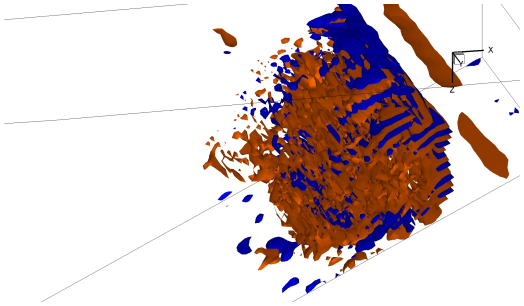
UNIVERSIDADE TÉCNICA DE LISBOA
INSTITUTO SUPERIOR TÉCNICO



Numerical Modeling of Non-Hydrostatic Processes in Estuarine and Coastal Regions

Hernâni António Cidade Moura Theias
Licenciado em Engenharia Civil

Dissertação submetida para obtenção do grau
de
Mestre em Hidráulica e Recursos Hídricos

Orientador :
Professor António Alexandre Trigo Teixeira
Co-Orientador:
Professor Paulo Miguel Chambel Filipe Lopes Teles Leitão

Primeira versão submetida ao júri
Setembro 2005

Contents

List of Figures

List of Tables

List of Symbols

Roman Letters

Symbol	Description	Units
b_i	body force component in the i direction per unit of mass	$\mathrm{m\,s^{-2}}$
C_D	drag coefficient	(-)
f_i	Coriolis force per mass unit in the i direction	$\mathrm{m\,s^{-2}}$
F_s	flux in s direction	$\mathrm{m\,s}$
Fr	Froude number	(-)
g	earth gravity	$\mathrm{m\,s^{-2}}$
h	water depth (measured from average water level)	m
$\mathbf{j}$	index vector (j, k, l)	(-)
k	turbulent kinetic energy per unit of mass	$\mathrm{m^2\,s^{-2}}$
l_m	turbulent mixing length	m
p	pressure	$\mathrm{N\,m^{-2}}$
p_a	atmospheric pressure	$\mathrm{N\,m^{-2}}$
q	normalized pressure correction	$\mathrm{m^2\,s^{-2}}$
Re	Reynolds number	(-)
Ri	Richardson number	(-)
S_{ij}	rate-of-strain tensor	$\mathrm{m\,s^{-2}}$
S_i	surface of face with normal in the i direction	$\mathrm{m^2}$
t	time	s
u	velocity in the x direction	$\mathrm{m\,s^{-1}}$
$\bar{u}$	time averaged velocity	$\mathrm{m\,s^{-1}}$
u'	random component of velcoity	$\mathrm{m\,s^{-1}}$
u_a	wind velocity	$\mathrm{m\,s^{-1}}$
u_i	velocity in the i direction	$\mathrm{m\,s^{-1}}$
v	velocity in the y direction	$\mathrm{m\,s^{-1}}$
w	velocity in the y direction	$\mathrm{m\,s^{-1}}$
x_i	i direction coordinate	m
z_b	distance from bottom cell center to bottom wall	m

Greek Letters

Symbol	Description	Units
$\boldsymbol{\delta}(s)$	displacement index vector in the s direction $\equiv (\delta_{1s}, \delta_{2s}, \delta_{3s})$	m
δ_{ij}	the Kronecker symbol	(-)
δx_s	grid spacing in the s direction	m
ε_0	roughness height	m
η	free-surface elevation	m
θ	semi-implicit coefficient	(-)
κ	von Kármán constant	(-)
ν	kinematic viscosity (either molecular or total)	$\mathrm{m^2\,s^{-1}}$
ν_t	turbulent (or eddy) viscosity	$\mathrm{m^2\,s^{-1}}$
ρ	fluid density	$\mathrm{kg\,m^{-3}}$
τ_{ij}	stress tensor	$\mathrm{N\,m^{-2}}$
τ_a	wind shear stress	$\mathrm{N\,m^{-2}}$
τ_b	bottom shear stress	$\mathrm{N\,m^{-2}}$
φ	latitude	$(^o)$
ω	earth angular rotation velocity	$\mathrm{rad\,s^{-1}}$
Ω	CV volume for mass properties	$\mathrm{m^3}$
Ω_i	CV volume in i velocity grid	$\mathrm{m^3}$

Ao meu pai.

O óptimo é inimigo do bom.

Agradecimentos

Em primeiro lugar quero agradecer ao Professor António Trigo Teixeira todo o apoio que me deu ao longo destes anos de uma tese que parecia não acabar nunca. Deu-me sempre a liberdade e o tempo que eu precisava para desenvolver os meus conhecimentos e as minhas capacidades. Aquilo que alguns podem considerar tempo perdido, em leituras, estudos diversos, tentativas abortadas, impasses de todo o género no estudo, não foram senão os exercício que me permitiram levar a cabo com êxito este projecto. Esta tese, desenvolvida e escrita em menos de seis meses, é a maturação de quatro anos de reflexões, estudos e trabalhos. Recordo aqui com a mais alta estima as palavras do Professor Quintela, palavaras que ele terá dito um dia a um colega de curso: "A investigação? 90% é músculo." Parece que alguns músculos levam o seu tempo a desenvolver-se. Agradeço com a maior das sinceridades ao Professor Trigo Teixeira por ter-me deixado a margem de manobra que precisava para investigar e de resto o apoio incondicional que me deu ao longo de todo este tempo.

Não posso deixar de agradecer ao Paulo, que me acompanhou estes últimos meses, seguindo de muito perto o meu trabalho com o entusiasmo e o rigor.

Devo também um grande agradecimento ao Ramiro, que me recebeu na Maretec dando-me um enquadramento logísitco e científico que eu nem sonhava. Tenho que felicitá-lo pela dinâmica de grupo que consegue, apesar de todas as dificuldades que surgem.

A todos aqueles que participaram de uma forma ou outra neste trabalho devo ainda uma eternidade de gratidão - porque eterno era o fardo de que me livraram. Ao Professor Aires pelos seus comentários sempre pertinentes, e ao Professor Luis Eça pela 'receita' do algoritmo do gradiente conjugado. Ao Professor Matos Silva, o meu guia espiritual, que me levou para o mundo da modelação numérica. Quanto aos meus caros colegas da Marateca, estive a fazer as contas, e se cito todos aqueles que me ajudaram na tese, ficam poucos de fora, e não quero ser ofensivo, por isso prefiro agradecer a todos e ninguém em particular. Aqueles que mais me ajudaram sabem quem são. Não me esqueço deles. Nem dos outros.

Ao meu amigo Luis Mendes, cabe-lhe aqui um lugar especial, já que fomos colegas de curso e colegas de mestrado, mas para além de tudo, amigos, sem dúvida de alguma maneira irmanados nesta luta de tese que sofremos de maneira semelhante.

Gostava de poder agradecer os meus amigos e à família por o seu contributo para este trabalho, mas, para ser completamente sincero, actuaram mais como fontes permanentes de entropia do que como estabilizadores. Mas ao que parece não fizeram senão cumprir o seu papel. Por isso, deixo-lhes aqui um abraço sem resentimentos.

Em particular, que fique escrito: *te agradezco por la fuerza que me diste todo este tiempo. Sí, tú también escribiste este montón de ecuaciones incomprensibles.*

Esta tese é dedicada àquele que me meteu nisto: o meu pai.

Acknowledgments

In the first place I would like to thank sincerely my supervisor, Professor António Trigo Teixeira, for the support he gave me all along the years of this seemingly everlasting thesis. I always felt free to take the time I needed to learn and apply. What some may feel like time waste, in lectures and readings, many times through dead end paths, was for me the real accomplishment of this thesis. It leaded to the present work, hopefully successful and mature. "Research? 90% is muscle.", quoting here with due esteem, Professor António Quintela, former full professor at IST, makes me think that some muscles might need time to grow, as it was the case for me. So I must sincerely thank Professor Trigo Teixeira for having let me do "research" – which does not mean that you always "find" – and for his unconditional support at any phase of this thesis.

Dr. Paulo Leitão's constant presence and support has been a crucial for the realization of this work. He followed closely, with care and enthusiasm its development.

I also whish to express gratitude to Professor Ramiro Neves, that received me at Maretec, providing me with the work environment that I needed. I must thank him and also congratulate him for his success in creating a truly motivated research group.

For those who, in a way or another, toke part in this work, I owe them an eternity of gratitude: to Profesor Aires for its pertinent commentaries; to Professor Luís Eça for his recipe of the Conjugate Gradient; to Professor Matos Silva for following my trajectory, giving me the first impulse towards numerical modeling science; to all the Maretec staff, that is too numerous to be cited here.

Special thanks to Luís, for being the supportive, sympathetic and amusing old friend he has always been, within the darkest hours of IST corridors.

Thanks also is due to family and other friends; to be honest, I can only recall them as a perpetuate source of entropy, which is nothing but their usual role and purpose. For which I thank them.

There was a particular force driving the last part of the thesis, an eastern rainstorm running-off all the way down to the Tagus estuary, and that probably motivated me more than anything. *Gracias Crisitina.*

Finally this thesis is dedicated to the original cause of... everything: my father.

Resumo

Em modelos oceânicos é usual a hipótese de distribuição hidrostática da pressão. No entanto, em escoamentos com acelerações verticais importantes esta hipótese perde validade, o que acontece, em regra geral, a escalas inferiores à escala geostrófica - quando, por exemplo, o modelo oceânico é utlizado para simular estuários e zonas costeiras. O estudo que segue desenvolve um modelo que é uma correcção de un programa existente, MOHID, que parte da hipótese hydrostática. Seguindo a abordagem do programa pré-existente, é utilizado o método dos volumes finitos para discretizar as equações de correcção da pressão que são depois introduzidas no modelo. O modelo é validado mostrando convergência para os casos estudados e apresentando soluções realistas numa série de testes onde, em princípio, um modelo hidrostático deve falhar. A introdução de fluxos difusivos parece causar problemas de estabilidade. O modelo foi ainda testado a duas e três dimensões com sucesso.

palavras-chave : modelação numérica, modelo oceânico, não-hidrostático, tridimensional, ondas internas, Kelvin-Helmoltz

Abstract

Ocean models that assume hydrostatic pressure distribution may fail when vertical accelerations become important, leading to a velocity field that is no longer bi-dimensional as the hydrostatic hypothesis implies. This is more common at scales smaller than the geostrophic scales, for instance, when ocean models are used to simulate estuaries and coastal regions. The following study develops a model that is a correction to an existing program called MOHID that makes the hydrostatic assumption. The program uses a finite volume approach which is followed in this model's application. Equations for the pressure correction are discretized and introduced in the pre-existing model. The model is validated by showing convergence and providing realistic answers to a series of problems (either quantitatively and qualitatively) for which it is known that the hydrostatic model is wrong. The only flaw of the model seems to be the difficulty in keeping under stability conditions when introducing turbulence effects through diffusive fluxes. The model is tested in two and three dimensions seeming to behave as expected.

keywords : numerical modeling, ocean model, non-hydrostatic, tridimensional, internal waves, Kelvin-Helmoholtz

Introduction

In the course of ocean modeling studies, an usual practice consists in considering the pressure distribution as hydrostatic. This approximation yields the Hydrostatic Primitive Equations (HPE) whose validity domain is, in principle, limited to large scales The HPE describe very accurately the global circulation of ocean ($L \sim 100\,\mathrm{km}$) and the geostrophic eddies and rings associated with its instability ($L \sim 10 - 100\,\mathrm{km}$) [5]. However at smaller scales, the flow is driven mainly by convection, gravity forces and buoyancy ($L < 10\,\mathrm{km}$) and somewhere between the geostrophic and convective scales ($L \sim 1 - 10\,\mathrm{km}$) the HPE may break down. We will show in a later chapter more precisely when this happens.

Figure 1: An example that puts into evidence non-hydrostatic processes in the atmosphere

Non-hydrostatic effects are readily seen in the atmosphere thanks to a natural tracer: water vapour. Figure 1 shows a stratus cloud in the atmosphere whose particular shape results from a well-known non-hydrostatic phenomenon: the Kelvin-Helmoltz instability. Analogous occurrences are found in the water, but picturing them is another issue. Differences in salinity or temperature are cause for such events under water that are called internal waves.

In a highly stratified region, the surface separating two layers of different densities - the pycnocline - can undergo wave motion. This motion, which

does not affect significantly the surface elevation is an example of an internal wave. Such waves can attain much larger amplitude than surface waves (up to 50 meters). Internal waves can be easily observed in the following satellite photograph (Figure 2).

Figure 2: Satellite photography of an internal wave occured off the coast of Somalia

The previous picture shows internal waves refracted by sea bed. The effect of the wave is visible by streaks of smooth water produced by the convergences above the wave troughs. The correct description of internal wave motion requires an accurate non-hydrostatic model because surface and pycnocline deformation are not coupled (internal waves can even break while there is no water surface deformation).

Flows over rapidly varying slopes and wind waves in shallow waters are other examples where the hydrostatic approximation does not apply. In fact, it only applies as long as vertical accelerations are negligible.

So, as seen we can find a variety of particular processes that occur in estuarine and coastal zones that do not comply with the HPE. In such domains, the associated scales of motion correspond precisely to this gray area where the HPE lose their validity ($L \sim 1 - 10\,\mathrm{km}$). It is not uncommon however to see models based on the HPE used for studying such flows - and with success. The difficulty resides in knowing beforehand whether or not the non-hydrostatic effects need to be considered for a particular flow.

In consequence, the idea of developing a model that is a correction to the hydrostatic model appears quite naturally. Following this idea, the method applied here, already used in previous works (see [5] and [6] for instance), consists in decomposing the total pressure in two terms: the hydrostatic pressure, p_H, and the non-hydrostatic pressure q:

$$p = p_H + q \tag{1}$$

This thesis can be divided in three parts: theory, application to an- hopefully -efficient algorithm and results.

The theory part includes only the first chapter and it is basically the derivation of the equations for the model going from the Navier-Stokes equations to the Reynolds equations. In this part, turbulence modelling is addressed some importance to justify the equations used by this model.

The second part is opened with chapter 2, in which the obtained set of differential equations is transformed into an algebraic system so it can be solved (i.e. discretization). Chapter 3 is entirely devoted to the deduction of the numerical equations for pressure correction, q. At the end of those steps we have the algorithm that yields the system of algebraic equations for our model.

The resulting system of equations is a sparse system of linear equations. Chapter 4, "Solution to the System" that exposes different efficients algorithms for solving this kind of systems.

Finally, a model should always be validated, and it is precisely the objective of the last part of the study. In chapter 5, schematic cases are used to test the model. Internal waves (solitary wave) and wind waves (wave propagation over a bar) are studied. Also the Kelvin-Helmholtz instabilities are reproduced in a classical problem for fluid flow: the lock exchange problem.

Literature Survey

Tridimensional non-hydrostatic ocean modeling studies began in the 90's [7]. Earlier models developed a decade before use the rigid lid approximation or take advantage of symmetries to reduce spatial dimensions (see for example [4]). It is interesting to mention here the concept developed by Leschziner and Rodi ([8]) that uses a symmetry plane with imposed pressure gradient in order to represent free-surface slope.

Alternatively to fully non-hydrostatic models there are the quasi non-hydrostatic models in which the precise balance between gravity and pressure gradient is relaxed introducing an additional term in the pressure equation.

Marshall exposes a non-hydrostatic pressure correction model in [5]. The idea is to provide a model that is a correction to an hydrostatic correction to study the small scale phenomena in ocean which are not in hydrostatic balance. This model is said to have only a marginal increment in computational cost relatively to an hydrostatic code when the solution is hydrostatic. It has been applied to large-scale global circulation and as expected gives results that are only slightly different from the hydrostatic solution.

A similar approach is used by V. Casulli ([9]) to derive a non-hydrostatic semi-implicit algorithm with finite difference formulation, later brought to the more generic finite volume method (see [6]). The model has been applied to several problems including interfacial instabilities, gravity waves and a practical application: the Venice Lagoon. It uses a pressure correction method that is decomposed in to steps. In the first step momentum equations are solved for provisional velocity fields and water level. In the second step an elliptical equation is deduced for pressure correction to contemplate non-hydrostatic effects. This correction is then applied to velocity field and water level. This is exactly the methodology used in this thesis.

Mahadevan ([10] and [11]) proposes a slightly different model that reduces discrepancies between the non-hydrostatic and hydrostatic models results when hydrostatic approximation is valid. This assessment is not done in this work to our misfortune, but it is left as a suggestion for further work.

Most recently, interfacial instabilities have been studied by Fringer and Street in [1]. The mechanism of interfacial wave diffusion through secondary cross-stream convective rolls is studied (see figure 3).

It is shown that this mechanism is responsible for approximately half of the wave energy diffusion. Fringer has also a lot of material published in his website but unpublished elsewhere that inspired a part of this work. There is an important part on overturning and breaking waves which is also a highly diffusive process. This shows the importance of non-hydrostatic effects in interfacial

Figure 3: Cross-stream convetcive rolls along interfacial wave [1]

waves which is the case of an oscillating isopycnic (also called internal wave). The same author is involved in a project from the Stanford University (USA) called SUNTANS. In the website (*suntans.stanford.edu*) there is a mention to a study of internal waves in Montery Bay (California), but no paper is still available. This could be the first practical application of a tridimensional non-hydrostatic code.

About MOHID

MOHID[1] is a water modelling system created in the remote year of 1985 as a bi-dimensional shallow waters equations solver. It suffered numerous alterations along the course of years that lead to a tridimensional finite volume solver for free-surface equations[2] for subcritical flow. Although the principal aim of the model is to simulate estuarine and near-shore zones, it can be used with success at larger or smaller scales.

MOHID has well implemented a free-surface model solver, a transport model for different species and also several turbulence models. The core of MOHID is the hydrodynamics code module (*hydrodynamicmodule*) that is is still at the hydrostatic approximation evolution step and that will be modified in order to contemplate non-hydrostatic modeling.

MOHID is based on finite volume approach for structured othogonal grids. It has the advantage that the existing hydrodynamics code, that will be the starting point of this work, has already been extensively tested.

Apart from the hydrodynamics module, another important module will be used: the *waterproperties* module, which calculates transport through advection and difusion of any relevant quantity. As we saw earlier, internal waves are an important non-hydrostatic phenomenon that occur in startified domains. This can only be studied if one is able to reproduce correctly temperature and salinity transport that are the main variables that affect water density.

Finally, MOHID package provides very interesting features for practical applications, like the coupling with different water quality models and sediment transport models, and also a complete set of tools for processing data.

[1] Practical applications of the model and more detailed information can be found at the website *www.mohid.com*.

[2] Porous media flow and river hydraulics are also handeled by MOHID but those are really different apllications, so they are not commented here.

Chapter 1

The Equations of Motion

> "Perhaps a *definition* might be formulated somewhat more precisely as follows: *Turbulent fluid motion is an irregular condition of flow in which the various quantities show a random variation with time and space coordinates , so that statically distinct average can be discerned.*" **J. O. HINZE** [12]

1.1 From Navier-Stokes to Reynolds Equations

This cumulation of rethorical precautions in one single sentence by someone so familiar with the subject puts towards us the difficulty of defining rigorously turbulence motion. Even if we imagine intuitively what turbulence *might* be, getting into the core of the subject is a serious piece of work. While understanding turbulence is still a challenge.

In face of the difficulty of this matter and because it is not the purpose of this work, we will not go deeply into it. We cannot however get round turbulence, at least, if we stick to our principle of scientific integrity, because it is a matter of capital importance in fluid flows.

In the following paragraphs we will show the derivation of the system of equations for the model. It is therefore essential to give some lights on turbulence before going further because the model incorporates turbulence effects.

But to begin with, we need to go back in time two hundred years. In the first half of the XIXth century, two scientists reached the same equation. In 1822, Claude Navier, a french civil engineer from the École Nationale des Ponts et Chaussées was first successful deducing the formulation which is the very heart of fluid dynamics. In 1845, George Stokes, Irish mathematician at University of Cambridge, replicated the results of Navier. This duplication of results was not entirely an accident, but was rather brought about by the lack of knowledge of the work of continental mathematicians at Cambridge. Therefore, this equation has been given a double paternity, with the assurance that none of the fathers contributed less than the other to enrich fluid dynamics science.

The Navier-Stokes equation is based on Newton's second law. It relates the fluid acceleration to the surface forces and body forces experienced by the fluid. For an incompressible viscous fluid this can be written :

$$\frac{\partial u_j}{\partial t} + u_i \frac{\partial u_j}{\partial x_i} = \frac{1}{\rho} \frac{\partial \tau_{ij}}{\partial x_j} + b_j \tag{1.1}$$

Where u_i is the velocity in the i direction, ρ is the fluid density, τ_{ij} is the stress tensor and b_j a term that groups body forces in the j direction.

In the following we will only be interested in Newtonian fluids, for which shear stresses are proportional to velocity gradients. So the stress tensor is:

$$\tau_{ij} = -p\delta_{ij} + \nu\rho \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \tag{1.2}$$

Where p is the thermodynamic pressure. This term expresses dissipation at molecular level. That is why the coefficient ν is often referred to as molecular viscosity.

The combination of the two previous equations yields, finally, the Navier-Stokes equation[1]:

$$\frac{\partial u_j}{\partial t} + u_i \frac{\partial u_j}{\partial x_i} = -\frac{1}{\rho} \frac{\partial p}{\partial x_j} + \nu \triangle u_j + b_j \tag{1.3}$$

This, along with continuity equation for incompressible flows (equation 1.4), forms a closed system.

$$\nabla u = 0 \tag{1.4}$$

The solution of this system of equations is turbulent and the above expressions, that looks rather simple, contain all the complexity of turbulent flows - along with the frustration of generations of physicists and mathematicians[2].

Reminding Hinze citation it is plain to see that we would like to retrieve such statistics on the flow. Osborne Reynolds, eminent mathematician, pioneer scientist but also notable civil engineer, applied this concept to the Navier-Stokes equations. Let us decompose the instantaneous velocity in an average term plus a random term:

$$u = \overline{u} + u' \tag{1.5}$$

The idea comes naturally once it is intuitively clear that there exists some regularity in highly turbulent flows. Even with the naked eye we can easily distinguish regular patterns in flows such as flow past a bluff body (see figure 1.1), or cavity flows[3].

Turbulence motion can be seen as the superposition of eddies of various sizes and vorticities with distinguishable upper and lower limits. The upper size

[1] Intermediate steps are skipped. Formal derivation of those equations can be found in any fluid dynamics text book. Our purpose is just to make our writing consistent, before making the main deduction of the non-hydrostatic pressure component equation.

[2] To whom may feel concern: the Clay Mathematics Institute of Cambridge (Massachusetts) will reward with a prize of one million dollars the demonstration of existence and smoothness of N-S equation solution in $\mathbb{R}^3$.

[3] Should this be a proof, we placed here one of the famous sketches of Da Vinci that first depicted those regular patterns of the flow. Awfully wrong some would say, but he had no laser, no hotwire anemometer, no coherent structure detection technique, not the slight idea of who Joseph Fourier was, not even a single drop of rhodamine.

Figure 1.1: Sketches of flow past objects by Leonardo Da Vinci, circa 1500

limit of the eddies is determined mainly by the size of the apparatus, whereas the lower limit is determined by viscosity effects [12]. The eddies have a periodicity that is related to their size. The smaller eddies responsible for transformation of kinetic energy into heat vanish with frequencies that go up to 10 kHz, while the biggest have a periodicity that can range from a fraction of second to years (at very large scales, e.g. on the oceans). In general, and that is one of the issues with turbulence, the range of temporal and spacial scales is broad.

In this context, a problem that arises consists in defining an average. Is it a time average, a space average or an ensemble average? Although statistically, for an ideal sample wisely chosen, they should be the same, in practice this does not occur, and choosing one or another may change a lot[4]. Reynolds opted for a time average. The average is defined as:

$$\overline{u} = \frac{1}{T}\int_0^T u\,(t+\tau)\,d\tau \tag{1.6}$$

Actually turbulent flows are never really stationary. Therefore, for practical reasons, we cannot carry out the averaging procedures with respect to time for infinite values of T. The flow may contain slow variations that we do not whish to regard as belonging to the turbulent motion of the flow. For instance, a flow past a cylinder is periodic for a certain range of Reynolds number. This should be considered when calculating the drag force over the cylinder as the development of von Kármán vortices in the wake of the flow breed non negligible pressure fluctuations that may affect considerably the drag coefficient (Figure 1.2). Hence the time average has to contemplate a duration clearly smaller than the period of oscillation of the vortex-trail.

On the other hand, the time interval T must be sufficiently large compared to the scale of turbulence in order to isolate the irregular component of velocity.

Finally, averaging 1.3 and applying 1.6 in 1.3 we get the Reynolds Averaged Navier Stokes (RANS) equations:

[4]Treatment for this subject is extensively found in [13]. We stop here philosophical divagation despite of the interest of this topic.

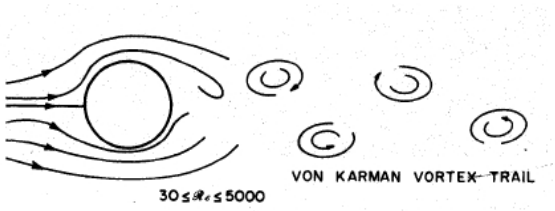


Figure 1.2: Flow past a cylinder for 30<Re<5000 [2]

$$\frac{\partial \overline{u}_j}{\partial t} + \overline{u}_i \frac{\partial \overline{u}_j}{\partial x_i} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_j} + \frac{\partial}{\partial x_i}\left(\nu \frac{\partial \overline{u}_j}{\partial x_i} - \overline{u'_i u'_j}\right) + b_j \tag{1.7}$$

1.2 Turbulence Modeling

"Big whorls have little whorls,
Which feed on their velocity;
And little whorls have lesser whorls,
And so on to viscosity
(in the molecular sense)." **L. F. RICHARDSON**

1.2.1 Eddy Viscosity Models

The equation 1.7 is similar to the Navier-Stokes equation, the difference being the cross correlation term ($\overline{u'_i u'_j}$) that does not vanish under the averaging operator. This introduces a new term in the system of equations that it is no longer closed. Hence, we need to model this term originated by the consideration of turbulent motion.

Put in this form the equation shows that turbulence has the same behaviour as the dissipative term. By analogy with the shear stresses, these terms are called Reynolds stresses. They are not real stresses, but as we can see, they have a similar effect on the flow. As a result of this last observation, the analogy between shear stresses and Reynolds stresses is very tempting and we might try to correlate the later with mean velocity gradients as in equation 1.2. This is called the turbulent viscosity or Boussinesq hypothesis and it writes:

$$\overline{u'_i u'_j} = \frac{2}{3} k \delta_{ij} - \nu_t \left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right) \tag{1.8}$$

The term of turbulent kinetic energy is used in order to guaranty that the trace of the matrix is indeed k defined as:

$$k = \overline{u'_i u'_i} \tag{1.9}$$

Given the turbulent viscosity field ν_t, equation 1.8 provides a most convenient closure to the Reynolds equations, which then have the same form as the Navier-Stokes equations:

$$\frac{\partial \overline{u}_j}{\partial t} + \overline{u}_i \frac{\partial \overline{u}_j}{\partial x_i} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_j} + (\nu + \nu_t) \triangle \overline{u}_j + b_j \tag{1.10}$$

Taking the new modified mean pressure as being:

$$\overline{p} \equiv \overline{p} + \frac{2}{3}\rho k \tag{1.11}$$

All models based on such an equation are inevitably called tubulent-or eddy-viscosity models. If the turbulent viscosity hypothesis is accepted as an adequate approximation, all that remains is to determine an appropriate specification for the eddy viscosity.

The eddy viscosity is generally about a thousand times the molecular viscosity. There are many models, more or less complex, available to calculate it, one of them, the Smagorinsky turbulent mixing-length is presented bellow.

We will consider this problem solved for the moment, and for notation purposes, we will use ν_v and $\nu_h$for vertical and horizontal effective viscosities, respectively.

We also consider the z-axis aligned with the vertical direction (i.e. the gravity gradient lines). Hence, we will have :

$$\frac{\partial \overline{u}_j}{\partial t} + \overline{u}_i \frac{\partial \overline{u}_j}{\partial x_i} = -\frac{1}{\rho}\frac{\partial \overline{p}}{\partial x_j} + \nu_h \triangle_h \overline{u}_j + \nu_v \triangle_v \overline{u}_j + b_j \tag{1.12}$$

1.2.2 Turbulent Mixing-Length Models

The simplest models used in turbulence modelling are the algebraic models. They relate directly the eddy viscosity to the mean quantities of the flow. The turbulent mixing-length model is based on the assumption that the turbulent viscosity can be expressed as a product of a velocity scale u^* and a lengthscale l_m.

$$\nu_t = u^* l_m \tag{1.13}$$

Consider a simple shear flow aligned with x axis, so equation 1.8 gives for the Reynolds shear stresses:

$$-\overline{uv} = -\nu_t \frac{\partial \overline{u}}{\partial y} \tag{1.14}$$

Either u^* or l_m can be specified at will. The Prandtl's mixing-length hypothesis assumes:

$$u^* = |\overline{uv}|^{1/2} \tag{1.15}$$

This last expression is used to relate turbulent viscosity to mean velocity gradients through the mixing length.

$$\nu_t = l_m^2 \left| \frac{\partial \overline{u}}{\partial y} \right| \tag{1.16}$$

The mixing-length depends on the nature of the flow and, in general, is space dependent. It has to be specified and the appropriate specification is inevitably dependent on the geometry of he flow. For a flow that has not been studied before, it requires a large measure of guesswork.

1.2.3 Smagorinsky Mixing-Length Model

Several generalizations of the previous have been proposed in order to allow the application of the mixing length hypothesis to all flows. On the basis on the mean rate of strain, Smagorinsky proposed:

$$\nu_t = l_m^2 \sqrt{2\overline{S}_{ij}\overline{S}_{ij}} \tag{1.17}$$

Where S_{ij} is the rate-of-strain tensor given by:

$$\overline{S}_{ij} = \frac{1}{2}\left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right) \tag{1.18}$$

Despite the fact this model suffers from incompleteness, this is the model we have used in this work, fortunately with success except in cases that were corrected with the anisotropic model presented bellow.

1.2.4 Anisotropic Model

The above model is isotropic. This can be particularly troublesome when applying this to a flow with anisotropic scales of motion.

This is why we already distinguished between vertical and horizontal eddy viscosities. In most circumstances we will be confronted to this problem, so we need to adapt the previous model. This is quite straightforward.

$$\nu_h^t = l_{mh}^2 \sqrt{2\left(\overline{S}_{xx}^2 + \overline{S}_{yy}^2 + 2\overline{S}_{xy}^2 + \overline{S}_{xz}^2 + \overline{S}_{yz}^2\right)} \tag{1.19}$$

$$\nu_v^t = l_{mv}^2 \sqrt{2\left(\overline{S}_{zz}^2 + \overline{S}_{zy}^2 + \overline{S}_{zx}^2\right)} \tag{1.20}$$

Moreover there are few chances that the mixing lengths will be the same in both scales of motion. Some trial and error estimations must be made, it is unfortunate but it is the only way to do this. A good start for estimating mixing length is to take the grid spacing or a reasonable fraction of it.

1.3 Free-Surface Model

As said before, free-Surface models often use the hydrostatic approximation. We will now derive the equations for the free surface model and see under which conditions the approximation is valid. From now on we will drop overbar of the mean operator, but only because we have decided to restrict our analysis to eddy viscosity models which allows us to consider only mean quantities of the flow, and forget about Reynolds stresses. It should be stressed that this is a particularity of the model, and therefore represents a limitation.

1.3.1 The Coriolis Effect

The governing equations generally used in ocean dynamics are the Navier-Stokes equations applied to a referential attached to earth surface. So, besides the gravity force that was not present before, the Coriolis forces, resulting from earth rotation, need also to be considered.

$$\frac{\partial u_j}{\partial t} + u_i \frac{\partial u_j}{\partial x_i} + f_j = -\frac{1}{\rho}\frac{\partial p}{\partial x_j} + \nu_h \triangle_h u_j + \nu_v \triangle_v u_j + g_j \tag{1.21}$$

$$f = \begin{pmatrix} 2w\omega\cos\varphi - 2v\omega\sin\varphi \\ 2u\omega\sin\varphi \\ 2u\omega\cos\varphi \end{pmatrix} = \begin{pmatrix} f_v w - f_h v \\ f_h u \\ f_v u \end{pmatrix} \tag{1.22}$$

Where ω is the angular velocity of earth rotation and φ the latitude of the place considered. Usually, f_v is neglected. This is required when the hydrostatic approximation is made for energy consistency [5]. This means the angular moment is only approximately conserved ([14] and [5]). A non-hydrostatic model should incorporate that parcel of the Coriolis force (implemented in [5]). But, as a first step in the implementation of the non-hydrostatic model, Coriolis forces will be neglected although we present their discretization and possible contribution to the model. Thus later correction will always be possible without having to rethink the whole model.

1.3.2 The Boussinesq Approximation

Another traditional approximation is the so-called Boussinesq approximation that exploits the fact that the density variation in he oceans are rather small, less than 3% or so, and therefore the density can almost be considered constant. However to contemplate buoyancy driven flows that are rather frequent we will consider variable density in gravity terms.

$$\frac{\partial u_j}{\partial t} + u_i \frac{\partial u_j}{\partial x_i} + f_j = -\frac{1}{\rho_0}\frac{\partial p}{\partial x_j} + \nu_h \triangle_h u_j + \nu_v \triangle_v u_j + \frac{\rho}{\rho_0} g_j \tag{1.23}$$

1.3.3 The Hydrostatic Approximation

In the vertical direction 1.23 equation writes:

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} + f_v u = -\frac{1}{\rho_0}\frac{\partial p}{\partial z} + \nu \triangle w + \frac{\rho}{\rho_0} g \tag{1.24}$$

The hydrostatic approximation assumes that vertical acceleration is small compared to other terms. As a result, every term is dropped but the gravity and pressure terms. We will see in the following a non-dimensional analysis that clarifies whenever this simplification is possible. In the end we have:

$$0 = -\frac{\partial p}{\partial z} + \rho g \tag{1.25}$$

1.3.4 Water Level Equation

Water level is referred to a reference water level, usually water level at rest, or the mean tide level (Figure 1.3). This gives the free-surface elevation, noted η. For the free surface, the kinematic boundary condition is:

$$\frac{\partial \eta}{\partial t} = w|_{z=\eta} \tag{1.26}$$

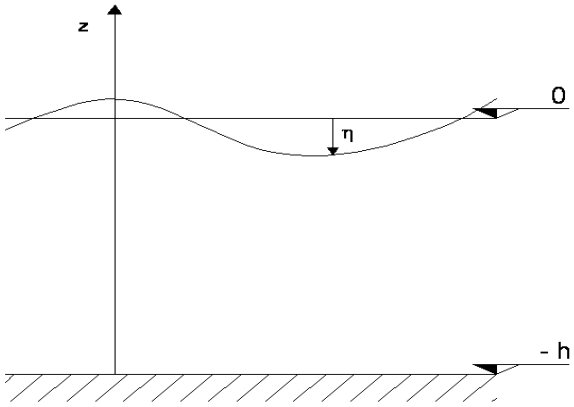


Figure 1.3: Water level definition

This simply traduces the idea that the flow field must be consistent with water surface movement.

Integrating the continuity equation (equation 1.4) along the vertical (z direction) and using the previous equation it is possible to obtain the equation for η in the system.

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}\left[\int_{-h}^{\eta} u dz\right] + \frac{\partial}{\partial y}\left[\int_{-h}^{\eta} v dz\right] = 0 \tag{1.27}$$

1.3.5 Boundary Conditions

Prescribed wind stresses can be imposed at the surface. Stresses are modeled as diffusive fluxes, the boundary condition being then:

$$\nu_h \left.\frac{\partial u}{\partial z}\right|_{z=\eta} = \tau_a^x = C_D \rho_a u_a \sqrt{u_a^2 + v_a^2} \tag{1.28}$$

$$\nu_h \left.\frac{\partial v}{\partial z}\right|_{z=\eta} = \tau_a^y = C_D \rho_a v_a \sqrt{u_a^2 + v_a^2} \tag{1.29}$$

In which u_a and v_a are wind velocity components in the x and y direction, respectively.

A similar condition can be applied at the bottom:

$$\nu_h \left.\frac{\partial u}{\partial z}\right|_{z=-h} = \tau_b^x = C_D \rho u \sqrt{u^2 + v^2} \tag{1.30}$$

$$\nu_h \left.\frac{\partial u}{\partial z}\right|_{z=-h} = \tau_b^y = C_D \rho v \sqrt{u^2 + v^2} \tag{1.31}$$

The drag coefficient is given this time using the quadratic friciton law given by:

$$C_D = \left[\frac{1}{\kappa}\ln\left(\frac{h+z_b}{\varepsilon_0}\right)\right]^{-2} \tag{1.32}$$

Where κ is the von Kármán universal constant, z_b is the distance between bottom and the point where the drag coefficient is calculated, and ε_0 is the roughness height. This formula implies that the bottom cell of the domain is situated within the log-law region. We will assume this last condition is always fullfilled-although a verification of this can only be made a posteriori and approximately.

This last expression completes the set of equations we will use in our model. In the following chapter,.further modification will put into evidence a non-hydrostatic term.

1.4 The Non-Hydrostatic Model

1.4.1 A Correction to the Hydrostatic Model

Behind the idea of the model as a correction to the hydrostatic case is the willingness of taking advantage of the hydrostatic hypothesis when possible and, more practically, to generalize a pre-existing model as it is the case.

The pressure is simply decomposed in its different parcels:

$$\frac{1}{\rho_0}p(x,y,z,t) = \underbrace{\frac{1}{\rho_0}p_a(x,y,t)}_{\text{Atmospheric}} + \underbrace{\underbrace{g\int_{-h}^{\eta}\frac{\rho-\rho_0}{\rho_0}d\zeta}_{\text{Baroclinic}} + \underbrace{g\left(\eta(x,y,t)-z\right)}_{\text{Barotropic}}}_{\text{Hydrostatic}} + \underbrace{q(x,y,z,t)}_{\text{Non-Hydrostatic}} \tag{1.33}$$

The last term, q, is the normalized non-hydrostatic pressure component that will be introduced in the system. We have then:

$$\begin{aligned}&\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} + f_v w - f_h v \\ &\quad = -\frac{1}{\rho_0}\frac{\partial p_a}{\partial x} - g\frac{\partial \eta}{\partial x} - g\frac{\partial}{\partial x}\int_{-h}^{\eta}\frac{\rho-\rho_0}{\rho_0}\,\mathrm{d}\zeta - \frac{\partial q}{\partial x} + \nu_h\triangle_h u + \nu_v\triangle_v u\end{aligned} \tag{1.34}$$

$$\begin{aligned}&\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} - f_h u \\ &\quad = -\frac{1}{\rho_0}\frac{\partial p_a}{\partial y} - g\frac{\partial \eta}{\partial y} - g\frac{\partial}{\partial y}\int_{-h}^{\eta}\frac{\rho-\rho_0}{\rho_0}\,\mathrm{d}\zeta - \frac{\partial q}{\partial y} + \nu_h\triangle_h u + \nu_v\triangle_v u\end{aligned} \tag{1.35}$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} + f_v u = -\frac{\partial q}{\partial z} + \nu_h\triangle_h w + \nu_v\triangle_v w \tag{1.36}$$

This last deduction is the all basis of our model. The assumption of hydrostatic pressure is equivalent of considering $q = 0$.

1.4.2 Non-Dimensional Analysis

The equation of motion can be written under a non-dimensional such that the dimensionless quantities are $O(1)$:

$$\begin{aligned} \widehat{x} = \frac{x}{L} \quad \widehat{y} = \frac{y}{L} \quad \widehat{z} = \frac{z}{H} \quad \widehat{u} = \frac{u}{U} \\ \widehat{v} = \frac{v}{U} \quad \widehat{w} = \frac{w}{W} \quad \widehat{t} = t\frac{U}{L} \end{aligned} \tag{1.37}$$

Note that in this in this analysis, it is important to distinguish between vertical and horizontal scales. The time scale is taken as the horizontal time scale, but we could have taken the vertical one as well. Using the divergence equation, we have that:

$$\frac{U}{L}\left(\frac{\partial \widehat{u}}{\partial \widehat{x}} + \frac{\partial \widehat{v}}{\partial \widehat{y}}\right) + \frac{H}{W}\frac{\partial \widehat{w}}{\partial \widehat{z}} = 0 \tag{1.38}$$

It follows that we should consider (luckily enough found in [10] and [15]):

$$\frac{W}{U} = \frac{H}{L} = \lambda \tag{1.39}$$

This implies that vertical and horizontal time scale are of the similar order of magnitude.

The reference pressure taken as:

$$\widehat{p} = \frac{p}{U^2} \tag{1.40}$$

We need to apply this analysis, not to the equations with q, but to the equations for the vertical momentum equation (substituting 1.37 and 1.39 in 1.23) we get:

$$\begin{aligned} \lambda\left(\frac{\partial \widehat{w}}{\partial \widehat{t}} + \widehat{u}\frac{\partial \widehat{w}}{\partial \widehat{x}} + \widehat{v}\frac{\partial \widehat{w}}{\partial \widehat{y}}\right) + \widehat{w}\frac{\partial \widehat{w}}{\partial \widehat{z}} + \mathrm{Ro}_v\,\widehat{u} \\ = \frac{1}{\lambda}\left(\frac{1}{\mathrm{Ri}} + \frac{1}{{\mathrm{Fr}_i}^2}\widehat{\delta\rho} - \frac{\partial \widehat{p}}{\partial \widehat{z}}\right) + \lambda\left(\frac{1}{\mathrm{Re}_h}\triangle_h \widehat{w} + \frac{1}{\mathrm{Re}_v}\triangle_v \widehat{w}\right) \end{aligned} \tag{1.41}$$

Where we have defined:

$$\mathrm{Ro}_h = \frac{fL}{U}, \text{ Rosby number} \tag{1.42}$$

$$\mathrm{Re}_h = \frac{UL}{\nu_h}, \text{ Reynolds number} \tag{1.43}$$

$$\mathrm{Re}_v = \frac{WH}{\nu_v}, \text{ Reynolds number} \tag{1.44}$$

$$\mathrm{Ri} = \frac{U^2}{gH}, \text{ Richardson number} \tag{1.45}$$

$$\mathrm{Fr}_i = \frac{U}{\sqrt{\frac{\delta\rho_0}{\rho_0}gH}}, \text{ Internal Froude number} \tag{1.46}$$

	Ocean	Estuary
H	$1\,\mathrm{km}$	$10\,\mathrm{m}$
L	$1000\,\mathrm{km}$	$1\,\mathrm{km}$
U	$0.1\,\mathrm{m\,s^{-1}}$	$1\,\mathrm{m\,s^{-1}}$
W	$10^{-4}\,\mathrm{m\,s^{-1}}$	$10^{-2}\,\mathrm{m\,s^{-1}}$
λ	10^{-3}	10^{-2}
Re_h	10^8	10^6
Re_v	10^2	10^3
Ro_v	10^2	10^{-2}
Ri	10^{-6}	10^{-2}
Fr_i	10^{-2}	1

Table 1.1: Estimation of Relevant Non-Dimensional Numbers

From this it becomes clear when approximations cited in the above paragraphs are possible.

In the hydrostatic approximations we assume all terms negligible except gravity and pressure term.

In the following table we give typical values for the non-dimensional numbers (we considered $f_v \simeq 10^{-5}\,\mathrm{s^{-1}}$ and $\nu \simeq 10^{-6}\,\mathrm{m^2\,s^{-2}}$).

This shows the decisive role of λ in the validity of hydrostatic approximation. Indeed, pressure terms, and other terms of the equation have a ratio of λ to λ^2. As one could have intuitively stated before, as the relation between vertical and horizontal scales increases, the flow becomes less and less bi-dimensional, leading to the need of considering non-hydrostatic effects.

For an estuarine region, the increase in Richardson and internal Froude numbers indicates that buoyancy and gravity effects in general can be predominant. With buoyancy and movement of stratified layers, non-hydrostatic processes are more likely to appear (e.g. internal waves). Nevertheless, the Boussinesq approximation does not break down in estuaries. The gravity term is still the most important even with $\mathrm{Ri} \simeq 10^{-2}$. Indeed it is proportional to $\frac{1}{\lambda^2\,\mathrm{Ri}} \simeq 10^8$

It is also clear from Rossby number, Ro_v, that as we go to a smaller scale we get rid of Coriolis effect-this is not a surprise of course, but we wanted to mention it. Note that the horizontal Rossby number is about the same. As said before the vertical parcel is only neglected for energy consistency with the hydrostatic approximation. In our model, the vertical component of the Coriolis acceleration will be considered.

The Reynolds numbers quantify the ratio between inertial and viscous forces. In oceans, viscous effects can often be neglected, at least if we only take into account molecular viscosity of the fluid. But, in our eddy viscosity model, the viscosity is the sum of the molecular viscosity plus the turbulent eddy viscosity which is, in general, about a thousand times bigger than molecular viscosity. According to the table above it would be reasonable to drop the horizontal diffusion term, but not the vertical one. As a matter of fact, an usual practice consists in having a very simple turbulence model for determining horizontal eddy viscosity and a more accurate one for the vertical transport.

From non-dimensional analysis we conclude that the only approximation we can reasonably accept is the Boussinesq approximation.

Chapter 2

Equations Discretization

2.1 Finite Volume Method

Until now, and to respect to a widespread convention, we presented the equations in their differential form. But there exists another form for the conservation equations: the integral form. The finite volume method uses them as the starting point:

$$\underbrace{\frac{\partial}{\partial t}\int_{\Omega}\phi \,\mathrm{d}\Omega}_{\text{Time Derivative}} + \underbrace{\int_{S}\phi\mathbf{v}\cdot\mathbf{n}\,\mathrm{d}S}_{\text{Advection}} = \underbrace{\int_{S}\Gamma\boldsymbol{\nabla}\phi\cdot\mathbf{n}\,\mathrm{d}S}_{\text{Diffusion}} + \underbrace{\int_{\Omega}s\,\mathrm{d}\Omega}_{\text{Sources-Sinks}} \tag{2.1}$$

The equation 2.1 expresses the conservation of a generic quantity, ϕ. In the finite volume approach, the solution domain is subdivided into a finite number of small control volumes (CVs). Usually, the computational node is assigned to the CV centroid. The integral equation (2.1) applies to each CV as well as to the solution domain as a whole. To obtain an algebraic equation for each CV-the discretized equations-the volume integral need to be approximated using a quadrature formulae.

The net flux through the CV boundary is the sum of integrals over the six CV faces:

$$\int_{S}\phi\,\mathrm{d}S = \sum_{s}\int_{Ss}\phi\,\mathrm{d}S \tag{2.2}$$

Using the mean theorem for integrals:

$$\int_{S_s}\phi\,\mathrm{d}S = \overline{\phi_s}S_s \tag{2.3}$$

Where $\overline{\phi_s}$ is the mean value over the s^{th} facet. The same can readily be applied to the volume integral:

In the case of MOHID we can admit the facets of the hexahedral CV aligned with axes [1]. Hence, the above integral is simply the sum over the six facets of the CV.

[1] In the case we use a curvilinear grid this assumption is untrue, but MOHID provides internally the necessary correction to the equations. In fact it is only an additional term that appears in the equations due to cell rotation with respect to main coordinate system.

$$\int_{\Omega} \phi \, \mathrm{d}\Omega = \overline{\phi} S \tag{2.4}$$

If the mean values are decomposed as an algebraic expression, we can retrieve the desired system of discrete equations. The way the mean values are calculated determines the numerical scheme, and hence the order and the stability of the method.

We will not present, in the following, the discretization for this generic transport equation although MOHID uses it for different properties (e.g. temperature and salinity). Details for this can be found in [16] and [17],or even in [6]. Its derivation can easily be deducted from the momentum equations discretization.

2.2 Notations and difference operators

In the following we will use a very specific notation that will be useful at a later stage of our derivation. Most of our fields are tridimensional, and is presented in a structured way so any location on the grid can be represented by a triplet. Let j, k, l be the indices of the triplet. The indices ranges are:

$$j = 1, .., N_j \tag{2.5}$$
$$k = 1, .., N_k \tag{2.6}$$
$$l = 1, .., N_l \tag{2.7}$$

The triplet can advantageously be replaced by an index vector, $\mathbf{j}$.

$$\mathbf{j} \equiv (j, k, l)$$

A displacement index vector can also be defined:

$$\boldsymbol{\delta}(s) = (\delta_{1s}, \delta_{2s}, \delta_{3s}) \tag{2.8}$$

Where δ_{ij} is the Kronecker symbol.

In the following we will need the two difference operators, Δ_s and ∇_s defined as follows.

$$\Delta_s (\cdot)_{\mathbf{j}} = (\cdot)_{\mathbf{j}} - (\cdot)_{\mathbf{j}-\boldsymbol{\delta}(s)} \tag{2.9}$$
$$\nabla_s (\cdot)_{\mathbf{j}} = (\cdot)_{\mathbf{j}+\boldsymbol{\delta}(s)} - (\cdot)_{\mathbf{j}} \tag{2.10}$$

The first is called Backwards Difference Operator (BDO) and the former forward difference operator (FDO). These operators are very useful in any structured approach for interpolations and derivatives estimations, as we will see later.

We stress here the fact that we will use most often s as a generic direction.

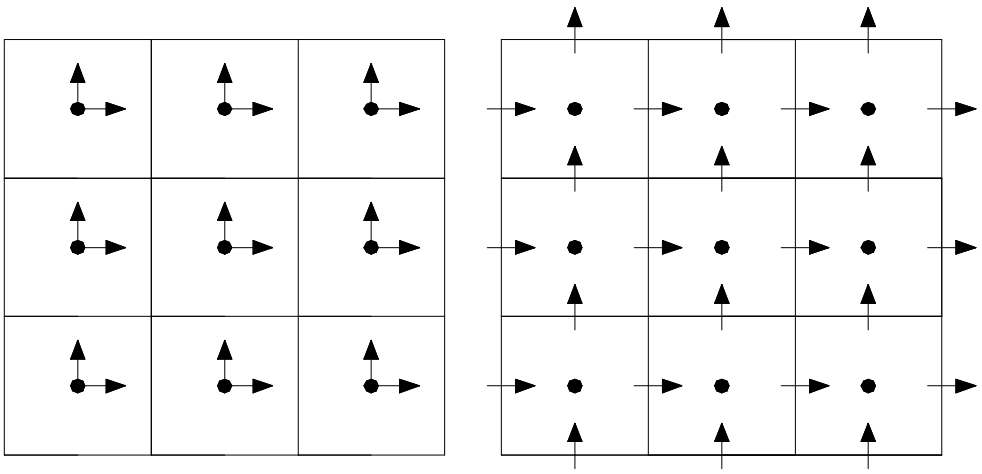


Figure 2.1: Collocated arrangment (left) and fully staggered arrangment of velocity components and other variables (right)

2.3 Data Arrangement

In this model, we will consider that the mean value over an element (face or cell) is given by the value at the centroid of the element. This is called the midpoint rule. The midpoint rule implies thereby that values are interpolated at various locations. But the locations where the interpolations must be done depend on the arrangement we have chosen for our problem.

There is no need for all variables to share the same grid; a different arrangement may turn out to be advantageous. Amongst the different choices to arrange the data over a grid, in the model we have applied the fully staggered arrangement. In the fully staggered grid arrangement, the components of velocity normal to the face are stored at the faces centers and the remaining variables are stored at the cell centers. Staggered grids arrangements are opposed to collocated grid arrangement where both velocities and other variables-such as pressure and densities-are stored at the cell centers. The two schemes are represented in figure 2.1.

The staggered arrangement has the advantage that velocities are given at the element faces. This guaranties local and global mass conservation. Another big advantage, according to [18], is that there is a strong coupling between the velocities and the pressure, avoiding some types of convergence problems and oscillations in pressure and velocity fields.

So, prior to decide the type of interpolation we will apply, it is important to know what kind of arrangement we have. MOHID early programmers have opted for a staggered arrangement. It is crucial to note that this affects the way control volumes are placed. When calculating the transport of any mass quantity, the volumes are centered on the grid, as expected. But when calculating momentum (balance (transport of velocity by itself) the CV are displaced half a step in the direction of the velocity being considered (see figure 2.2). To be consistent with the staggered approach, the CV's used for momentum equations

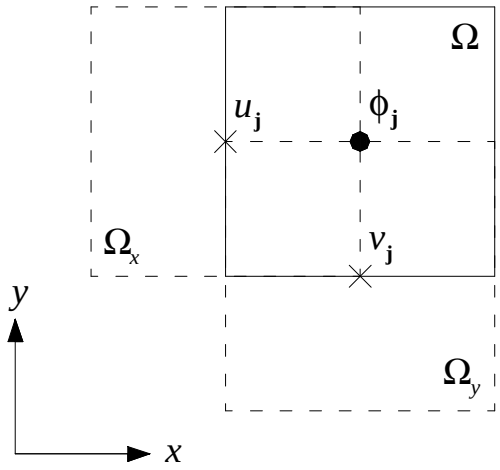


Figure 2.2: CV for mass properties (Ω), x velocities (Ω_x) and y velocities (Ω_y)

must be centered at the velocities locations. This may look like a detail, but as we will see later, it is not.

2.4 Interpolation Schemes

The simplest interpolation is the upwind interpolation. The value chosen is the value from closest upstream node. For instance to interpolate the value of a generic variable ϕ at the face $\mathbf{j} + \boldsymbol{\delta}(s)/2$:

$$\phi_{\mathbf{j}+\boldsymbol{\delta}(s)/2} = \begin{cases} \phi_{\mathbf{j}+\boldsymbol{\delta}(s)} \text{ if } u_{\mathbf{j}+\boldsymbol{\delta}(s)/2} > 0 \\ \phi_{\mathbf{j}} \text{ if } u_{\mathbf{j}+\boldsymbol{\delta}(s)/2} < 0 \end{cases} \tag{2.11}$$

Then we have linear interpolations which write:

$$\phi_{\mathbf{j}+\boldsymbol{\delta}(s)/2} = \lambda \phi_{\mathbf{j}+\boldsymbol{\delta}(s)} + (1-\lambda)\,\phi_{\mathbf{j}} \qquad \text{with } 0 < \lambda < 1 \tag{2.12}$$

It is a first order approximation, unless $\lambda = 0.5$, in which case the scheme is second order and it is called central difference approximation.

Higher order schemes exist of course. They can be linear or not, and involve values at more than two locations.

All the above applies also to time interpolation. In general, we will use:

$$\phi_{\mathbf{j}}^{n+1/2} = \theta \phi_{\mathbf{j}}^{n+1} + (1-\theta)\,\phi_{\mathbf{j}}^{n} \qquad \text{with } 0 < \theta < 1 \tag{2.13}$$

Of course, we can be forced to mix time and space interpolations which can become a bit intricate.

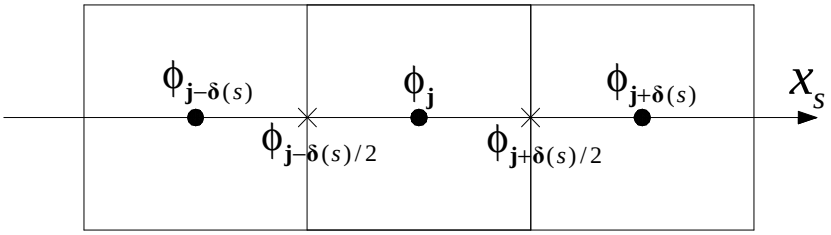


Figure 2.3: Notation of indices for a generic variable ϕ in the x_s direction.

2.5 Derivative estimations

In the course of the discretization procedure we will need to calculate derivatives. Space derivatives can be calculated in many different ways. Using a Taylor series expansion we have that:

$$\left(\frac{\partial\phi}{\partial x_s}\right)_{\mathbf{j}} = \frac{\phi_{\mathbf{j}+\boldsymbol{\delta}(s)} - \phi_{\mathbf{j}}}{x_{\mathbf{j}+\boldsymbol{\delta}(s)} - x_{\mathbf{j}}} + O(\Delta x) = \frac{\nabla_i \phi_{\mathbf{j}}}{\nabla_i x_{\mathbf{j}}} + O(\Delta x) \tag{2.14}$$

$$\left(\frac{\partial\phi}{\partial x_s}\right)_{\mathbf{j}} = \frac{\phi_{\mathbf{j}} - \phi_{\mathbf{j}-\boldsymbol{\delta}(s)}}{x_{\mathbf{j}} - x_{\mathbf{j}-\boldsymbol{\delta}(s)}} + O(\Delta x) = \frac{\Delta_i \phi_{\mathbf{j}}}{\Delta_i x_{\mathbf{j}}} + O(\Delta x) \tag{2.15}$$

$$\left(\frac{\partial\phi}{\partial x_s}\right)_{\mathbf{j}} = \frac{\phi_{\mathbf{j}+\boldsymbol{\delta}(s)} - \phi_{\mathbf{j}-\boldsymbol{\delta}(s)}}{x_{\mathbf{j}+\boldsymbol{\delta}(s)} - x_{\mathbf{j}-\boldsymbol{\delta}(s)}} + O(\Delta x^2) \tag{2.16}$$

The first scheme is called forward difference scheme (FDS), the second backward difference scheme (BDS), and the latest central difference scheme (CDS).

The schemes presented above can been used either in space or in time. Usually, for the time derivatives we will use the FDS, while for space derivatives we will use CDS or even more complex schemes.

A time derivative would write:

$$\left(\frac{\partial\phi}{\partial t}\right)^n = \frac{\phi^{n+1} - \phi^n}{\Delta t} + O(\Delta t) \tag{2.17}$$

At this point, there is a balance that needs to be done, between the accuracy of the numerical scheme and the complexity of the resulting system. MOHID gives preference to low order numerical schemes providing an high performance solver and uses time discretization schemes that are at most first order accurate in time. The lack of accuracy is then compensated with more restrictive Δt and Δx.

In the later paragraphs, in order to compress a bit the notation, we will use:

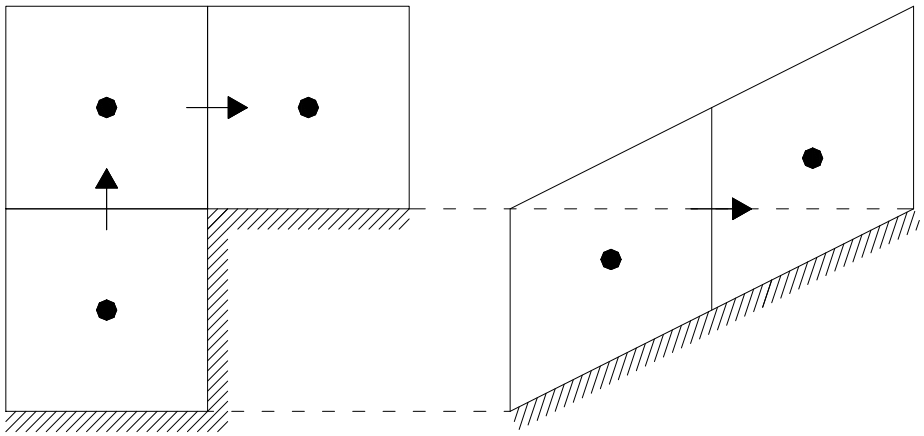


Figure 2.4: Representation of a cartesian mesh (left) and a sigma mesh (right) layers near the bottom

$$(\delta x_s)_{\mathbf{j}} = \Delta_s (x_s)_{\mathbf{j}+\boldsymbol{\delta}(s)/2} \tag{2.18}$$

2.6 Grid Geometry

Before getting into the core of discretization, we must add a note on grid geometries. For horizontal discretization, MOHID has implemented generic curvilinear grids, but we will only use uniform rectangular grids. As far as vertical discretization is concerned, MOHID allows for the use of two type of grids (or a combination of those two types): sigma and cartesian mesh types.

The sigma grid is a grid that follows the bottom and adapts cells height uniformly according to water surface deformation. Cartesian mesh is fixed except for the last layer that moves with surface elevation.

The first type of grid introduces additional error because cells have a slope which is not contemplated in the model. This brings more diffusion and errors in energy conservation. However, this grid is smooth and near the bottom the flow is better represented.

The sigma grid topology generates too much diffusion. In coarse meshes this can be quite troublesome. Moreover, element size is not uniform and it is difficult to control element size. This can lead to local instability if elements are too small. The Cartesian meshes have the disadvantage of not following the bottom and creating a sort of macro-roughness in slopes that come to have the appearance of steps. Those steps can be a source of trouble as they constitute stagnation points.

As far as we know, the choice of the grid is far from being obvious. For instance, we know that if we want to represent a long, uniform slope, we better

use the sigma grid. While if we want to use this to represent a more irregular bottom, with abrupt slopes, the recommended mesh will be the cartesian that ensure momentum conservation.

2.7 Continuity Equation

This is the easiest step. In the integral form, the balance of flux through CV surface is:

$$\int_{\Omega} (\mathbf{v} \cdot \mathbf{n}) \ \mathrm{d}S = 0 \tag{2.19}$$

And then we see how fortunate we are to have a staggered arrangement so velocity values are at faces centers, and more, the faces are aligned with the axis[2]. There are various declinations of the difference equation, here's one with the BDS:

$$\sum_{i=1}^{3} \Delta_i \left(F_i\right)_{\mathbf{j}}^{n^*} = 0 \tag{2.20}$$

Where

$$F = uS \tag{2.21}$$

Which is:

$$\sum_{i=1}^{3} \Delta_i \left[\theta_i \left(F_i\right)_{\mathbf{j}}^{n+1} + (1-\theta_i)\left(F_i\right)_{\mathbf{j}}^{n}\right] = 0 \tag{2.22}$$

Note that we could have used the FDS:

$$\sum_{i=1}^{3} \nabla_i \left[\theta_i \left(F_i\right)_{\mathbf{j}}^{n+1} + (1-\theta_i)\left(F_i\right)_{\mathbf{j}}^{n}\right] = 0 \tag{2.23}$$

That is also valid. This is the expression we will use in later stages. That is why we mention it.

We will use this equation at different stages of the calculation, but mostly, at the end to get a conservative velocity field.

2.8 Momentum Equations

We show here the result of the previous applied to the set of differential equations. We first give the equations in their integral form (equations 2.24).

$$\begin{aligned} \frac{\partial}{\partial t}\int_{\Omega} u_i \ \mathrm{d}\Omega + \int_{S} u_i \left(\mathbf{v} \cdot \mathbf{n}\right) \ \mathrm{d}S + \int_{\Omega} f_i \ \mathrm{d}\Omega \\ = -\int_{\Omega} \frac{1}{\rho_0}\frac{\partial p}{\partial x_i} \ \mathrm{d}\Omega + \int_{S} \left(\nu^m \frac{\partial u_i}{\partial x_m}\boldsymbol{i}_m\right) \boldsymbol{\cdot n} \ \mathrm{d}S \end{aligned} \tag{2.24}$$

[2] Remember also that we have hexahedral cells

This is lengthy and in the following we separate the equations in their physical terms: time derivative, convection, diffusion, Coriolis terms, and finally, pressure.

2.8.1 Time derivative

The time derivative is decomposed using a FDS[3]:

$$\frac{\partial}{\partial t}\int_{\Omega} u_i \, \mathrm{d}\Omega = \int_{\Omega} \frac{\partial u_i}{\partial t} \, \mathrm{d}\Omega \simeq \frac{(\Omega_i)^n_{\mathbf{j}}}{\Delta t}\left[(u_i)^{n+1}_{\mathbf{j}} - (u_i)^n_{\mathbf{j}}\right] \tag{2.25}$$

2.8.2 Convective Acceleration

For each CV, the integral can be decomposed in a sum of integral over the faces. Each of those integrals is approximated by the product of the face surface times an estimation of the mid point value of the integrand.

$$\int_S u_i \left(\mathbf{v}\cdot\mathbf{n}\right) \, \mathrm{d}S \simeq \sum_{s=1}^{3} \Delta_s \left[(u_i)^{n^*}_{\mathbf{j}+\boldsymbol{\delta}(s)/2} \, (F_s)^n_{\mathbf{j}+\boldsymbol{\delta}(s)/2}\right] \tag{2.26}$$

Here appears one of the drawbacks of the staggered arrangement. It is indeed very appropriate for calculating the advection of a mass property-a quantity calculated at the CV centers-, once the fluxes are calculated at the faces. Whereas for applying the momentum equations, it is, the advection of velocity itself, we need to interpolate advective fluxes at the faces. Using linear interpolations:

$$(u_i)^{n^*}_{\mathbf{j}+\boldsymbol{\delta}(s)/2} = \frac{1}{2}\left[(u_i)^{n^*}_{\mathbf{j}+\boldsymbol{\delta}(s)} + (u_i)^{n^*}_{\mathbf{j}}\right] \tag{2.27}$$

$$(F_i)^{n}_{\mathbf{j}+\boldsymbol{\delta}(s)/2} = \frac{1}{2}\left[(F_i)^{n^*}_{\mathbf{j}+\boldsymbol{\delta}(s)} + (F_i)^{n^*}_{\mathbf{j}}\right] \tag{2.28}$$

Only the velocity term is time interpolated, the values for the flux being taken at the instant n:

$$(u_i)^{n^*}_{\mathbf{j}} = \theta_i \, (u_i)^{n+1}_{\mathbf{j}} + (1-\theta_i)\,(u_i)^n_{\mathbf{j}} \tag{2.29}$$

It is important to note MOHID allows for higher order interpolation schemes although in that case the higer order scheme must be fully explicit ($\theta_i = 0$). For simplicity sake, we will consider only one type of discretization for this term, that nevertheless permits to expose the method fully and in a way that embraces all numerical issues. The same problem applies to the next-diffusion-term.

2.8.3 Diffusion Term

Using the CDS for estimating the derivative, the diffusion term can be discretized as:

$$\int_S \left(\nu_m \frac{\partial u_i}{\partial x_m}\boldsymbol{i}_m\right)\cdot\mathbf{n} \, \mathrm{d}S \simeq \sum_{s=1}^{3} \Delta_s \left[(\nu_s)^n_{\mathbf{j}+\boldsymbol{\delta}(s)/2}\left(\frac{\partial u_i}{\partial x_s}\right)^{n^*}_{\mathbf{j}+\boldsymbol{\delta}(s)/2} (S_s)^n_{\mathbf{j}+\boldsymbol{\delta}(s)/2}\right] \tag{2.30}$$

[3] The first equality holds because Ω is a closed volume.

The derivative is interpolated with a FDS (so at the end we retrieve only $\mathbf{j-1},\ \mathbf{j}, \mathbf{j+1}$):

$$\left(\frac{\partial u_i}{\partial x_s}\right)^{n^*}_{\mathbf{j}+\boldsymbol{\delta}(s)/2} = \frac{\nabla_s \left(u_i\right)^{n^*}_{\mathbf{j}}}{\left(\delta x_s\right)^n_{\mathbf{j}}} \tag{2.31}$$

The eddy viscosity can be calculated using any convenient closure equation. Smagorinsky mixing length model has been exposed before. In our case we will use either Smagorinsky, either a constant eddy viscosity model. In the former, the viscosity is considered isotropic.

To apply Smagorinsky, we use again a CDS:

$$\nu_t = l_m^2 \sqrt{2 S_{ij} S_{ij}} \tag{2.32}$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \simeq \frac{1}{2}\left(\frac{\Delta_j \left(u_i\right)^n_{\mathbf{j}}}{\left(\delta x_j\right)^n_{\mathbf{j}}} + \frac{\Delta_i \left(u_j\right)^n_{\mathbf{j}}}{\left(\delta x_i\right)^n_{\mathbf{j}}}\right) \tag{2.33}$$

It is important to keep this term explicit otherwise we have cross-terms that would result in a non-linear system.

The diffusion is, as the advection, a semi-implicit term. Thereby, the velocity needs also to be time-interpolated.

2.8.4 The Coriolis Term

In this model, the Coriolis term is kept fully explicit. To apply the mid-point rule, the term must be interpolated in space, because control volumes are not centered with the required velocity value:

$$\int_{\Omega_x} f_x \ \mathrm{d}\Omega = \int_{\Omega_x} f_h v \ \mathrm{d}\Omega \simeq f_h \left(v_i\right)^n_{j+\frac{1}{2},k-\frac{1}{2},l} \left(\Omega_x\right)^n_{\mathbf{j}} \equiv \left(f_x\right)^n_{\mathbf{j}} \left(\Omega_x\right)^n_{\mathbf{j}} \tag{2.34}$$

$$\int_{\Omega_y} f_y \ \mathrm{d}\Omega = \int_{\Omega_x} f_h u \ \mathrm{d}\Omega \simeq f_h \left(u_i\right)^n_{j-\frac{1}{2},k+\frac{1}{2},l} \left(\Omega_y\right)^n_{\mathbf{j}} \equiv \left(f_y\right)^n_{\mathbf{j}} \left(\Omega_x\right)^n_{\mathbf{j}} \tag{2.35}$$

2.8.5 The Pressure Term

This term is the volume integral of a space derivative. The space derivative is estimated with a CDS which gives:

$$\frac{1}{\rho_0}\int_\Omega \frac{\partial p}{\partial x_i}\ \mathrm{d}\Omega = \int_\Omega \left(\frac{1}{\rho_0}\frac{\partial p_a}{\partial x_i} + g\frac{\partial \eta}{\partial x_i} + g\frac{\partial}{\partial x_i}\int_{-h}^{\eta} \frac{\rho-\rho_0}{\rho_0}\ \mathrm{d}\zeta - \frac{\partial q}{\partial x_i}\right)\ \mathrm{d}\Omega \tag{2.36}$$

We will first concentrate on the baroclinic and the atmospheric pressure terms.

$$\int_\Omega \frac{\partial p_a}{\partial x_i}\ \mathrm{d}\zeta \simeq \frac{\Delta_i \left(p_a\right)^{n^*}_{\mathbf{j}}}{\left(\delta x_i\right)^n_{\mathbf{j}}} \left(\Omega_i\right)^n_{\mathbf{j}} \qquad i \neq 3 \tag{2.37}$$

In the next equation appears a line integral inside the volume integral.

$$\int_\Omega g\frac{\partial}{\partial x_i}\left(\int_{-h}^{\eta}\frac{\rho-\rho_0}{\rho_0}\,\mathrm{d}\zeta\right)\,\mathrm{d}\Omega \simeq \frac{g\left(\Omega_i\right)_{\mathbf{j}}^n}{\left(\delta x_i\right)_{\mathbf{j}}^n}\Delta_i\left[\frac{1}{\rho_0}\sum_{\zeta=1}^{N_l}\left(\rho_{jk\zeta}-\rho_0\right)\Delta_z\left(z_j\right)_{jk\zeta}^n\right] \qquad i\neq 3 \tag{2.38}$$

To shorten the writing of this heavy term, we state that:

$$P_{\mathbf{j}} = \sum_{\zeta=1}^{N_l}\left(\rho_{jk\zeta}-\rho_0\right)\Delta_z\left(z_j\right)_{jk\zeta}^n \tag{2.39}$$

Now the free-surface term. The main problem with it, is that we want to have it time-interpolated. And for that we will use the same semi-implicit coefficients as for velocity.

$$g\int_\Omega\frac{\partial\eta}{\partial x_i}\,\mathrm{d}\Omega \simeq g\frac{\left(\Omega_i\right)_{\mathbf{j}}^n}{\left(\delta x_i\right)_{\mathbf{j}}^n}\Delta_i\left[\theta_i\eta_{\mathbf{j}}^{n+1}+\left(1-\theta_i\right)\eta_{\mathbf{j}}^n\right] \qquad i\neq 3 \tag{2.40}$$

The same applies to the non hydrostatic term:

$$\int_\Omega\frac{\partial q}{\partial x_i}\,\mathrm{d}\Omega \simeq \Delta_i\frac{\left(\Omega_i\right)_{\mathbf{j}}^n}{\left(\delta x_i\right)_{\mathbf{j}}^n}\left[\theta_i q_{\mathbf{j}}^{n+1}+\left(1-\theta_i\right)q_{\mathbf{j}}^n\right] \qquad i=1,2,3 \tag{2.41}$$

This is true in theory. To develop a non-hydrostatic model we will need to modify the last equation.

2.9 Free Surface Equation

We will not use the free surface equation as seen before, applied to a whole water column. We are interested in the continuity equation applied to the surface cell.

If we make the mass balance at the top cell, we have that :

$$\left(S_z\right)_{\mathbf{j}}^{n+1}\frac{\eta_{\mathbf{j}}^{n+1}-\eta_{\mathbf{j}}^n}{\Delta t}+\Delta_x\left(F_x\right)_{\mathbf{j}}^{n^*}+\Delta_y\left(F_y\right)_{\mathbf{j}}^{n^*}+\left(F_z\right)_{\mathbf{j}}^{n^*}=0 \tag{2.42}$$

In truth it is nothing but a mass balance at the upper cell of the domain. This last equation is used to close our system.

2.10 Boundary Conditions

2.10.1 Dirichlet Type Boundaries

As a general rule, either we have imposed values (Dirichlet boundaries) or we have imposed fluxes (Neumann boundary type). Those two types of boundary must be treated differently.

At the faces adjacent to boundaries, we can have imposed values for velocity. On the face with normal perpendicular to the boundary, the velocity is zero if the boundary is closed, but it can be any imposed value if the boundary is opened. On any other faces, the velocity is zero (see figure 2.5). The imposed

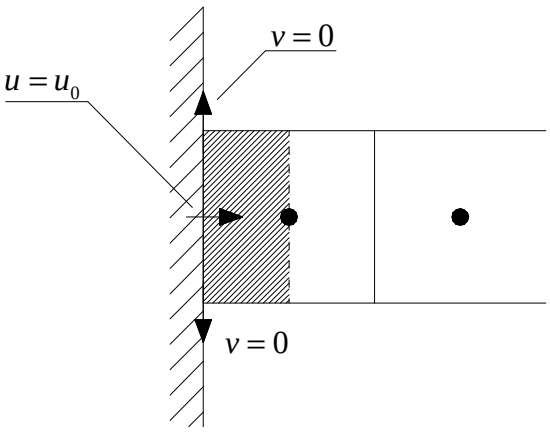


Figure 2.5: Schematic view of Dirichlet boundary conditions treatment

values of velocity are used to calculate convective and diffusive fluxes at CV faces.

The same applies if we have imposed mass properties values imposed at boundary cells.

2.10.2 The von Neumann Type Boundaries

A von Neuman boundary condition appears when specifying shear stresses at boundary faces. As we saw earlier, it is often written in the form:

$$\tau = \nu \boldsymbol{\nabla} u \cdot \mathbf{n} \tag{2.43}$$

It can hence be treated exactly as a diffusive flux on the boundary face. This causes no trouble except in the case of bottom friction where it is possible that reverse flow occurs only by the effect of friction. This is of course unrealistic and attention should be paid in the numerical discretization of this term so this does not occur. In [19], a specific numerical scheme is developed in order to avoid reverse flow near walls.

Chapter 3

The non-hydrostatic correction model

3.1 A Two Steps Method

The model has been developed inspired having in mind that it should work as a correction to the hydrostatic model. We have fortunately an hydrostatic model that works.

We will decompose the solution method in two steps. In the first step we will calculate a provisional solution by neglecting the implicit contribution of the non-hydrostatic pressure. The second step is a corrective step that affects pressure, but also velocities and water levels. Those are calculated in such a way that the resulting field is globally and locally mass conservative.

3.2 First Step: Hydrostatic Pressure

The first step consists in performing the calculation while neglecting the implicit part of the non-hydrostatic pressure.

As we omit a part of the discretization, the resulting velocity and water surface elevation field that we will note $\tilde{u}_i, \tilde{\eta}$ need to be corrected. Those are called the provisional velocities and elevation.

The momentum equation becomes:

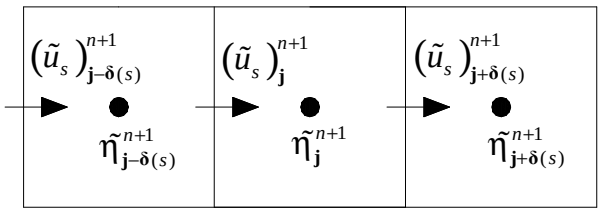


Figure 3.1: Calculation molecule in the generic s direction

$$
\begin{aligned}
&(\tilde{u}_i)^{n+1}_{\mathbf{j}} - (u_i)^n_{\mathbf{j}} + \Delta t\,(f_i)^n_{\mathbf{j}} \\
&+\frac{\Delta t}{(\Omega_i)^n_{\mathbf{j}}}\sum_{s=1}^{3}\Delta_s\left\{\left[\frac{\theta_i}{2}\left((\tilde{u}_i)^{n+1}_{\mathbf{j}+\boldsymbol{\delta}(s)} + (\tilde{u}_i)^{n+1}_{\mathbf{j}}\right) + (1-\theta_i)\,(u_i)^n_{\mathbf{j}+\boldsymbol{\delta}(s)/2}\right](F_s)^n_{\mathbf{j}+\boldsymbol{\delta}(s)/2}\right\} \\
&-\frac{\Delta t}{(\Omega_i)^n_{\mathbf{j}}}\sum_{s=1}^{3}\Delta_s\left\{(\nu_s)^n_{\mathbf{j}+\boldsymbol{\delta}(s)/2}\,\frac{(S_s)^n_{\mathbf{j}+\boldsymbol{\delta}(s)/2}}{(\delta x_s)^n_{\mathbf{j}}}\Delta_s\left[\theta_i\,(\tilde{u}_i)^{n+1}_{\mathbf{j}} + (1-\theta_i)\,(u_i)^n_{\mathbf{j}}\right]\right\} \\
&= \delta_{iz}\left\{\frac{\Delta t}{\rho_0}\frac{\Delta_i\,(p_a)^{n^*}_{\mathbf{j}}}{(\delta x_i)^n_{\mathbf{j}}} + \frac{g\Delta t}{(\delta x_i)^n_{\mathbf{j}}}\Delta_i P_{\mathbf{j}} + g\Delta t\Delta_i\left[\theta_i\tilde{\eta}^{n+1}_{\mathbf{j}} + (1-\theta_i)\,\eta^n_{\mathbf{j}}\right]\right\} \\
&+\frac{\Delta t}{(\delta x_i)^n_{\mathbf{j}}}\,(1-\theta_i)\,\Delta_i q^n_{\mathbf{j}}
\end{aligned}
\tag{3.1}
$$

Also, the free surface equation is:

$$
\begin{aligned}
0 = &\left(\tilde{\eta}^{n+1}_{\mathbf{j}} - \eta^n_{\mathbf{j}}\right)(S_z)^n_{\mathbf{j}} \\
&+\Delta t\Delta_x\left(\theta_x\,(\tilde{u}_x S_x)^{n+1}_{\mathbf{j}} + (1-\theta_x)\,(u_x S_x)^n_{\mathbf{j}}\right) \\
&+\Delta t\Delta_y\left(\theta_y\,(\tilde{u}_y S_y)^{n+1}_{\mathbf{j}} + (1-\theta_y)\,(u_y S_y)^n_{\mathbf{j}}\right) \\
&+\Delta t\left(\theta_z\,(\tilde{u}_z S_z)^{n+1}_{\mathbf{j}} + (1-\theta_z)\,(u_z S_z)^n_{\mathbf{j}}\right)
\end{aligned}
\tag{3.2}
$$

In this system, we have $3\,(N_j+1)\,(N_k+1)\,(N_l+1) + N_j N_l$ unknowns. The momentum equation applies to each interior cell.

This means we have $3\,(N_j-1)\,(N_k-1)\,(N_l-1)$ equations. Then we have the free-surface equation that applies to each surface cell ($N_j N_l$ equations). The remain are given by boundary conditions. Boundary conditions for velocity are given at cell faces (for u velocity,$2N_k N_l$, for v,$2N_j N_l$, for w, $2N_j N_k$). This closes the linear system.

The calculation molecule in the s direction draws as in the figure 3.1.

As s ranges from 1 to 3, we have that for each equation, the number of unknowns involved is 9 for velocity field and 5 for water level (indeed, in the vertical direction the only unknowns are related to velocity field). We end up with a sparse linear system that contains 14 unknowns in each line. We can solve this system in different ways, but a suitable approach used for a long time with

success by MOHID, is to make a time splitting in the two horizontal directions that reduces a lot the complexity of the system. This is explained in the next chapter. Consider for the moment that we have successfully solved the system retrieving the provisional values of u_i and η.

3.3 Second Step: Non-Hydrostatic Correction

3.3.1 Equation for the Interior of the Domain

In the second step of calculations, the new velocity field and the new water elevations (respectively u_i and η) are computed by correcting the provisional values after including the non-hydrostatic terms.

$$(u_i)_{\mathbf{j}}^{n+1} = (\tilde{u}_i)_{\mathbf{j}}^{n+1} - \theta_i \frac{\Delta t}{(\delta x_i)_{\mathbf{j}}^n} \Delta_i \tilde{q}_{\mathbf{j}}^{n+1} \tag{3.3}$$

The meaning of this provisional pressure correction is the correction that has to be made to the provisional hydrostatic pressure to get the total pressure, it is:

$$p_{\mathbf{j}}^{n+1} = g\left(\tilde{\eta}_{\mathbf{j}}^{n+1} - z_{\mathbf{j}}\right) + \tilde{q}_{\mathbf{j}}^{n+1} \tag{3.4}$$

There is actually no physical meaning to this quantity.
The fluxes can be expressed as:

$$(F_i)_{\mathbf{j}}^{n+1} = \left(\tilde{F}_i\right)_{\mathbf{j}}^{n+1} - \theta_i \Delta t \frac{(S_i)_{\mathbf{j}}^n}{(\delta x_i)_{\mathbf{j}}^n} \Delta_i \tilde{q}_{\mathbf{j}}^n \tag{3.5}$$

If we substitute this last relationship in 2.23, we have that.

$$\sum_{i=1}^{3} \nabla_i \left[\theta_i \left(\tilde{F}_i\right)_{\mathbf{j}}^{n+1} - \theta_i^2 \Delta t \frac{(S_i)_{\mathbf{j}}^n}{(\delta x_i)_{\mathbf{j}}^n} \Delta_i \tilde{q}_{\mathbf{j}}^n + (1-\theta_i)(F_i)_{\mathbf{j}}^n\right] = 0 \tag{3.6}$$

Hence, developing,

$$\begin{aligned} & \sum_{i=1}^{3} -(a_i)_{\mathbf{j}+\boldsymbol{\delta}(i)}^n \tilde{q}_{\mathbf{j}+\boldsymbol{\delta}(i)}^n + \left[(a_i)_{\mathbf{j}+\boldsymbol{\delta}(i)}^n + (a_i)_{\mathbf{j}}^n\right] \tilde{q}_{\mathbf{j}}^n - (a_i)_{\mathbf{j}}^n \tilde{q}_{\mathbf{j}-\boldsymbol{\delta}(i)}^n \\ = & \sum_{i=1}^{3} \nabla_i \left[\theta_i \left(\tilde{F}_i\right)_{\mathbf{j}}^{n+1} + (1-\theta_i)(F_i)_{\mathbf{j}}^n\right] \end{aligned} \tag{3.7}$$

Where the coefficients are given by:

$$(a_i)_{\mathbf{j}}^n = \theta_i^2 \Delta t \frac{(S_i)_{\mathbf{j}}^n}{(\delta x_i)_{\mathbf{j}}^n} \tag{3.8}$$

3.3.2 Equation for the Free-Surface

At the upper cell the above does not apply and we need another equation that is very similar.

$$\begin{aligned} 0 &= \left(\eta_{\mathbf{l}}^{n+1} - \eta_{\mathbf{l}}^{n}\right)(S_z)_{\mathbf{l}}^{n} \\ &+\Delta t \nabla_x \left[\theta_x (F_x)_{\mathbf{l}}^{n+1} + (1-\theta_x)(F_x)_{\mathbf{l}}^{n}\right] \\ &+\Delta t \nabla_y \left[\theta_y (F_y)_{\mathbf{l}}^{n+1} + (1-\theta_y)(F_y)_{\mathbf{l}}^{n}\right] \\ &+\Delta t \left[\theta_z (F_z)_{\mathbf{l}}^{n+1} + (1-\theta_z)(F_z)_{\mathbf{l}}^{n}\right] \end{aligned} \tag{3.9}$$

Where $\mathbf{l}$ denotes the upper cell index:

$$\mathbf{l} = (j, k, N_l) \tag{3.10}$$

Supposing that the pressure at the upper cell is hydrostatic, we have that:

$$p_{\mathbf{l}}^{n+1} = g\left(\eta_{\mathbf{l}}^{n+1} - z_{\mathbf{l}}\right) = g\left(\tilde{\eta}_{\mathbf{l}}^{n+1} - z_{\mathbf{l}}\right) + \tilde{q}_{\mathbf{l}}^{n+1} \tag{3.11}$$

This yields:

$$\eta_{\mathbf{l}}^{n+1} = \tilde{\eta}_{\mathbf{l}}^{n+1} + \frac{\tilde{q}_{\mathbf{l}}^{n+1}}{g} \tag{3.12}$$

Substituting 3.12 and 3.3 in 3.9 we finally get:

$$\begin{aligned} &\frac{(S_z)_{\mathbf{l}}^{n}}{g\Delta t}\tilde{q}_{\mathbf{l}}^{n+1} + (a_z)_{\mathbf{l}}^{n}\tilde{q}_{\mathbf{l}}^{n+1} - (a_z)_{\mathbf{l}}^{n} q_{\mathbf{l}-\boldsymbol{\delta}(z)}^{n+1} \\ &- (a_x)_{\mathbf{l}+\boldsymbol{\delta}(x)}^{n}\tilde{q}_{\mathbf{l}+\boldsymbol{\delta}(x)}^{n+1} + \left[(a_x)_{\mathbf{l}+\boldsymbol{\delta}(x)}^{n} + (a_x)_{\mathbf{l}}^{n}\right]\tilde{q}_{\mathbf{l}}^{n+1} - (a_x)_{\mathbf{l}}^{n} q_{\mathbf{l}-\boldsymbol{\delta}(x)}^{n+1} \\ &- (a_y)_{\mathbf{l}+\boldsymbol{\delta}(y)}^{n} q_{\mathbf{l}+\boldsymbol{\delta}(y)}^{n+1} + \left[(a_y)_{\mathbf{l}+\boldsymbol{\delta}(y)}^{n} + (a_y)_{\mathbf{l}}^{n}\right]\tilde{q}_{\mathbf{l}}^{n+1} - (a_y)_{\mathbf{l}}^{n} q_{\mathbf{l}-\boldsymbol{\delta}(y)}^{n+1} \\ &= \frac{(S_z)_{\mathbf{l}}^{n}}{\Delta t}\left(\eta_{\mathbf{l}}^{n} - \tilde{\eta}_{\mathbf{l}}^{n+1}\right) - \left[\theta_z \left(\tilde{F}_z\right)_{\mathbf{l}}^{n+1} + (1-\theta_z)(F_z)_{\mathbf{l}}^{n}\right] \\ &-\nabla_x \left[\theta_x \left(\tilde{F}_x\right)_{\mathbf{l}}^{n+1} + (1-\theta_x)(F_x)_{\mathbf{l}}^{n}\right] - \nabla_y \left[\theta_y \left(\tilde{F}_y\right)_{\mathbf{l}}^{n+1} + (1-\theta_y)(F_y)_{\mathbf{l}}^{n}\right] \end{aligned} \tag{3.13}$$

3.3.3 Boundary Conditions

We still miss a set of equations delivered by boundary conditions. The boundary conditions will depend on those imposed for velocity.

If we have von Neumann boundary conditions for velocity, water level is imposed and the above equation drops for boundary cells.

Whereas we have a Dirichlet boundary condition, we need a flux condition for water level, hence pressure correction. We will impose null flux as in [5] :

$$\boldsymbol{\nabla} q \cdot \mathbf{n} = 0 \tag{3.14}$$

This implies that also:

$$\boldsymbol{\nabla}\tilde{q}\cdot\mathbf{n}=0 \tag{3.15}$$

This yields a very simple difference equation, once the faces are aligned with the coordinate axes. It can be one of the two difference equations bellow, depending on which boundary is being treated.

$$\begin{aligned}\nabla_s\tilde{q}_{\mathbf{j}}^{n+1} &= 0 &(3.16)\\ \Delta_s\tilde{q}_{\mathbf{j}}^{n+1} &= 0 &(3.17)\end{aligned}$$

3.3.4 Final Correction

The above linear heptadiagonal system needs to be solved. Once we have the field for provisional pressure correction we apply it to the velocities.

$$(u_i)_{\mathbf{j}}^{n+1}=(\tilde{u}_i)_{\mathbf{j}}^{n+1}-\theta_i\frac{\Delta t}{(\delta x_i)_{\mathbf{j}}^{n}}\Delta_i\tilde{q}_{\mathbf{j}}^{n+1}\quad\text{only for } i=1,2 \tag{3.18}$$

In most of our application mass conservation is a critical issue. It is therefore capital to keep our field conservative. So to derive the vertical velocity field by applying the continuity equation instead of the correction equation above.

Having made the corrections, we have that:

$$\begin{aligned}\theta_z\left(u_z\right)_{\mathbf{j}+\boldsymbol{\delta}(z)}^{n+1} &= \left(u_z\right)_{\mathbf{j}}^{n+1}-\left(1-\theta_z\right)\nabla_z\left(u_z\right)_{\mathbf{j}}^{n}\\ &\quad-\nabla_x\left[\theta_x\left(F_x\right)_{\mathbf{j}}^{n+1}+\left(1-\theta_x\right)\left(F_x\right)_{\mathbf{j}}^{n}\right]/\left(S_z\right)_{\mathbf{j}}^{n}\\ &\quad-\nabla_y\left[\theta_y\left(F_y\right)_{\mathbf{j}}^{n+1}+\left(1-\theta_y\right)\left(F_y\right)_{\mathbf{j}}^{n}\right]/\left(S_z\right)_{\mathbf{j}}^{n} \qquad (3.19)\end{aligned}$$

Remembering that at the bottom we have the non-slip condition:

$$\left(u_z\right)_{1}^{n+1}=0 \tag{3.20}$$

We can easily go up the water column from bottom to top and retrieve the values of vertical velocities.

We need also to correct water level accordingly:

$$\eta_{\mathbf{l}}^{n+1}=\tilde{\eta}_{\mathbf{l}}^{n+1}+\frac{\tilde{q}_{\mathbf{l}}^{n+1}}{g} \tag{3.21}$$

And, finally, the pressure correction is:

$$q_{\mathbf{j}}^{n+1}=\tilde{q}_{\mathbf{j}}^{n+1}-\tilde{q}_{\mathbf{l}}^{n+1} \tag{3.22}$$

This implies of course that the pressure correction at the bottom is zero, but that was expected, indeed, we assumed that the pressure distribution was hydrostatic in the upper cell.

Chapter 4

Solution to the System

"O rage! Ô désespoir! Ô vieillesse ennemie! N'ai-je donc tant
vécu que pour cette infamie?" Don Dièguе, in *Le Cid*, **P. CORNEILLE**

Somewhere above we mentioned that any appropriate solver would solve the linear system we developed by discretizing our equations. The ideal solver should be fast because the system has to be solved at each time step, which means thousands of times in an usual simulation. We did not reached this point of the business to see our theory crumble and fall apart. Or as would say Don Dièguе, have we worked so long to contemplate this infamy? We have to solve two distinct systems. The one retrieved from the hydrostatic hypothesis (step one), and the other for the pressure correction (step two).

In the first case, we will decompose our system in various tri-diagonal systems, so we can use the simplest solver for diagonal systems: the Thomas algorithm. It is a $O(N)$ solver only valid for tri-diagonal systems. This is the reason why MOHID does not allow for high discretization schemes in time: the resulting algebric system would not be tri-diagonal and it would not be possible to use such a fast solver.

In the former case, this decomposition is not feasible, leaving us with one unpleasant situation: finding the appropriate solver. A look at the literature indicates us the appropriate solver: the Bi-Conjugate Gradient Stabilized Algorithm (Bi-CGSTAB). This algorithm is commonly used and widespread for solving sparse linear system ([18]).

4.1 Time-Splitting

The Leendertsee scheme is a numerical decomposition of our system that eases its solving. It is processed in six successive steps. We decompose the solving into the two horizontal direction, and solve the free-surface equation on its own.

We start with the known fields $\eta^n, u_x^n, u_y^n, u_z^n$, adding at each step a new field that is used afterwards.

As we can see this is a this procedure is executed in two time steps. But we can without any difficulty turn it into a two half time steps scheme. That is what is done usually. The semi-implicit coefficients are crucial for stability. We tried here to reproduce the stability conditions of the algorithm described

Equation	Unknown	θ_x	θ_y	θ_z
Free-Surface	η^{n+1}	1	0	0
X-Momentum	u_x^{n+1}	1	0	0
Pressure Correction	q^{n+1}	1	0	1/2
Continuity	u_z^{n+1}	1	0	1/2
Free-Surface	η^{n+2}	0	1	0
Y-Momentum	u_y^{n+2}	0	1	0
Pressure Correction	q^{n+2}	0	1	1/2
Continuity	u_z^{n+2}	0	1	1/2

Table 4.1: Leendertsee Scheme

in [6]. According to the same source, an overall value of θ comprised between 0.5 and 1 should provide the conditions for stability in the von Neumann sense. This is why in the course of the above scheme the semi-implicit coefficients for vertical velocity is 1/2, so at the end we have an overall coefficient of 1. But this is only an intuitive approach, we are free to take any value between 0.5 and 1 (as application of the model show).

With this time-splitting scheme it is possible to reduce our gargantuan system of equations to a tri-diagonal system at each step. And tri-diagonal systems are easily solved with the Thomas algorithm.

This time-splitting is of course a filthy way to avoid solving a huge system of equation. But since it has been integrated in MOHID in its early beginnings with glittery success, the author of these lines is fairly inclined in letting the voice of empirical knowledge speak.

We will not be so lucky however when trying to solve the equations for the pressure correction q. The equations for the pressure correction are introduced just before the continuity equation solving. As we see, at the moment, we have two semi-implicit coefficients that are not zero. This means the system we need to solve in not tri-diagonal but penta-diagonal. In consequence we need another algorithm in order to solve that penta-diagonal system: the Bi-Conjugate Gradient Stabilized algorithm.

4.2 Solution to Sparse Linear Systems

4.2.1 The Thomas Algorithm

The Thomas Algorithm is deduced in appendix A. It is very useful for solving tri-diagonal linear systems. It has a computational expense that is only $O(N)$ where N is the number of equations. By comparison, the classical Gauss method for matrix inversion is $O(N^2)$. See appendix A for a complete description of the Thomas algorithm.

4.2.2 Iterative Solver

> "When I decided to learn the Conjugate Gradient Method (henceforth, CG), I read four different descriptions, which I shall politely not identify. I understood none of them." **J.R. SHEWCHUK** [20]

To begin with[1], consider the system

$$A\mathbf{x} = B \tag{4.1}$$

The basic idea behind iterative solvers is to split the matrix A in a sum of two matrices, M and N, so we have.

$$M\mathbf{x} = B - N\mathbf{x} \tag{4.2}$$

Based on this last expression we can mount an iterative method:

$$\mathbf{x}_{i+1} = M^{-1}B - M^{-1}N\mathbf{x}_i \tag{4.3}$$

For an iterative method to be effective, either it must converge in few iterations, either iterations are inexpensive operations. The second is usually easier to achieve. Inexpensive iterations require that the product $N\mathbf{x}_i$ and the inverse of the matrix M is easy to find.

Classical iterative solvers include the Jacobi method and the Gauss-Seidel method. A less classical approach but still in the same line is the incomplete LU decomposition, or Stone's method (a description of this method and others can be found in [18]). Those methods are sensitive to the initial guess. They converge very fast near the solution but may fail far from the solution.

A very popular group of methods that uses another philosophy, is derivated from the Conjugate Gradient Method.

Let the residual be defined as:

$$\mathbf{r}_i = A\mathbf{x}_i - B \tag{4.4}$$

The idea of this method is to minimize the residual or other derived quantity that ensures that the minimization process leads to the solution of the system - as the quadratic form in 4.5.

$$f(\mathbf{x}) = \frac{1}{2}\left\|A\mathbf{x} - B\right\|^2 \tag{4.5}$$

A gradient method as Cauchy's steepest descent method can then be used to minimize this quantity.

Usually the CG based methods are coupled with classical iterative schemes. In this case the method is said to be preconditioned. Anyway, for deep and clear insight, see reference [20].

Based on the method of the Conjugate Gradients, there is a series of very efficient algorithm that are vastly used to solve linear systems. One of them is the Bi-Conjugate Gradients Stabilized algorithm (Bi-CGSTAB) for which an explicit description is given in reference [21]. In our model we used Stone's method for pre-conditioning with success.

[1] The subject is difficult, and, as mentioned above, most of the literature about it is rather obscure. The following is just a very superficial overview and unfortunately does not provide any help in clarifying this matter. See for instance [20] for detailled explanation of CG methods.

4.2.3 Convergence Criteria

One problem that arises when using iterative solvers is to know when to stop. One common practice is based on the difference between two successive iterates; the procedure is stopped when this difference, measured by some norm, is less than a pre-selected value. Unfortunately, this difference may be small when the absolute value of the error is small and a proper normalization is essential. An adequate estimation of the error can be found in [18]:

$$\varepsilon^i = \frac{\left\|\boldsymbol{\delta}^i\right\|}{\sqrt{\ell^2 + 1}} \tag{4.6}$$

Where we have defined:

$$\boldsymbol{\delta}^i = \mathbf{x}_i - \mathbf{x}_{i-1} \tag{4.7}$$

Where ℓ is an estimation of the magnitude of the largest eigenvalue of the matrix of our system A.

$$\ell = \sqrt{\frac{z^i}{z^{i-1}}} \tag{4.8}$$

And

$$z^i = \boldsymbol{\delta}^{i-2} \cdot \boldsymbol{\delta}^i - \boldsymbol{\delta}^{i-1} \cdot \boldsymbol{\delta}^{i-1} \tag{4.9}$$

Due to the oscillations of the solutions, the estimate may not be accurate on any particular iteration, but, quite good on the average. In order to remove the effects of the oscillation, the eigenvalue estimates should be averaged over a range of iterations. Depending on the problem and the number of anticipated iterations, the averaging range may vary from 1 to 50 (typically 1% of the expected number of iterations) [18].

Chapter 5

Application to Schematic Cases

5.1 Free-Surface Oscillation in a Water Basin

To begin with, we will apply the model to two dimensional domains, to see if, indeed, the non-hydrostatic properties of a flow are caught. First we will study an oscillating basin. The free-surface is initially at rest with a constant slope. Then the free-surface is left oscillating freely. The expected solution is a standing wave of wavelength, $\lambda = 2L$, where L is the basin length. Neglecting diffusion, the celerity of the wave can be approximated by:

$$c = \sqrt{\frac{g\lambda}{2\pi} \tanh\left(\frac{2\pi H}{\lambda}\right)} \tag{5.1}$$

Where L is the wavelength, in this case the length of the basin and H the oscillation amplitude.

The square water basin has a length $L = 10\,\mathrm{m}$ and a depth $h = 10\,\mathrm{m}$ and the slope $\eta = 0.02x - 0.1$. The simulations where carried in a 20x20 grid with $\Delta t = 0.01\,\mathrm{s}$. In figure 5.1 we show the solutions obtained with and without the non-hydrostatic correction. As this case was two-dimensional, the Leendertsee scheme ha been somewhat changed to a 3 steps method, where the Y-momentum equation is not considered. The value for θ_x is set to 1.

The convergence criteria for the non-hydrostatic system was set to 10^{-6}. With a Pentium IV, 2.4GHz, the time per iteration was $0,54\,\mathrm{s}$ for the hydrostatic case and $0,99\,\mathrm{s}$ for the non-hydrostatic case. This increment in computational expense was not unexpected (see reference [6]).

We can see that only the non-hydrostatic model is able to reproduce the wave with the right period and wavelength.

5.2 The Lock-Exchange Problem

After the first basic test was completed with some success, we tried a case that is considered as a severe test case to the model. The lock exchange problem is a classic in mechanics; it is used to show a natural instability that appears in shear

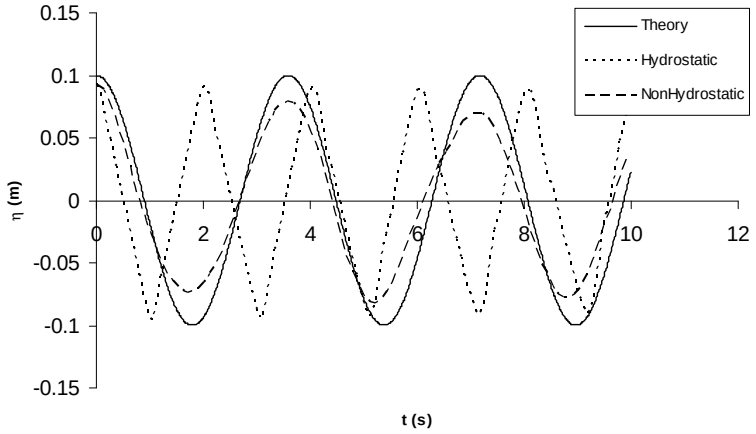


Figure 5.1: The oscillating basin: solutions comparison

Grid	N_x	N_z	Δx(m)	Δz(m)	Δt(s)	$\nu(\mathrm{m^2\,s^{-1}})$	*Advection*
Coarsen	100	20	0.025	0.015	0.01	0.0	*Upwind*
Refined	400	80	0.00625	0.00375	0.0025	2.0×10^{-4}	*CDS*

Table 5.1: Table of simulations for the lock exchange problem

layers: the Kelvin-Helmholtz instability. Kelvin-Helmoltz instabilities develop at the interface of fluids in motion with different properties (either physical or kinematic). We showed illustrations of this in figures 1 and 2. The lock exchange problem consists of a rectangular basin with two compartments which are filled with miscible fluids of different densities. At the beginning of the simulation, the separation between the two compartments is removed and fluids are left in free motion. The baroclinic unbalance initiates the movement and the interface slides, providing a mean to recover equilibrium.

For this analysis two meshes where considered. First a coarse mesh was built. The results on the first being promising, but by an excess of numerical diffusion, we decided to refine the mesh. Main characteristics of the two simulations are given in the table 5.1. In both cases a uniform, isotropic, turbulent eddy viscosity was considered. The water basin is 2 meters long and has a depth of 0.3 meters. To keep the first simulation simple, we used an upwind scheme for advection and no diffusion.

The first grid main objective is to show the inaptitude of the hydrostatic model in the prevision of such a flow. While the second grid serves to show that the model works well with high resolution, and that the diffusion problem encountered in the primary simulation can be solved.

From the figure 5.2 it is possible to conclude as expected that the hydrostatic model is inadequate for this simulation. While in the non-hydrostatic model the behaviour seems much more realistic. The difference is striking after five seconds: while the hydrostatic model gives a step shaped front, the new model give a smooth interface, much closer to the physical solution The initial Kelvin-

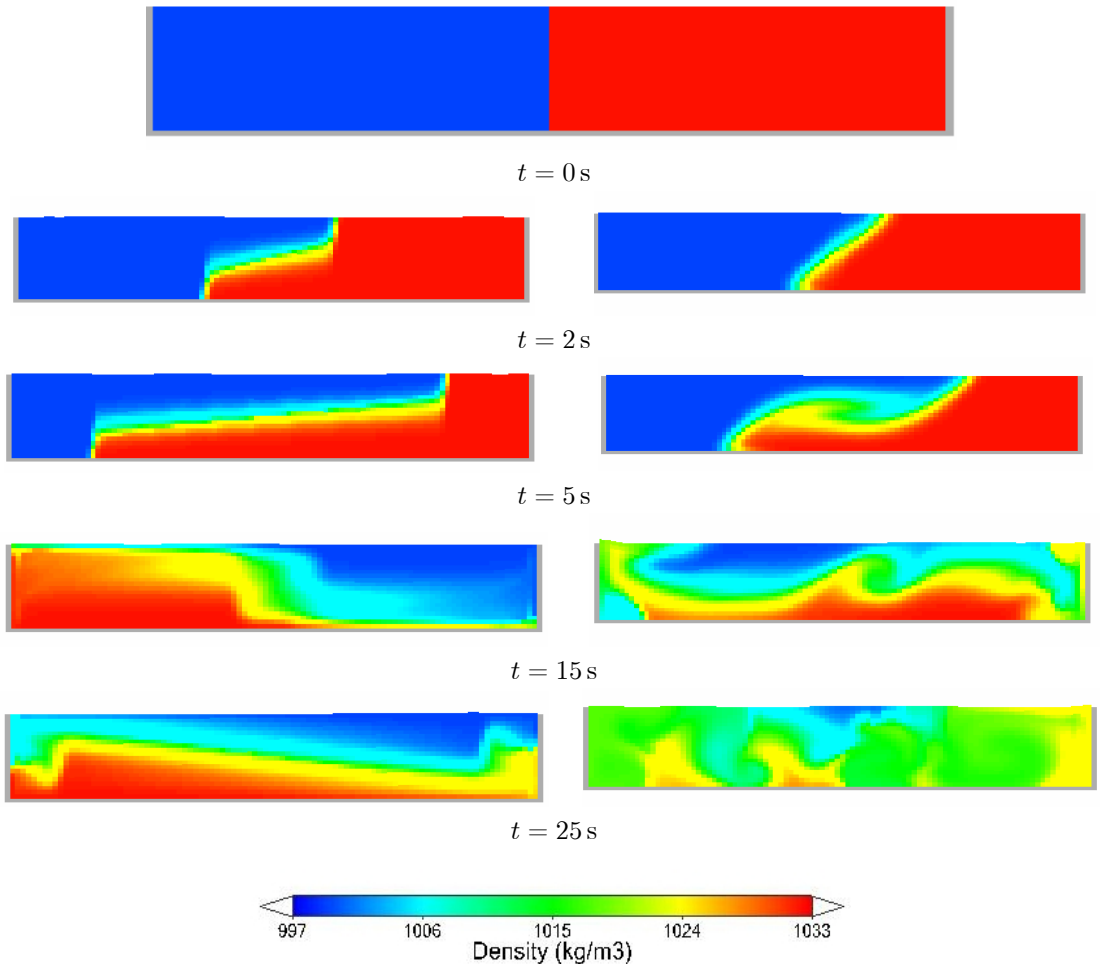


Figure 5.2: The lock exchange problem: comparison between non-hydrostatic and hydrostatic solutions

Helmholtz instability develops and is followed by the development of new ones. This goes on until the fluid is completely mixed (about 30 seconds after the beginning of the simulation). But as we will see later this quick mixing comes more from numerical diffusion than natural diffusion. And, at the end, after complete mixing, when we should have a resting fluid, because of the excess of diffusion, instabilities develop, leading to artificial vortices that get amplified with time. This last and unexpected result can be corrected if we refine the grid, adjust turbulent properties and adopt a less diffusive numerical scheme as seen before.

The second simulation, with a more refined grid, shows a much better and realistic behaviour. It is well known that upwind schemes are diffusive in nature. A way to avoid this is to change the advection scheme as we have done, from upwind to CDS. As a consequence, numerical diffusion drops and a more realistic result is achieved. In figure 5.3 we see that apparently the two liquids do not mix but rather slide over each other with some zones of interpenetration at the interface created by the various Kelvin-Helmholtz instabilities. At the end of the simulation, the fluid tends to the hydrostatic rest, as expected, forming two distinct layers of fluid, with the denser fluid at the bottom.

5.3 Wave Propagation Over a Bar

> "(...) Under the sea, is where I'll be
> No talking 'bout the rain no more
> I wonder what thunder will mean, when only in my dream
> The lightning comes before the roar
> Under the sea, down here with me I find I'm not the only one
> Who ponders what life would mean if we hadn't been
> So disappointed in the sun
> And that's why we're thinking
> That's why we're drinking in a bar under the sea (...)", in *Disappointed in the Sun* by Tom Barman

The hydrostatic delivers a unreasonable answer to this problem-following the conclusions of [6]-and our non-hydrostatic version seems not to be keen on keeping stable. Hundreds of simulations with different meshes and parametrization for turbulence have been realized without success. We will only expose the best results. It was in this context that appeared to us the necessity to develop a valid turbulence model. The homogeneous eddy viscosity model was not valid anymore, indeed nor the turbulent viscosity was homogeneous, neither it was isotropic. With this case, we are for the first time confronted to the problem of different length scales. This practical application consists in the simulation of a laboratory experiment carried out at Delft Hydraulics in a wave flume [3]. The flume geometry is given in the drawing (figure 5.4).

In theory, a monochromatic wave passing over the bar will shorten in length and will rise. Just after the bar, the wave suffers a division and a smaller wave propagates upstream, while main part of the energy goes through the shore, to finally break. Our model does not allow for wave breaking, but increasing artificially viscosity and roughness near the shore guaranties energy dissipation so we do not have reflection of waves.

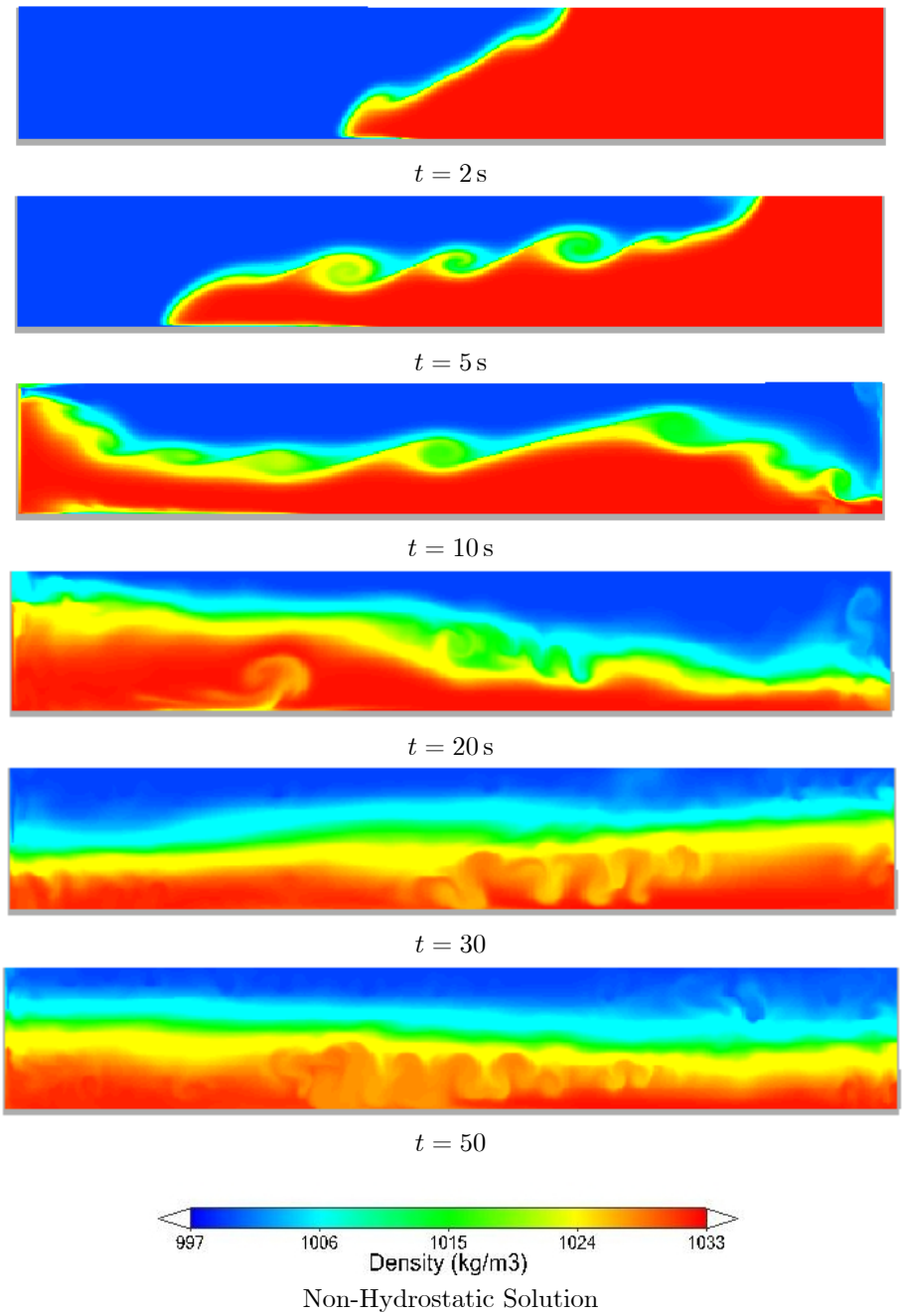


Figure 5.3: The lock exchange problem: comparison between non-hydrostatic and hydrostatic solutions

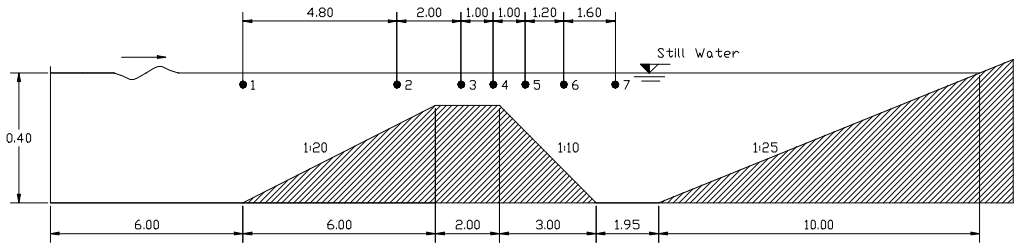


Figure 5.4: Longitudinal section of the wave flume: numbers 1 to 7 indicate wave gauge positions (as in [3])

The inlet wave has an amplitude of 1 cm and a period of 2 s. We present, as a reference, hydrostatic model results at various station (Figure **??**).

Should we conclude that we were *disappointed over a bar*? Indeed our model revealed here one of its flaws. Only when diffusion is turned off the problem disspaears. However with diffusion turned off, the results are unrealsitic. There are four main suspects:

- Courant number too high
- The assumption that pressure on the first–surface–cell is hydrostatic
- The sigma mesh deformation that does not take into account gain or loss in potential energy–as shown earlier
- The turbulence model

In the first case, the problem should be solved with a smaller time step. Experience showed this chage to be without effect.

The second possible suspect is eliminated if we get the same results while making a simulation we use a first cell considerably thiner than the others. This is precisely what happens.

The third problem should be solved if the initial grid is replaced by a cartesian grid. That is what we have done without success.

Of course, all these different possibilities were submitted to different parameters, as values for the residuals, turbulence models, etc., yielding no better results than before.

The turbulence model, used with success in other cases, can be at the origin of the flaw, because diffusion triggers the instabilities. Unfortunately, no information was found on turbulence parametrization for this case. It is the principal–and *usual*–suspect. This last hypothesis is confirmed by the fact that this problem vanishes with decreasing viscosity.

It is important, however, to stress that this does not prove the inability to simulate wind waves as the image (figure 5.6) shows. This image was obtained after 42 seconds of non-hydrostatic simulation. Although the results seem realistic they do not match the laboratory measurements.

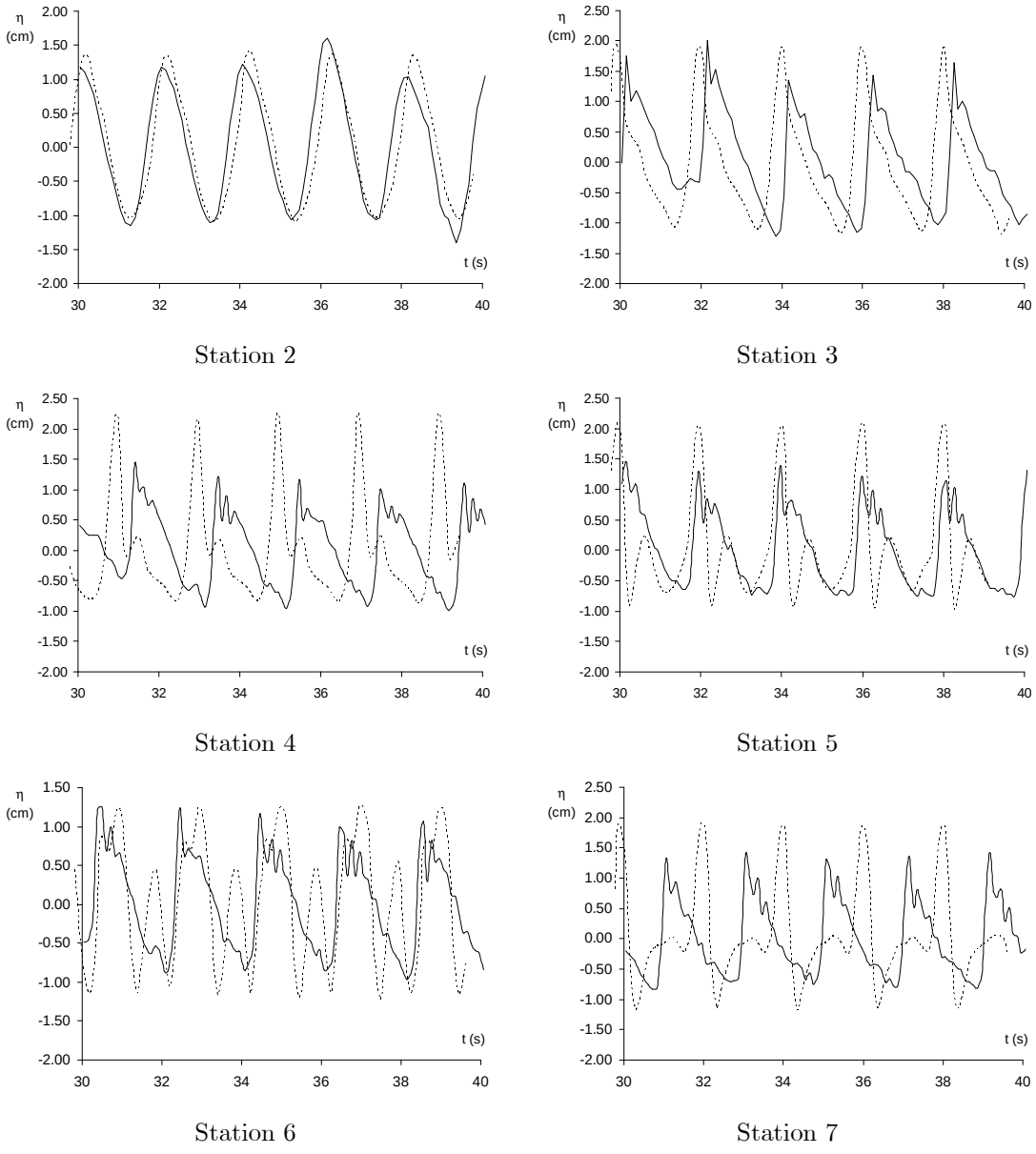


Figure 5.5: Wave over a bar: comparison of numerical solution (plain line) and measured data (dot line)

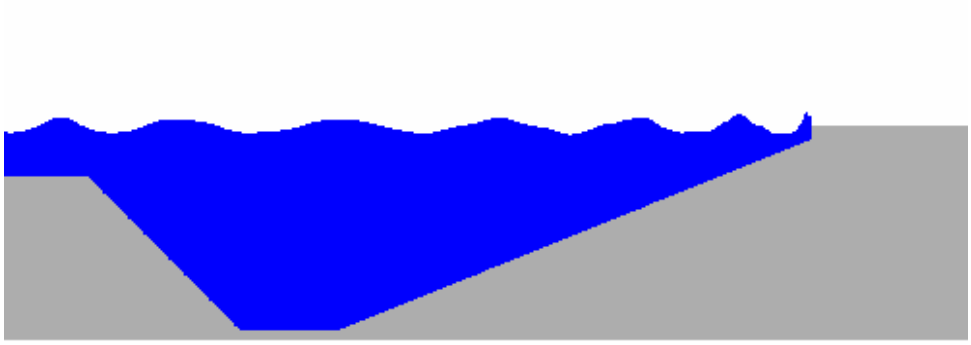


Figure 5.6: Wave over a bar propagation. A view of the non-hydrostatic surface elevation past the bar. Vertical coordinate is exagerated 10 times.

Grid	N_x	N_y	N_z	Δx(m)	Δy(m)	Δz(m)	Δt(s)
Sigma	200	20	50	0.01	0.01	$0.004 - 0.001$	0.0025

Table 5.2: Table of simulations for the solitary wave problem

5.4 Solitary Wave

This last case is used to show that the scheme works in 3D. Our model seems correct in its previsions despite the stability issues.

What we study here is the breaking of a solitary wave over a slope. In a closed basin we have two liquids of different densities the lighter being on the top. The basin is two meters long, and has a 0.2×0.2 square section. The slope is $1/10$.

We have initially a deformed Gaussian pycnocline that is compensated by water surface deformation. The internal wave Gaussian profile is given by:

$$\zeta(x) = \zeta_0 \mathrm{e}^{-\left(\frac{x}{\sigma}\right)^2} \tag{5.2}$$

The free-surface profile is:

$$\eta(x) = \eta_0 + \frac{\rho_2 - \rho_1}{\rho_2} \left(\zeta(x) - \zeta_0\right) \tag{5.3}$$

The resulting wave propagation is observed. A series of simulations have been tried but we only retain one of them although all of them were carried out with success (see table 5.2). This one seemed to be the most interesting case.

In figure 5.9, we see plainly the mechanism through which breaking induces dissipation. The solitary waves forms two longitudinal vortices that suffer a deformation when the wave reaches the slope. At wave breaking, these coherent structures then split in a multitude of small transversal rolls-that are also coherent structures-that drive the diffusion process. Results are quite noisy, but time did not allow to filter results-which could have been quite interesting. We clearly see next free surface vortical structures that should not be there. This is noise generated by the model. A better turbulence model would avoid such

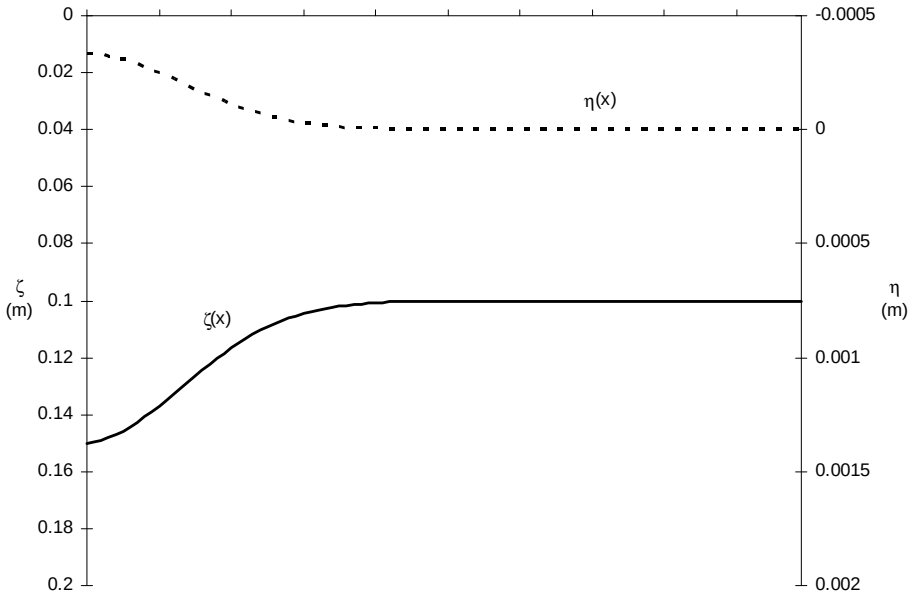


Figure 5.7: Solitary wave: initial free-surface and density interface profiles

problems. The wave movement is well reproduced, at least qualitatively. Displacement and height increase along the slope are well reproduced. There is no turn-over at the breaking (see [1] for a discussion on interfacial waves).

We can compare our wave to another case, similar to ours, with [4]. Density profile and intitial free-surface elevation are different, but an analogous vortical structure interaction can be depicted (see figure 5.10).

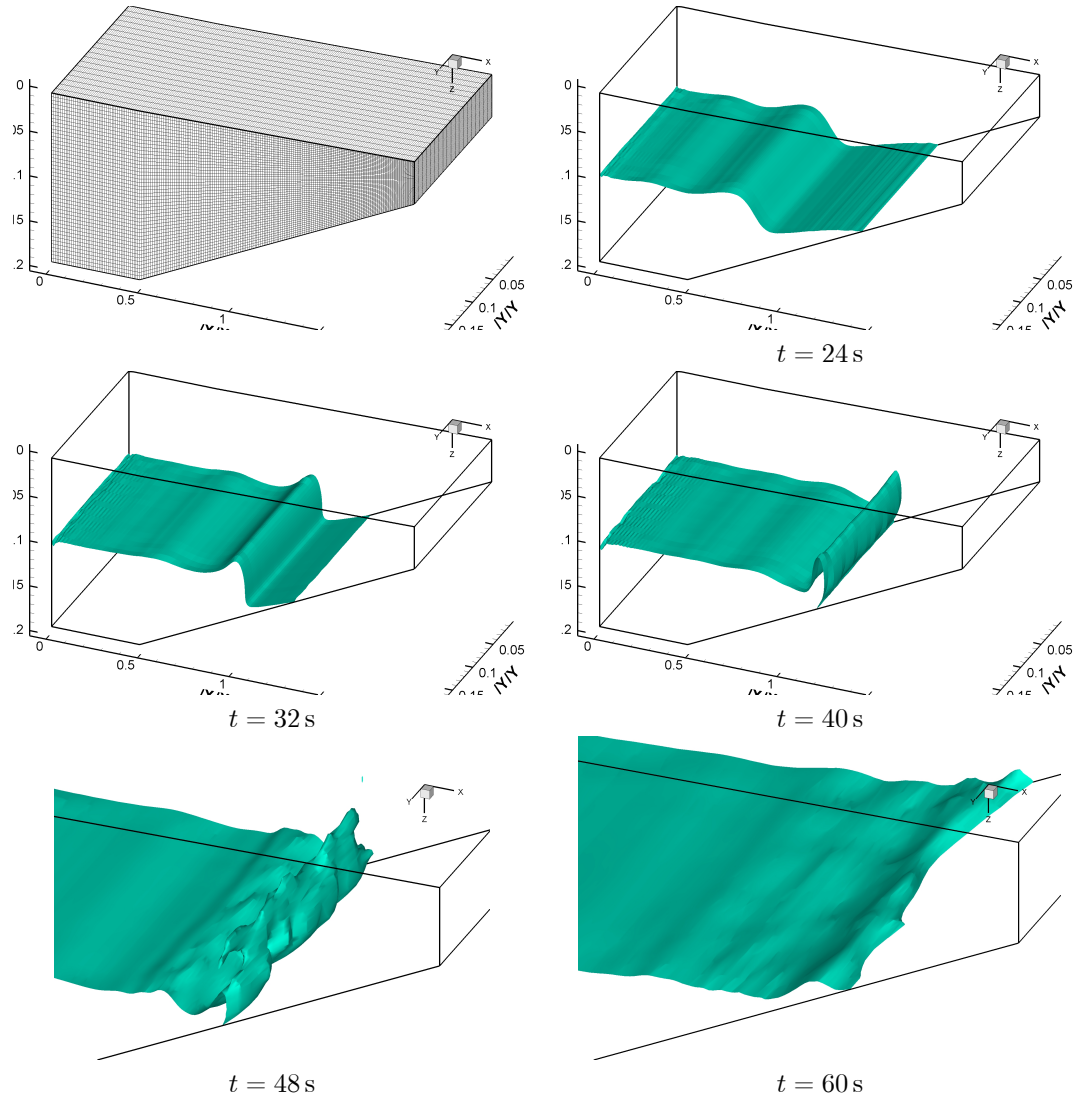


Figure 5.8: Grid used for simulations (upper left) and density contour surfaces of propagating solitary internal wave (density = 1029 kg/m$\hat{3}$)

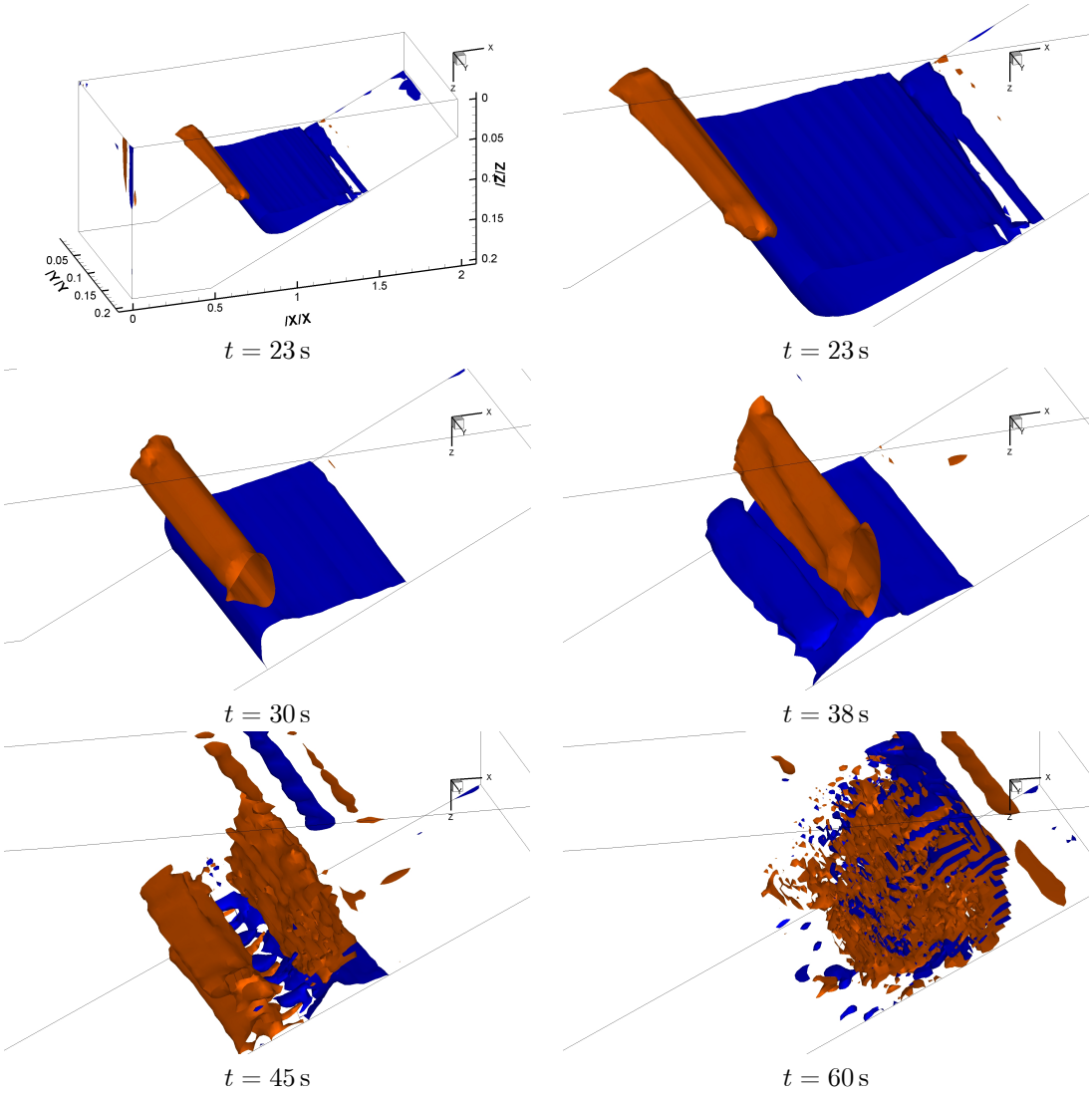


Figure 5.9: Longitudinal vorticity surface contours of a breaking solitary wave (red is +1 rad/s and blue -1 rad/s)

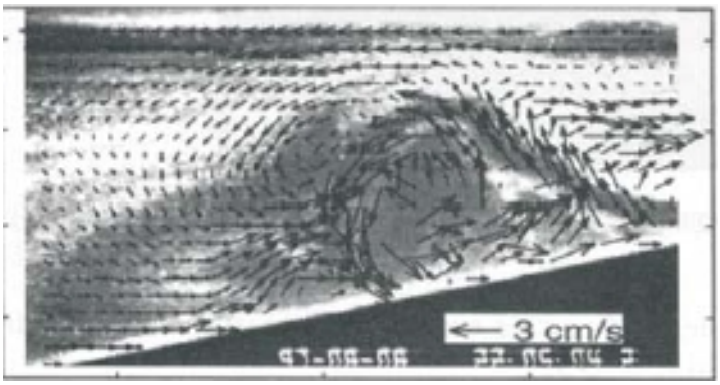


Figure 5.10: Laboratory observations with PIV for velocity field associated with the shallowing of an internal wave on a linear slope (from [4])

Chapter 6

Conclusions

From the previous results it is possible to conclude that:

1. The model has shown to be successful in most cases, showing realistic results either qualitatively and quantitatively. In the oscillating basin case the model proves being accurate quantitatively in the simulation of a standing wave. In the lock exchange problem, the model develops as expected the Kevin-Helmholtz instabilities. The case of the breaking internal wave is also realistic from a qualitative point of view. The most controversial case is, undoubtedly the propagation of a surface wave. The relatively bad results obtained may have their origin in the limitations of MOHID itself which was never thought to be used for short period surface waves (that implies high wave slopes).

2. The hydrostatic model fails obviously in all the above examples. It also shows to be much less dissipative. The case of the oscillating basin speaks by itself. The resulting standing wave shows a movement with inferior period, like if the overall mass of fluid was more 'rigid'. Also, in the lock exchange problem the fact that interfacial inclusion cannot appear prevents the mixing between the two layers of fluid and also the consequent transfer of energy that would take the fluid to rest sooner.

3. The 3D example has showed that the non-hydrostatic correction has been well integrated to the time splitting scheme. It has depicted well the structures responsible for diffusion.

4. The time of computation is estimated as being from two to six times the time required for the hydrostatic computation. However not only the new model is more restrictive as far as time step is concerned, but also, the time of computation depends greatly on the convergence criteria (residual value). Low values for residuals avoids the appearance of spurious modes, but can easily lead to unbearable computational cost. The problem resides in the fact that in certain circumstances MOHID seems to generate a lot of noise.

5. Compared to an usual hydrostatic simulation with MOHID, the overhead is important, but this is due to the fact that MOHID uses a non-iterative solver. If MOHID had to deal with more complex discretization schemes

it would need to develop an iterative solver, in which case, the pressure correction step would represent a small increment in computational cost.

6. The model shows a flaw that seems to be linked to turbulence modeling. This raises a stability issue but does not shows that it is not suitable to simulate wind waves. This topic should be investigated further.

7. The limits of this model are not very well known, but experience suggests values of $\theta_z = 1$ for the 2D cases and $\theta_z = 1/2$ for 3D case. Also, global residuals should be kept under 10^{-6}.

8. As a suggestion for further work, comes of course, the application of this model to a practical case. But this cannot be done before the model can be run in parallel (which is not the case for the moment). Indeed it is far too expensive to be run in a single machine. There are also some important work that needs to be done to see how this model behaves if the flow is hydrostatic. As wind waves generate stability problems that diffusion seem to amplify, it would be interesting to quantify numerical diffusion and trying to find a way to get rid of it, but also to know how the diffusion interferes with stability.

Appendix A: The Thomas Algorithm

In a tri-diagonal system, each equation can be written as (imagine a one-dimensional case):

$$a_i^S x_{i-1} + a_i^P x_i + a_i^N x_{i+1} = b_i$$

At each location, the solution can be written with the recurrence relation:

$$x_i = w_i \phi_{i+1} + g_i$$

We have also,

$$x_{i-1} = w_{i-1} x_i + g_{i-1}$$

Substituting this in the system,

$$a_i^S \left(w_{i-1} x_i + g_{i-1} \right) + a_i^P x_i + a_i^N x_{i+1} = b_i$$

And

$$x_i = \underbrace{-\frac{a_i^N}{a_i^P + a_i^S w_{i-1}}}_{w_i} x_{i+1} + \underbrace{-\frac{b_i - a_i^S g_{i-1}}{a_i^P + w_{i-1}}}_{g_i}$$

From the last equation we retrieve a recurrence formula for w_i and g_i:

$$\begin{aligned} w_i &= -\frac{a_i^N}{a_i^P + a_i^S w_{i-1}} \\ g_i &= -\frac{b_i - a_i^S g_{i-1}}{a_i^P + w_{i-1}} \end{aligned}$$

At the upper boundary, we have a condition that is (without loss of generality):

$$a_0^P x_0 + a_0^N x_1 = b_0$$

So, we have directly the first values for w_0 and g_0:

$$\begin{aligned} w_0 &= -\frac{a_0^N}{a_0^P} \\ g_0 &= \frac{b_0}{a_0^P} \end{aligned}$$

Applying the recursive relation we have the values for w_i and g_i from 0 to M. This constitutes the first step of the Thomas algorithm. We can apply the last values to get x_M. Indeed, the downstream boundary condition is such that a_M^N is zero.

$$a_M^S x_{M-1} + \phi_M^P x_M = b_M$$

Hence,

$$x_M = g_M$$

Using the recurrence relation we can retrieve all the values for x from $M-1$ to 0. It is the second step (backward substitution).

Appendix B: The Bi-CGSTAB Algorithm

We only give the algorithm that was implemented in MOHID. This algorithm has been presented in [21] and we transcribe it here to avoid any confusion, as there are several algorithm exposed in the article.

Let $\mathbf{x}_0$ be an initial guess of the system $A\mathbf{x} = \mathbf{b}$ and $\mathbf{r}_0 = A\mathbf{x}_0 - \mathbf{b}$. Let $\hat{\mathbf{r}}_0$ be an arbitrary vector, such that $\langle \mathbf{r}_0, \hat{\mathbf{r}}_0 \rangle \neq 0$ (usually, $\hat{\mathbf{r}}_0 = \mathbf{r}_0$).

Let the scalars, $\rho = \alpha = \omega = 0$ and the vectors, $\mathbf{v} = \mathbf{p} = \mathbf{0}$. M is a pre-conditioning matrix that should be close to A but that give a system easy to solve. In our case we toke the LU matrix of Stone's algorithm (with success).

The algorithm then writes:

for $i = 0, 1, 2, ..$

$\rho' = -\omega\rho$;

$\rho = \langle \mathbf{r}_i, \hat{\mathbf{r}}_0 \rangle$; $\beta = \alpha\rho/\rho'$;

$\mathbf{p} = \mathbf{r}_i - \beta\,(\mathbf{p} - \omega\mathbf{v})$;

Solve $\hat{\mathbf{p}}$ from the system $M\hat{\mathbf{p}} = \mathbf{p}$;

$\mathbf{v} = A\hat{\mathbf{p}}$;

$\mathbf{s} = \mathbf{r}_i - \alpha\mathbf{v}$;

$\gamma = \langle \mathbf{v}, \hat{\mathbf{r}}_0 \rangle$; $\alpha = \rho/\gamma$;

Solve $\hat{\mathbf{s}}$ from the system $M\hat{\mathbf{s}} = \mathbf{s}$;

$\mathbf{t} = A\hat{\mathbf{s}}$;

$\omega = \langle \mathbf{s}, \mathbf{t} \rangle \,/\, \langle \mathbf{t}, \mathbf{t} \rangle$;

$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha\hat{\mathbf{p}} + \omega\hat{\mathbf{s}}$;

$\mathbf{r}_{i+1} = \mathbf{s} - \omega\mathbf{t}$;

if $\mathbf{x}_{i+1}$ is accurate enough, then quit;

endfor

Appendix C: Stone's Method

The Stone's method is not so often mentioned in literature so we decided to expose it here. Stone's method is also called incomplete LU decomposition. We know that the LU decomposition is a good-general purpose solver but does not take advantage of the a system sparseness. Let $A = M - N$ where we decompose $M = LU$.

The method is here applied to the case that interests us: a pentadiagonal system.

$$a_i^W x_{i-d} + a_i^S x_{i-1} + a_i^P x_i + a_i^N x_{i+1} + a_i^E x_{i+d} = b_i$$

The ILU (incomplete LU) decomposition proceeds as in LU decomposition but the, for every element of the original matrix A that is zero„ the corresponding element of L or U is set to zero. This decomposition is not exact but nothing prevents us from using M it as the matrix for the iterative method of the Bi-CGSTAB algorithm. The elements of L and U matrices must be calculated only once:

The coefficients must be calculated in this order. Out of boundaries u and l elements are considered equal to zero. In the Bi-CGSTAB algorithm we have for instance that $M\hat{\mathbf{s}} = \mathbf{s}$ must be solved.

The inversion of M is quite straightforward. We only give here the formulas.

The above cited system can also be written (with the ILU decomposition):

$$U\hat{\mathbf{s}} = L^{-1}\mathbf{s} = \mathbf{r}$$

$\mathbf{r}$ is easily computed:

$$r_i = \left(s_i - l_i^S s_{i-1} - l_i^W s_{i-d}\right)/l_i^P$$

The solution is then:

$$\hat{s}_i = r_i - u_i^N r_{i+1} - u_i^E r_{i+d}$$

$$\begin{array}{l} l_i^W = l_{i,i-d} = a_i^W / \left(1 + \alpha u_{i-d}^N\right) \\ l_i^S = l_{i,i-1} = a_i^S / \left(1 + \alpha u_{i-1}^E\right) \\ l_i^P = l_{i,i} = a_i^P + \alpha\left(l_i^W u_{i-d}^N + l_i^S u_{i-1}^E\right) - l_i^W u_{i-d}^E + l_i^S u_{i-1}^N \\ u_i^N = u_{i,i+1} = \left(a_i^N - \alpha l_i^W u_{i-d}^N\right)/l_i^P \\ u_i^E = u_{i,i+d} = \left(a_i^E - \alpha l_i^S u_{i-1}^E\right)/l_i^P \end{array}$$

Bibliography

[1] O. B. Fringer and R. L. Street, "The dynamics of breaking progressive interfacial waves," *Journal of Fluid Mechanics* **494**, pp. 319–353, 2003.

[2] E. Simiu and R. H. Scalan, *Wind Effects on Structures*, Wiley Inter-science, 1996.

[3] S. Beji and J. A. Battjes, "Numerical simulation of nonlinear wave propagation over a bar," *Coastal Engineering* **23(1)**, pp. 1–16, 1994.

[4] D. Bourgault and D. E. Kelly, "A laterally-averaged nonhydrostatic ocean model," In preparation.

[5] J. Marshall, C. Hill, L. Perelman, and A. Adcroft, "Hydrostatic quasi-hydrostatic and nonhydrostatic ocean modeling," *Journal of Geophysical Research* **102**, pp. 5733–5752, 1997.

[6] V. Casulli and P. Zanolli, "Semi-implicit numerical modeling of non-hydrostatic free-surface flows for environmental problems," *Mathematical and Computer Modelling* **36**, pp. 1131–1149, 2002.

[7] R. Brugge, H. L. Jones, and J. C. Marshall, "Non-hydrostatic ocean modeling for studies of open-ocean deep convection," in *Deep Convection and Deep Water Formation in the Oceans*, **57**, pp. 325–340, Elsevier Oceanography Series, New York, 1991.

[8] M. Leschziner and W. Rodi, "Calculation of strongly curved open channel flows," *Journal of the Hydraulics Division* **105(HY10)**, pp. 1297–1314, 1979.

[9] V. Casulli, "A semi-implicit finite difference method for non-hydrostatic, free-surface flows," *International Journal for Numerical Methods in Fluids* **30**, pp. 425–440, 1999.

[10] A. Mahadevan, J. Oliger, and R. Street, "A nonhydrostatic mesoscale ocean model. part i: Ill-posedness and scaling," *Journal of Physical Oceanography* **26**, pp. 1868–1880, 1996.

[11] A. Mahadevan, J. Oliger, and R. Street, "A nonhydrostatic mesoscale ocean model. part ii: Numerical implementation," *Journal of Physical Oceanography* **26**, pp. 1880–1890, 1996.

[12] J. O. Hinze, *Turbulence*, Mc Graw Hill, New York, 1975.

[13] S. B. Pope, *Turbulent Flows*, Cambridge University Press, Cambridge, 2000.

[14] L. H. Kantha and C. A. Clayson, *Numerical Models of Oceans and Oceanic Processes*, Academic Press, San Diego, 2000.

[15] J. A. Jankowski, *A non-hydrostatic model for free surface flows*. PhD thesis, Institut für Strömungsmechanik und Elektronisches Rechnen im Bauwesen, Universität Hannover, Hannover, 1998.

[16] P. Leitão, *Integração de Escalas e Processos na Modelação do Ambiente Marinho*. PhD thesis, Instituto Superior Técnico, Lisboa, 2002.

[17] F. Martins, P. Leitão, A. Silva, and R. Neves, "3d modelling in the sado estuary using a new generic vertical discretization approach," *Oceanologica Acta* **24(1)**, pp. 1–12, 2000.

[18] J. H. Ferzieger and M. Peric, *Computational Methods for Fluid Dynamics*, Springer-Verlag, Berlin, 1997.

[19] J. Backhaus, "A three-dimensional model for the simulation of shelf sea dynamics," *Deutsche Hydrographische Zeitschrift* **38(H4)**, pp. 165–187, 1985.

[20] J. R. Shewchuk, *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. School of Computer Science Carnegie Mellon University, Pittsburgh, 1994. warp.cs.cmu.edu.

[21] G. L. G. Sleijpen and H. A. V. der Vorst, "Hybrid bi-conjugate gradient methods for cfd problems," in *Computational Fluid Dynamics Review*, M. Hafez and K. Oshima, eds., John Wiley & Sons, 1995.