(-g\frac{\partial \eta}{\partial x_i} \right) dz &= -hg\frac{\partial \eta}{\partial x_i} \\ &= -hg\frac{\partial (h+z_b)}{\partial x_i} \\ &= -\frac{1}{2}g\frac{\partial h^2}{\partial x_i} - gh\frac{\partial z_b}{\partial x_i} \end{aligned} \tag{A.13a}$$

$$\int_b^\eta \frac{\partial}{\partial x_j}(2\nu S_{ij})\, dz = \frac{\partial}{\partial x_j}\int_b^\eta (2\nu S_{ij})\, dz + (2\nu S_{ij})|_b \frac{\partial z_b}{\partial x_j} - (2\nu S_{ij})|_\eta \frac{\partial \eta}{\partial x_j} \tag{A.13b}$$

$$\int_b^\eta \frac{\partial}{\partial z}(2\nu S_{i3})\, dz = (2\nu S_{i3})\,|_\eta - (2\nu S_{i3})\,|_b \tag{A.13c}$$

with the shear stress at the bed and at the free surface defined as:

$$\tau_{\eta_i} = (2\nu S_{i3})|_\eta - (2\nu S_{ij})|_\eta \frac{\partial \eta}{\partial x_j} \tag{A.13d}$$

$$\tau_{b_i} = (2\nu S_{i3})|_b - (2\nu S_{ij})|_b \frac{\partial z_b}{\partial x_j} \tag{A.13e}$$

By combining Eqs. A.13d and A.13e with Equations (A.13a) to (A.13c), and after cancelling out similar terms, Eq. A.12 reads:

$$\int_b^\eta \left(-g\frac{\partial \eta}{\partial x_i} + \frac{\partial}{\partial x_j}(2\nu S_{ij}) + \frac{\partial}{\partial z}(2\nu S_{i3}) \right) dz = -\frac{1}{2}g\frac{\partial h^2}{\partial x_i} - gh\frac{\partial z_b}{\partial x_i} + \\ + \frac{\partial}{\partial x_j}\int_b^\eta \cancel{(2\nu S_{ij})}\, dz + \frac{1}{\rho}(\tau_{\eta_i} - \tau_{b_i}) \quad \text{(A.14)}$$

where the third term is cancelled out. This term accounts also for dispersion-diffusion processes, i.e., the longitudinal and horizontal exchange of momentum. It is usually neglected since it is assumed that the vertical momentum exchange is dominant over the horizontal one.

Thus rewriting Eq. A.8, once the depth integration is performed, yields:

$$\frac{\partial}{\partial t}(HU_i) + \frac{\partial}{\partial x_j}(HU_iU_j) = -\frac{1}{2}g\frac{\partial h^2}{\partial x_i} - gh\frac{\partial z_b}{\partial x_i} + \frac{1}{\rho}(\tau_{\eta_i} - \tau_{b_i}) \quad \text{(A.15)}$$

The resulting non-linear shallow water equations (NSWE) are given by Eq. A.7 and Eq. A.15. After expanding index notation in Eq. A.15 they read:

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hU) + \frac{\partial}{\partial y}(hV) = 0 \quad \text{(A.16a)}$$

$$\frac{\partial}{\partial t}(hU) + \frac{\partial}{\partial x}\left(hU^2\right) + \frac{\partial}{\partial y}(hUV) = -\frac{1}{2}g\frac{\partial h^2}{\partial x} - gh\frac{\partial z}{\partial x} + \frac{1}{\rho}(\tau_{\eta_x} - \tau_{b_x}) \quad \text{(A.16b)}$$

$$\frac{\partial}{\partial t}(hV) + \frac{\partial}{\partial y}\left(hV^2\right) + \frac{\partial}{\partial x}(hVU) = -\frac{1}{2}g\frac{\partial h^2}{\partial y} - gh\frac{\partial z}{\partial y} + \frac{1}{\rho}(\tau_{\eta_y} - \tau_{b_y}) \quad \text{(A.16c)}$$

which is consistent with Eq. 2.10, presented in Chapter 2.

Appendix B

Laboratory Tests

The description of the physical model developed and used for the validation of the numerical model is presented in this chapter.

B.1 Experimental Setup

The experiments were carried out at the National Autonomous University of Mexico (UNAM), in the "Laboratorio de Costas y Puertos" (Coastal and Port Engineering Laboratory). This laboratory is equipped with a wave canal, which is 37 meters long, 80 centimeters wide and 120 centimeters high; supported by a tubular structure (PTR steel) at a height of 80 centimeters above the floor level. The wave generation equipment, piston type, is composed of a steel plate that moves back and forth, which is supported by a horizontal track and computer controlled.
The ramp of constant slope (1:5) was placed inside the wave canal and the run up for different wave conditions and roughness was investigated. The ramp was composed of a metal structure, which, by means of bolts, holds the acrylic sheets on which the waves act. In turn, these acrylic sheets were later covered with 150-grit wet sandpaper and extra-thick grit sandpaper, giving rise to three roughness scenarios (see Figs. B.1 and B.2).

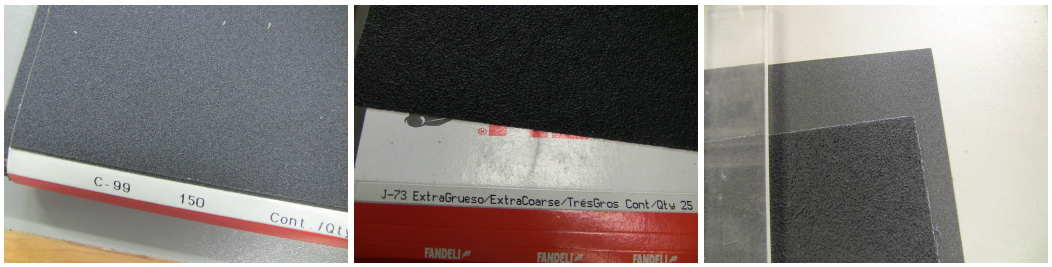
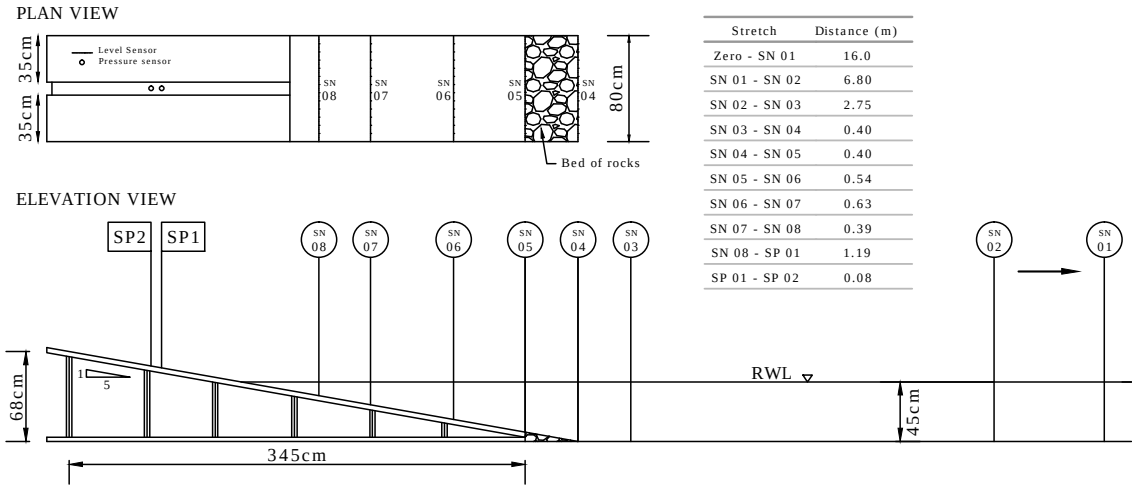


Figure B.1: Acrylic sheets and sand paper used for the experiment.

Figure B.2: Ramp metal structure covered with acrylic sheets (right) and with sandpaper (left).

As for the data logging, 4 level sensors were placed along the canal (SN01 - SN04), 4 level sensors (SN05 - SN08) and 2 pressure sensors (SP01, SP02) along the ramp. The specific location of such sensors and a layout of the experiment are shown in Fig. B.3. Additionally, a bedrock was placed at the foot of the ramp in order to have a gradual transition from the canal bottom to the ramp.



Stretch	Distance (m)
Zero - SN 01	16.0
SN 01 - SN 02	6.80
SN 02 - SN 03	2.75
SN 03 - SN 04	0.40
SN 04 - SN 05	0.40
SN 05 - SN 06	0.54
SN 06 - SN 07	0.63
SN 07 - SN 08	0.39
SN 08 - SP 01	1.19
SP 01 - SP 02	0.08

Figure B.3: Layout of the experiment.

Besides the level sensors (Fig. B.4) and the pressure sensors (Fig. B.5), it was used a Fastec Imaging high speed camera (Fig. B.6), Hispec 2g mono model, which was set to record 150 frames per wave. While the sensors allow to get a characterization of the wave patterns, this high speed camera allows to obtain a photo record of the moment when the wave hits the ramp and the highest

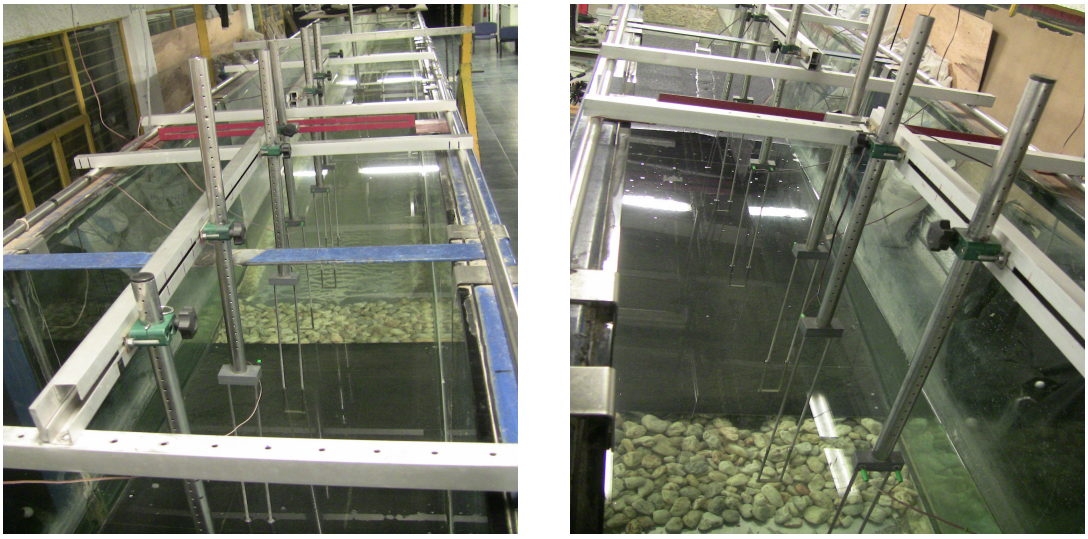


Figure B.4: Level sensors.

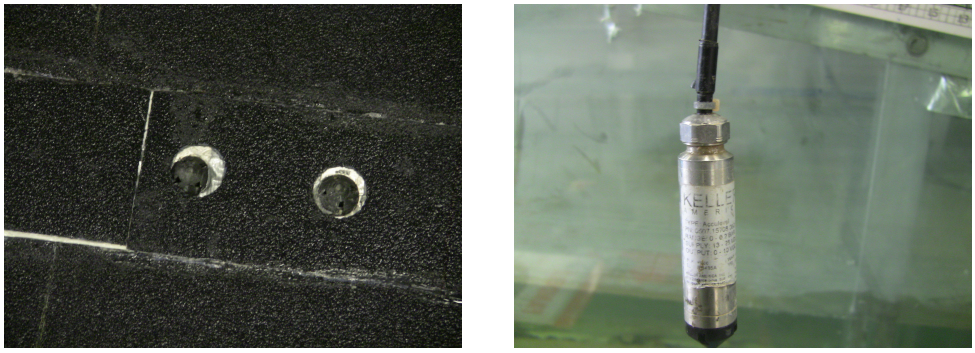


Figure B.5: Pressure sensors.

run up achieved by the wave, which was the major objective of the experiment. A scale of lengths was placed in the horizontal local plane of the ramp, in order to associate each value of the scale with a value of run up measured to the mean depth, 45 cm.

Two different wave conditions were tested, a regular wave train (8 cm height and 3.9 s period) and a solitary wave (8.5 cm height), both selected by trial and error so that overtopping did not ever occur. Therefore, three different roughness conditions and two wave profiles give rise to a total of 30 tests, since 5 tests were carried out for each combination possible. Each regular wave test had a duration of 5 minutes, with the photo record starting at minute 3, considering that the flow is fully developed after such time. For the solitary wave, each test had a duration of 1 minute.

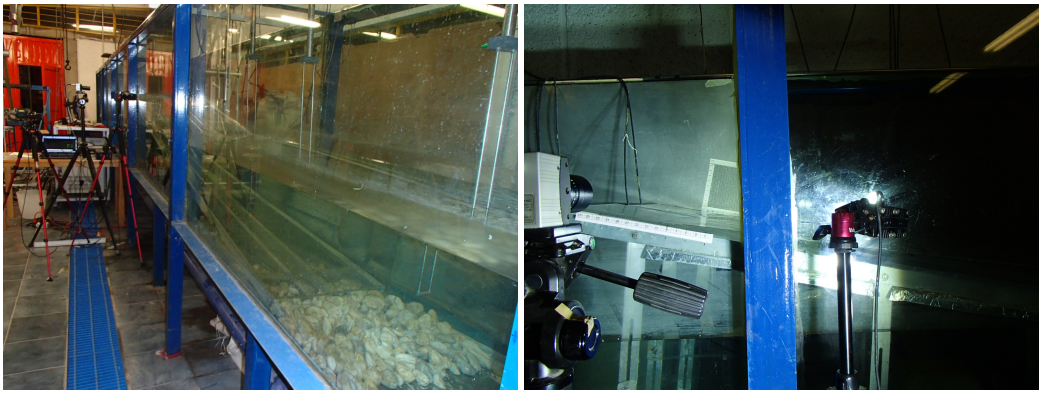


Figure B.6: Fastec Imaging high speed camera.

B.2 Experimental Results

The results of the tests can be divided into two categories: the photographic record for the run up measurement and the level and pressure sensors records. For each regular wave test (15 trials), a set of 4100 photographs were obtained, which captured the maximum run up of approximately 27 waves per test, and for each solitary wave test (15 trials) the record consisted of 150 photographs. When it regards to the sensors, they were set to record at a speed of 100 values per second. As a result, 30000 values were obtained for each regular wave test and 6000 for each solitary wave test. For the regular wave tests, the data processing included a mean level correction followed by a positive-going zero crossing analysis to define the period and height of the waves. For the solitary wave, data processing was just to find the zero crossings to calculate the wave height. Results for both tests are presented in Tables B.1 to B.4.
For the image processing, it was required to develop a graphical interface in matlab (Fig. B.8), by means of which the images were loaded and the run up was calculated as a matrix subtraction operation. The first step is to load the image in binary format, in which the measurements are to be performed, and the grayscale image as obtained by the camera. Step 2 consists of marking two guide points along the lines of reference on the ramp. Depending on where the maximum run up is displayed, the guide points are obtained from a perpendicular between the reference lines, which must also pass through the maximum. In addition to the two guide points, it should be noted that one in which the maximum occurs, so that a total of three points are considered.

The third and last step is to the carry out the measurement, by clicking on the "medir" button. The value of the bottom line of reference has to be introduced in the blank space, since the algorithm measures the relative distance between the two reference lines. Finally the total distance, referred to the zero value in the scale, is displayed in the blank space designated for such purpose. An example of the measurement process is shown in Fig. (B.8).

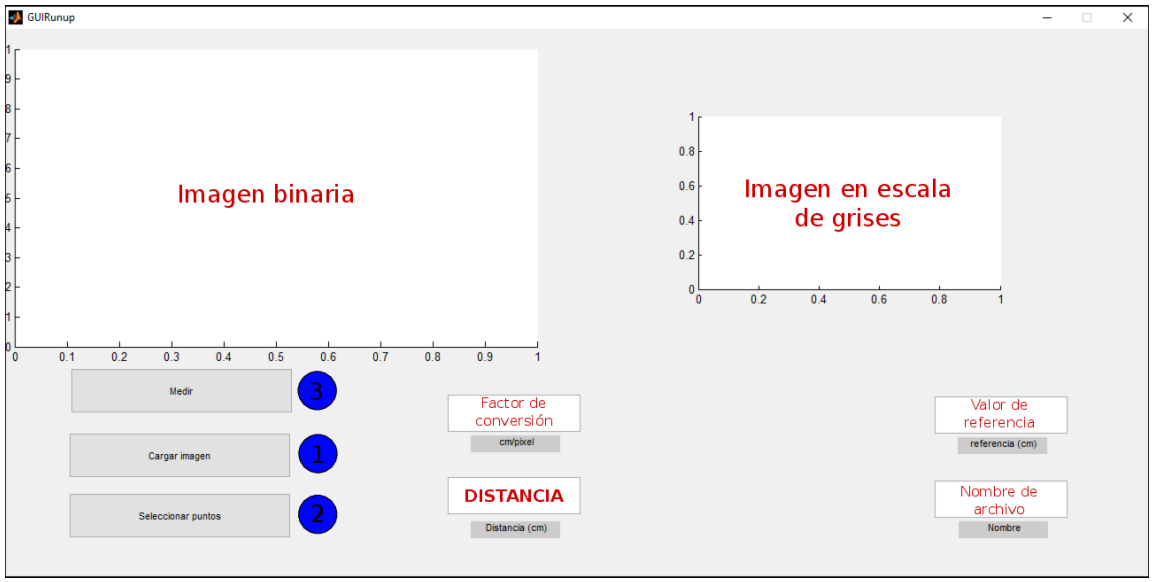


Figure B.7: Graphical interface for image processing in matlab.

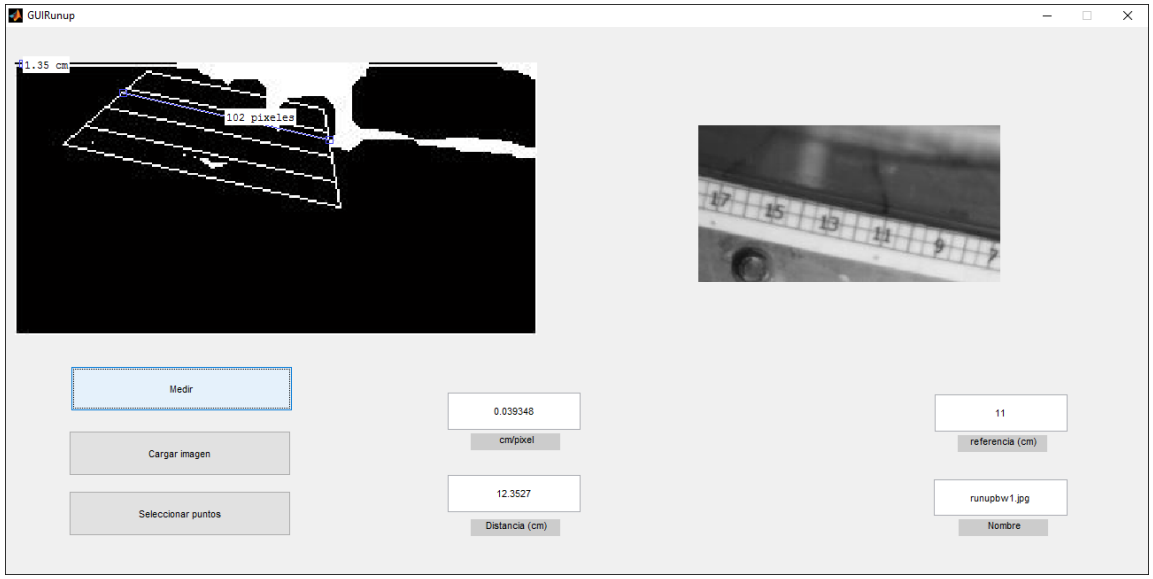


Figure B.8: Images processing.

Table B.1: Sensor 1 record - Acrylic sheet - Regular wave train.

Test	1	2	3	4	5	Mean	STD
Height (cm)	3.8999	3.9002	3.9003	3.9004	3.8995	3.9000	0.0003
Period (s)	10.748	10.740	10.772	10.770	10.772	10.7603	0.0127

Table B.2: Sensor 1 record - 150 grit sandpaper - Regular wave train.

Test	1	2	3	4	5	Mean	STD
Height (cm)	3.8998	3.8997	3.8996	3.8999	3.8999	3.8998	0.0001
Period (s)	10.715	10.642	10.725	10.670	10.712	10.6929	0.0288

Table B.3: Sensor 1 record - extra-thick grit sandpaper - Regular wave train.

Test	1	2	3	4	5	Mean	STD
Height (cm)	3.9003	3.8999	3.9000	3.9002	3.9001	3.9001	0.0001
Period (s)	10.861	10.913	10.879	10.907	10.931	10.8981	0.0252

Table B.4: Solitary wave height - Average of sensors 1, 2 and 3 (units in cm).

	Test	1	2	3	4	5	Mean	STD
Material	**Acrylic**	8.508	8.428	8.463	8.368	8.447	8.443	0.0457
	150 grit	8.614	8.593	8.604	8.601	8.583	8.599	0.0104
	Extra-thick grit	8.531	8.472	8.518	8.550	8.583	8.531	0.0367

Table B.5: Mean value of height and run up for all cases.

	Regular wave		Solitary wave	
Material	**H (cm)**	**Run up (cm)**	**H (cm)**	**Run up (cm)**
Acrylic	24.19	10.2	24.85	8.4
150 grit	22.48	10.2	24.71	8.6
Extra-thick grit	21.18	10.4	23.37	8.5

Bibliography

[1] Amiram Harten, Peter D . Lax, B. V. L. (1983). On upstream differencing and godunov-type schemes for hyperbolic conservation laws. *Society for Industrial and Applied Mathematics Stable*, 25:35–61. 30, 31

[2] Bates, P. D., Horritt, M. S., and Fewtrell, T. J. (2010). A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *Journal of Hydrology*, 387(1-2):33–45. 1

[3] Benkhaldoun, F., Elmahi, I., Seaı, M., et al. (2007). Well-balanced finite volume schemes for pollutant transport by shallow water equations on unstructured meshes. *Journal of computational physics*, 226(1):180–203. 1

[4] Bermudez, A. and Vazquez, M. E. (1994). Upwind methods for hyperbolic conservation laws with source terms. *Computers and Fluids*, 23(8):1049–1071. 43

[5] Borthwick, A., Cruz Leon, S., and Józsa, J. (2001). The shallow flow equations solved on adaptive quadtree grids. *International Journal for Numerical Methods in Fluids*, 37(6):691–719. 19

[6] Bradford, S. F. and Sanders, B. F. (2002). Finite-volume model for shallow-water flooding of arbitrary topography. *Journal of Hydraulic Engineering*, 128(3):289–298. 1

[7] Casulli, V. and Cheng, R. T. (1992). Semi-implicit finite difference methods for three-dimensional shallow water flow. *International Journal for Numerical Methods in Fluids*, 15:629–648. 2

[8] Chávez, V., Mendoza, E., Ramírez, E., and Silva, R. (2017). Impact of inlet management on the resilience of a coastal lagoon: La mancha, veracruz, mexico. *Journal of Coastal Research*, 77(sp1):51–61. 57

[9] Chertock, A., Kurganov, A., and Petrova, G. (2006). Finite-volume-particle methods for models of transport of pollutant in shallow water. *Journal of Scientific Computing*, 27(1-3):189–199. 1

[10] CICESE (2019). Predicción de mareas. http://redmar.cicese.mx/meteoro/graph/prediccion.php. Accessed: 2019-11-20. 58

[11] Clay Mathematics Institute (no year). Navier–Stokes Equation | Clay Mathematics Institute. 7

[12] Darbani, M., Ouahsine, A., Villon, P., Naceur, H., and Smaoui, H. (2011a). Meshless method for shallow water equations with free surface flow. *Applied Mathematics and Computation*, 217(11):5113–5124. 2

[13] Darbani, M., Ouahsine, A., Villon, P., Naceur, H., and Smaoui, H. (2011b). Meshless method for shallow water equations with free surface flow. *Applied Mathematics and Computation*, 217:5113–5124. 2

[14] Delestre, O., Lucas, C., Ksinant, P.-A., Darboux, F., Laguerre, C., Vo, T.-N.-T., James, F., and Cordier, S. (2013). SWASHES: a compilation of shallow water analytic solutions for hydraulic and environmental studies. *INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS Int. J. Numer. Meth. Fluids*, 72:269–300. 47

[15] Díaz, M. C., Fernández-Nieto, E. D., and Ferreiro, A. (2008). Sediment transport models in shallow water equations and numerical approach by high order finite volume methods. *Computers & Fluids*, 37(3):299–316. 1

[16] Donea, J., Huerta, A., Ponthot, J.-P., and Rodriguez-Ferran, A. (2004). Arbitrary lagrangian eulerian methods. In *Encyclopedia of Computational Mechanics*, pages Chapter 14, Vol. 1. E. Stein, R. de Borst and T.J.R. Hughes, Wiley & sons. 2

[17] García-Navarro, P., Brufau, P., Burguete, J., and Murillo, J. (2008). The shallow water equations: An example of hyperbolic system. *Monografías de la Real Academia de Ciencias de Zaragoza*, 31:89–119. 15

[18] Godlewski, E. and Raviart, P.-A. (1996). *Numerical Approximation of Hyperbolic Systems of Conservation Laws*, volume 118. Springer-Verlag New York. 15

[19] Godunov, S. K. (1959). A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Mat. Sb. (N.S.)*, 47(89):271–306. 29, 30

[20] Grinsted, A., Ditlevsen, P., and Christensen, J. H. (2019). Normalized us hurricane damage estimates using area of total destruction, 1900- 2018. *Proceedings of the National Academy of Sciences*, 116(48):23942–23946. 3

[21] Grose, W. L. and Hoskins, B. J. (1979). On the influence of orography on large-scale atmospheric flow. *Journal of the Atmospheric Sciences*, 36(2):223–234. 1

[22] Heniche, M., Secretan, Y., Boudreau, P., and Leclerc, M. (2000). A two-dimensional finite element drying-wetting shallow water model for rivers and estuaries. *Advances in water resources*, 23(4):359–372. 1

[23] Johnson, R. W. (2016). *Handbook of fluid dynamics*. Crc Press. 7

[24] Kranenburg, C. (1992). Wind-driven chaotic advection in a shallow model lake. *Journal of hydraulic Research*, 30(1):29–46. viii, 52, 55

[25] Li, S. and Duffy, C. J. (2011). Fully coupled approach to modeling shallow water flow, sediment transport, and bed evolution in rivers. *Water Resources Research*, 47(3). 1

[26] Li, S. and Duffy, C. J. (2012). Fully-coupled modeling of shallow water flow and pollutant transport on unstructured grids. *Procedia Environmental Sciences*, 13:2098–2121. 1

[27] Łukaszewicz, G. and Kalita, P. (2016). *Navier–Stokes Equations: An Introduction with Applications*. Springer. 7

[28] MacDonald, I., Baines, M. J., Nichols, N. K., and Samuels, P. G. (1997). Analytic Benchmark Solutions for Open-Channel Flows. *Journal of Hydraulic Engineering*, 123(11):1041–1045. 47, 48

[29] Murillo, J. and García-Navarro, P. (2012). Augmented versions of the HLL and HLLC Riemann solvers including source terms in one and two dimensions for shallow flow applications. *Journal of Computational Physics*, 231(20):6861–6906. vii, xix, 5, 15, 29, 31, 33, 35, 47, 66

[30] OECD (2016). *Financial Management of Flood Risk*. OECD Publishing. 3

[31] Peiró, J. and Sherwin, S. (2005). Finite difference, finite element and finite volume methods for partial differential equations. In Yip, S., editor, *Handbook of Materials Modeling*, pages 2415–2446. Springer Netherlands. 2

[32] Posada-Vanegas, G. (2007). *Modelo numérico hidrodinámico tridimensional para la predicción de la evolución de una descarga de una substancia conservativa de un emisor submarino*. PhD thesis, PhD thesis, 126p.(in spanish). 19

[33] Ritchie, H. and Roser, M. (2019). Natural disasters. *Our World in Data*. https://ourworldindata.org/natural-disasters. 4

[34] Rivera, J., Chávez, V., Silva, R., and Mendoza, E. (2019). Modelling the effects of the artificial opening of an inlet. *Journal of Coastal Research*, pages 128–135. 57

[35] Roe, P. (1981). Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 43(2):357–372. 30

[36] Rogers, B., Fujihara, M., and Borthwick, A. G. L. (2001). Adaptive Q-tree Godunov-type scheme for shallow water equations. *International Journal for Numerical Methods in Fluids*, 35:247–280. 22, 52, 57

[37] Salazar Carrillo, E. E. (2014). *Implementación y Validación de un Modelo Numérico de Circulación Oceánica para el Golfo de México*. PhD thesis, Universidad Nacional Autónoma de México. 29

[38] Samet, H. (1982). Neighbor Finding Techniques for Images Represented by Quadtrees. *Computer graphics and image processing*, 18:37–57. 19

[39] Sanders, B. F., Schubert, J. E., and Gallegos, H. A. (2008). Integral formulation of shallow-water equations with anisotropic porosity for urban flood modeling. *Journal of hydrology*, 362(1-2):19–38. 38

[40] Schubert, J. E. and Sanders, B. F. (2012). Building treatments for urban flood inundation models and implications for predictive skill and modeling efficiency. *Advances in Water Resources*, 41:49–64. 38

[41] Smy, T., Walkey, D., and Dew, S. (2001). Transient 3d heat flow analysis for integrated circuit devices using the transmission line matrix method on a quad tree mesh. *Solid-State Electronics*, 45(7):1137–1148. 19

[42] Tan, W.-Y. (1992). *Shallow water hydrodynamics: Mathematical theory and numerical solution for a two-dimensional system of shallow-water equations*, volume 55. Elsevier. 1

[43] Tennekes, H. and Lumley, J. L. (1972). *A first course in turbulence*. MIT Press, Cambridge, Mass. 2

[44] Toro, E. F., Spruce, M., and Speares, W. (1994). Restoration of the contact surface in the hll-riemann solver. *Shock waves*, 4(1):25–34. 30, 31

[45] Valiani, A. and Begnudelli, L. (2006). Divergence form for bed slope source term in shallow water equations. *Journal of Hydraulic Engineering*, 132(7):652–665. xix, 5, 38, 39, 66

[46] Walko, R. L. and Avissar, R. (2008). The ocean–land–atmosphere model (olam). part i: Shallow-water tests. *Monthly Weather Review*, 136(11):4033–4044. 1

[47] Wang, Z. (1998). A quadtree-based adaptive cartesian/quad grid flow solver for navier-stokes equations. *Computers & Fluids*, 27(4):529–549. 19

[48] Wilks, G. (2007). The navier-stokes equations: A classification of flows and exact solutions. by drazin philip & riley norman. cambridge university press, 2006. 206 pp. isbn 0521 681626.£ 27.99 (paperback). *Journal of Fluid Mechanics*, 577:506–507. 7

[49] Yiu, K., Greaves, D., Cruz, S., Saalehi, A., and Borthwick, A. (1996a). Quadtree grid generation: information handling, boundary fitting and cfd applications. *Computers & Fluids*, 25(8):759–769. 19

[50] Yiu, K. F. C., Greaves, D. M., Cruz, S., Saalehi, A., and Borthwick, A. G. L. (1996b). Quadtree grid generation: Information handling, boundary fitting and CFD applications. *Computers and Fluids*, 25(8):759–769. 22