Mathematical Engineering Department

Report on waves and surfing

Partial Differential Equations, Numerics and Control

Author:

Dalphin Jérémy

Internship supervisors

- *at the Ecole Nationale Supérieure des Mines de Nancy :*

  Henrot Antoine

- *at the Basque Center for Applied Mathematics :*

  Zuazua Enrique
  Barros Ricardo

Abstract

The recent development of the Basque Center for Applied Mathematics (BCAM) leads it to collaborate in a project with Instant Surfing. This association wants to build an efficient wave maker with a moving bottom underwater.

The shape of a bottom that would create the highest waves has to be found. This report presents the chosen model : the forced Korteweg and de Vries equation (fKdV). Then, it describes the algorithm of the design optimization process.

Finally, the results of the program are investigated for various bottoms. The behaviour of the waves, the tuning of some parameters, the physical relevance of the model and the errors committed via the algorithm are analysed.

Contents

Acknowledgements

First, I want to thank strongly my tutors Enrique Zuazua and Ricardo Barros for the integration within the team, the availability and the advice they gave me throughout this wonderful work experience.

Finally, I wish to express my gratitude to all BCAM team for their good humor, their dynamism and their warm hospitality.

Introduction

Personal career

Student at the Ecole Nationale Supérieure des Mines de Nancy, I decided to experience a gap year during the period 2009-2010. This choice would allow me to step back on my career plan, to think about the guidelines to make for my future jobs and to gain work experience through various internships in the field of research in applied mathematics.

In 2008, I followed the first year of Master degree in Mathematics at the Henri Poincaré University of Nancy in parallel with the courses of the Mathematical Engineering Department at the Mines. My knowledge is mainly specialized around two axes : probability and partial differential equations. But it has also diversified into algebra, optimization and numerical simulation.

In September 2009, I joined the Centre de Recherche et Innovation Gaz et Energies Nouvelles (CRIGEN) of the gas company GDF SUEZ in order to achieve a six-month internship. In march 2009, I have joined the Basque Center for Applied Mathematics (BCAM) for another internship in Bilbao, Spain. I was integrated to the research line Partial Differential Equations, Numerics and Control directed by Mr. Zuazua.

Subject of the internship

BCAM is a new research center that aims to become an international reference in the field of applied mathematics. Promoting its skills to local companies during workshops, a collaboration with Instant Surfing emerges from this dynamical process. The basque project is directed by professional surfers who try to elaborate fine technologies for the art of riding waves.

Using their wide experience in water, they were granted a patent on a wavemaker prototype of a new type. Although most of the current machines are generating waves by dropping accelerated water in a pool, this one is based on the translation of a bottom underwater. Instant Surfing is now seeking for the most efficient one : the wave maker has to create the highest stable wave as possible.

During my six-month internship in BCAM, I studied the forced Korteweg-de Vries model (fKdV) in order to give answers about how the wave height, shape and stability are ruled by bottom parameters such as speed, shape and water depth. Then, I developed an optimization algorithm so as to obtain an optimal bottom shape that could improve the wave maker. Finally, I investigated the pertinence of the results obtained. The table below sums up monthly the principal steps of my internship.

March	Elaboration of the subject
April	Bibliographic work
May	Studying the forced KdV model
June	Elaborating the optimization algorithm
July	Convergence of the algorithm
August	Writing the report

Layout of the report

First, this report will present my internship environment : BCAM as a new structure emerging from a dynamical research process and from collaborations with local companies. I will develop the context and the challenges of the subject : giving some mathematical clues about the elaboration of an efficient wave maker including a clearly statement of the problematic.

A precise state of the art will be done concerning the elaboration of a wave maker : the choice of the forced KdV model, the computation of the fKdV equation, a blockage coefficient and bottom shape effects in the generation of waves.

Then, under some usual hypothesis made in the context of water waves, I will derive from Euler equations the forced KdV model and discretize the equation at stake. Then, I will present the results obtained and justify my choices in the modelization.

Finally, I will define mathematically the optimization problem : specifying, justifying the functionnal to minimize and the set of admissible solutions chosen. I will describe the optimization algorithm and how the parameters have to be tuned in order to get convergence. I will finally conclude on their influence in the process.

Part I

Presentation of the internship : where did it take place ? From which expectations did it come ? What form does it take ?

Chapter 1

Environment of the internhip : a new research center collaborating with a local project

First, we briefly present BCAM as a growing research center, then Instant surfing as a local project specialized in the art of riding waves, and finally my internship as the concretization of a collaboration between the two entities.

1.1 The Basque Center for Applied Mathematics : a new structure emerging from a dynamical process

Through the Department of Education, Universities and Research, the Basque Government decided to set up a new entity in 2007 : Ikerbasque. This Basque Foundation for Science was charged with three objectives :

- the attraction and recovery of front-rank researchers ;
- the creation of new research centers with standards of excellence ;
- social outreach for science.

Following the second issue, Ikerbasque instructed Mr. Zuazua to study the viability of a mathematical research center in the Basque Country. In 2008, the Board of Trustees decided to concretize the idea and created such a structure as part of the Basque Excellent Research Centres program.

The Basque Center for Applied Mathematics (BCAM)[1] emerged with the commitment to put the Basque Country firmly on the international map in terms of cutting edge research and a strong cooperative spirit. The center started operations recently in September 2008 and is located in Bilbao, Spain.

Formed by a group of highly trained researchers and an extensive network of international excellence, BCAM aims to become an international reference in the field of applied mathematics, promoting scientific and technological developments worldwide.

[1] *http://www.bcamath.org/public_home/ctrl_home.php*

1.2 Instant Surfing : a local project about the elaboration of a wave maker

For surfing lovers, defining the perfect wave has always been a non-trivial problem. In the context of surfing competition, the ability to generate always the same accurate wave is certainly of first interest. Wave pools aim to solve that problem, by controlling all the elements that go into creating perfect surf. However, there are only a handful of wave pools that can simulate really good surfing waves, owing primarily to construction, operation costs and potential liability.

That is why some engineers from Donostia decided to set up a project called Instant Surfing[2]. These surfing lovers try to elaborate fine technologies for the art of riding waves. More precisely, it consists in the construction of a wave maker that would generate the same accurate wave in order to develop competition in pools. Using their wide experience in water, they recently obtained a patent about a wave maker of a new type. After some years of investments, they would like to concretize their project by opening an effective surfing center in the region of San Sebastian.

1.3 The internship at the boundary between two expectations

Research in applied mathematics concerns the development of techniques used in the application of mathematical knowledge to other domains. That is why it appears of first interest for companies who definitively need the researchers' experience to innovate more and always solve new problems arising from all areas of industry.

Therefore, BCAM wants to promote itself to companies as an excellent center whose research problematics are closely related to the industrial ones. In order to keep abreast of the crucial issues in fluid dynamics, the center organized a workshop in February 2010 where lots of local firms where invited.

Instant Surfing presented its work. The engineers who built the wavemaker had no real experience in wave theory and in numerical simulation. As they would like to concretize their project, they ask BCAM about some mathematical answers about the efficiency and the improvement of their wave maker.

The internship subject was created to study the wave maker mathematically in order to give some answers about Instant Surfing expectations. I was integrated to the research line Partial Differential Equations, Numerics and Control directed by Mr. Zuazua.

[2] *http://instantsurfing.net/*

Chapter 2

Context of the internship : the elaboration of an efficient wave maker for surfing

We present the need of wave makers for surfing competitions, their current operating principle and the constraints under which they are subject. Then, we describe the new prototype elaborated by Instant surfing and the questions arising from it.

2.1 The need of wave makers for surfing competitions

Those involved in surfing competitions should be able to surf exactly in the same conditions in order to satisfy a certain equality in the evaluation of performances. This is clearly not possible in the sea where waves depends on many uncontrollable parameters like weather, wind, stream,...

That is why the elaboration of a wave maker in a pool seems to be a reasonnable answer to this need. However, before debating about the way to build such a machine, a problem arises when trying to define a wave for surfing competition. It mainly depends of :

- its shape that must ensure its stability ;
- its speed that allow its propagation ;
- its height that must be sufficiently high.

Moreover, we can distinguish also two completly different aspects in the generation of surfing waves :

1. The formation of the wave. Accelerated water is generally dropped in a pool in order to generate a solitary wave : a very stable gravity wave propagating without changing its shape and ruled by only one parameter : its height.
2. The breaking of the wave. The depth of the pool ground is usually reduced in order to create the shoaling effect[11] : a slope going from the ground to the water surface will increase the wave height and will break the wave at last.

The purpose of this report will be devoted to the first aspect of the wave maker.

2.2 The constraints involved in the construction of a wave maker

In order to build an efficient wave maker, many problems quickly arise in the running of such a machine. Although the first preoccupation of surfers is the quality of the wave generated, economical constraints must also be taken in account. First, the security of the surfers in the pool must be ensured by the wave maker.

Then, a wave maker consumes a lot of energy : lifting water each time one needs to produce a wave is very greedy energetically speaking. Indeed, the volume of water lifted is equal to the volume of the deformation produced. This implies even more energy if the wave height is high.

Finally, the depth of the pool is very important in the propagation of waves. Deeper is the pool and higher are the waves that can be propagated. This implies an extra cost for filling such a pool with water. These two economical aspects are fundamental in the viability of a surfing center and they often are the reasons of its closure.

2.3 The prototype developped by Instant surfing : a moving bottom underwater

Instant surfing decided to use another identified phenomena to generate waves : a bottom is translating under the water and creates a solitary wave upstream the disturbance. This method should be less greedy in energy than the previous one. However, the security is not ensured in this case.

In order to secure the wave maker, Instant Surfing also decided to modify the breaking wave principle. A slope parallel to the bottom rails gets from underwater to an artificial beach. This has two natural effects [11] :

- The adherence of the ground : the waves generated in front of the bottom and perpendicularly to the beach tend to turn parallely to it.
- The shoaling effect : the waves increase in height and breaks at last.

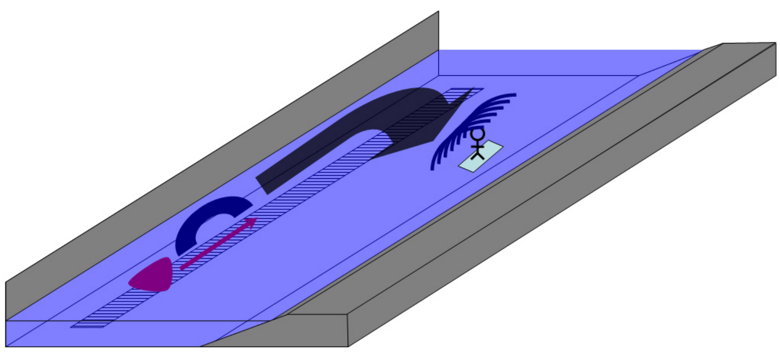

Figure 2.1: *The wave maker works by moving a bottom underwater which generates solitary waves upstream that tend to turn and break parallely to the beach where surfers are safe.*

Chapter 3

Subject of the internship : can mathematics and numerical simulations give some answers about the efficiency of such a wave maker, especially concerning the shape of the bottom ?

Now that Instant surfing patented their wavemaker prototype, we describe the new needs of the project where BCAM can bring some answers. Then, a clear statement of the problematic will be made and finally we will explain the complexity of the problem.

3.1 Beyond experiments, the need of mathematics and numerical simulations

After some years of expensive investments, Instant Surfing was granted a patent for their wavemaker prototype. However, the real concretization of the project would be the construction of a real surfing center in the region of San Sebastian. Therefore, the engineers started to think about the possible commercialization of their wave maker.

This implies some studies about the security, the viability and the efficiency of their system. For example, surfers must not get wounded by the moving bottom. Moreover, the cost in energy, water and maintenance must lead to a reasonnable entrance price.

Finally, the wave height generated is still too small for surfing competitions. In order to improve again the wave maker, two possibilities are considered :

- Instant Surfing can spend more money in experiences that should enhance the wavemaker performances. The engineers pratical skills are of fundamental importance in the tuning of the waves generated.
- BCAM can offer them some mathematical clues about the efficiency of their wavemaker completed by some numerical simulations, a cheaper way to better the current prototype.

3.2 Statement of the report topic

The first objective of the report will try to investigate mathematically the relations between the wave parameters and the bottom ones. More precisely, an accurate model is needed in order to obtain a deeper understanding about how the height, the shape, the speed and the stability of the generated wave is linked to the length, the height, the shape, the speed and the depth of the moving bottom.

However, we will concentrate mainly on the following design optimization problem : how the shape of the bottom can affect the height of the wave in order to maximize it ? Using the model, numerical simulations and an optimization algorithm will be developped in order to obtain pertinent results on that precise question.

3.3 Complexity of the problem

First of all, this design optimization approach is very new. Indeed, although many models from water waves theory has been tremendously studied, very few experiments have been done concerning some design optimization problems. The only paper found in the litterature that studies the shape effects of submerged objects in the generation of solitary waves is D. Zhang and A.T. Chwang's one [5].

Then, the problem lies at the frontiers of three domains : mathematics, physics and numerics. The chosen model must be as close as possible from the physical observations whereas simplicity is required for the computation and the mathematical study.

Finally, the validity of the results obtained via the algorithm is not obvious. Indeed, in order to compute the optimization problem, a discretization of the continuous model is necessary. Convergence, stability and consistency of the algorithm are non trivial questions difficult to prove.

Part II

State of the art

Chapter 4

A model adapted to the wave maker

We recall here the main properties of the wave equation and how useful they are in the context of wave makers. Then, we present the Korteweg-de Vries equation arising in many descriptions of real wave propagation, and one of its solution recovering almost the desired properties : the solitary wave. Finally, we introduce the forced Korteweg-de Vries equation.

4.1 The wave equation

The simplest model for one-dimensional water motion a mathematician can think of is the wave equation. Restricted to right-going waves at speed c_0 for our purpose, the water elevation ζ is the famous D'Alembert solution :

$$\begin{cases} \dfrac{\partial \zeta}{\partial t} + c_0 \dfrac{\partial \zeta}{\partial x} = 0 \\ \\ \zeta(x,0) = \zeta_0(x) \end{cases} \Longrightarrow \forall x \in \mathbb{R},\ \forall t \in \mathbb{R}_+, \quad \zeta(x,t) = \zeta_0\left(x - c_0 t\right)$$

This model has a lot of very useful properties for a wave maker :

- The speed of the wave c_0 can be controlled because it is determined by the water depth denoted h_0. Indeed, the relation $c_0 = \sqrt{g h_0}$ holds where g is the gravitaty acceleration.
- There is no interaction between two waves of this type. Indeed, the wave equation is linear and the superposition principle holds[1].
- As the wave do not change its shape when propagating, the wave profile is just a shift of the initial one generated.

Unfortunately, this model is valid only for long waves of small amplitude. Altought it is used for describing the spreading of tsunamis [14] because the ocean depth is very high compared to the amplitude of the wave, it cannot be used for the description of a wave generated in a pool that must be sufficiently high for surfing. Indeed, more physical and complex effects always appear. Normal waves would tend to either flatten out due to dispersion or steepen and topple over due to the non-linearity.

[1] The superposition principle says that if ζ_1 and ζ_2 are solutions of the wave equation, then $\zeta_1 + \zeta_2$ also.

4.2 The Korteweg-de Vries equation

The Korteweg-de Vries (KdV) equation is the simplest relation that incorporates both nonlinearity and dispersion [10]. In fact, it often occur in the description of real wave propagation. Consider a linear one-dimensional wave motion with dispersion. As waves of different wave number k propagate at different velocities c, the dispersion relation must take the form :

$$\omega(k) = kc\left(k^2\right)$$

since only odd derivative of the wave elevation ζ are allowed. Let's suppose that for infinitely long waves ($k \to 0$), there exists a non-zero speed of propagation c_0, then we have in first order of approximation the relation :

$$\frac{\omega}{k} \sim c_0 - \vartheta k^2$$

and usually long waves travel the fatest, so $\vartheta > 0$. This approximate dispersion relation is then clearly obtained by inserting the harmonic wave solution $(x,t) \mapsto e^{i(kx-\omega t)}$ in the dispersion wave equation :

$$\frac{\partial \zeta}{\partial t} + c_0 \frac{\partial \zeta}{\partial x} + \vartheta \frac{\partial^3 \zeta}{\partial x^3} = 0$$

Moreover, if the medium in which the propagation is occuring is a classical continuum, then the time evolution will be given by the material derivative $\frac{D}{Dt} = \frac{\partial}{\partial t} + u\frac{\partial}{\partial x}$. If these two effects, dispersion and nonlinearity, are to balance, we should obtain the relation :

$$\frac{\partial \zeta}{\partial t} + c_0 \frac{\partial \zeta}{\partial x} + \alpha\left[\zeta \frac{\partial \zeta}{\partial x} + \vartheta \frac{\partial^3 \zeta}{\partial x^3}\right] = 0$$

where α is a small parameter measuring the weak non-linearity and long waves. Thus, we have the equation :

$$\frac{\partial \zeta}{\partial \tau} + \zeta \frac{\partial \zeta}{\partial \xi} + \vartheta \frac{\partial^3 \zeta}{\partial \xi^3} = 0 \qquad \xi = x - c_0 t, \quad \tau = \alpha t$$

which is the KdV equation for small amplitude long waves, valid in an appropriate region of the (x,t)-plane defined by $x - c_0 t = O(1)$ and $t = O\left(\alpha^{-1}\right)$, as $\alpha \to 0$.

4.3 The solitary wave

In 1834, John Scott Russell first observed the "Great Wave of Translation" in the Glasgow canal while he was conducting experiments to determine the most efficient design for canal boats. Intrigued by something very peculiar in a seemingly ordinary event, he decided to perform some laboratory experiments, generating what he called solitary waves by dropping a weight at one end of a water channel.

However, Scott Russell's observations of the single localised entity could not be explained by the existing water-wave theories. Challenging about its existence the mathematical community influenced by Airy and Stokes who had difficulty to accept the validity of his experiments, it took half a century to Rayleigh and Boussinesq to obtain a formula for the wave and derive it from the Navier Stokes equation [7]. Korteweg and de Vries finally unified the approach in 1895 showing that the solitary wave is one solution of the KdV equation :

$$\zeta\left(x,t\right)=a\ \mathrm{sech}^{2}\left[\frac{1}{2h_0}\sqrt{\frac{3a}{h_0}}\left(x-c_0\left(1+\frac{a}{2h_0}\right)t\right)\right]$$

where h_0 is the water depth and a the wave amplitude.

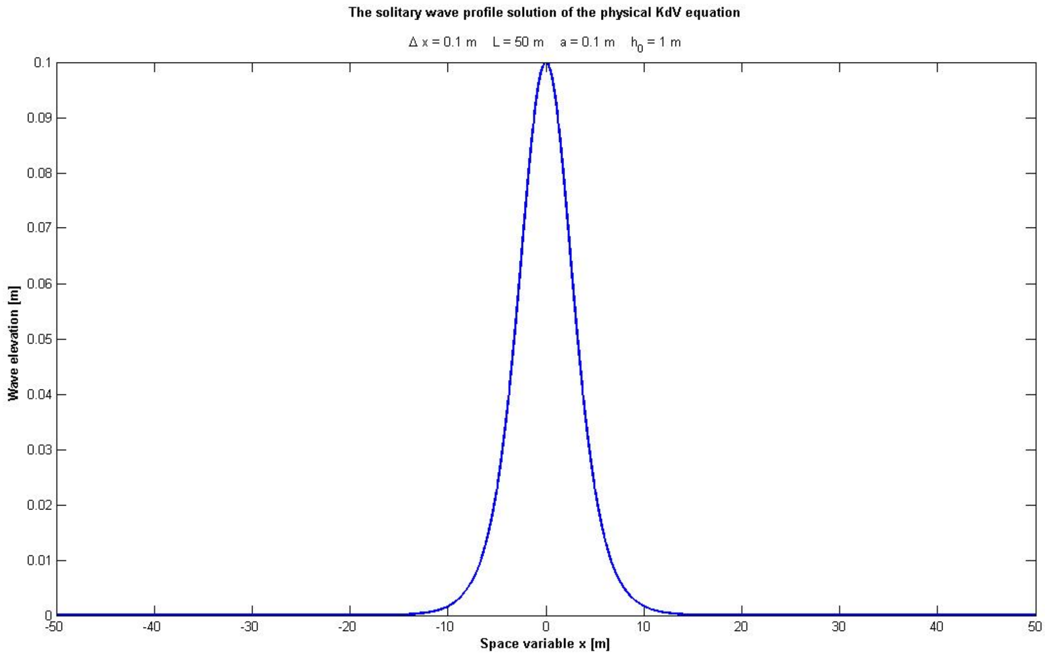

Figure 4.1: *Profile of a solitary wave solution of the KdV equation.*

All the key properties of the solitary wave were hidden in Russell's *Report on Waves* until Zabusky and Kruskal re-discovered the unusual interactions between this waves [12] called solitons and leading to a broad development on integrable systems like the elaboration of the scattering method.

- When generated, a solitary wave is very stable and can travel over very large distances without changing its initial shape. This property is very useful for a wave maker.
- Higher waves travel faster because we have the relation $c^2 = g\left(a+h_0\right)$ where a is the amplitude of the wave, h_0 the undisturbed water depth and g the gravity acceleration.
- Although there are interactions between two solitary waves due to the nonlinearity, they will never merge and they seem not to interact with each other.

With an accurate balance between dispersion effects that tend to spread out the wave profile, and nonlinearity ones that create shocks, the KdV equation has a more physical and stable solution which almost recover all the properties found in d'Alembert solution. That is why a wave maker often tries to generate such a solitary wave ruled by only one parameter : its amplitude.

4.4 The forced Korteweg-de Vries equation

Solitary waves can be generated by a pressure or a bottom disturbance.

- A ship can apply a moving pressure which acts as a forcing term and generate waves. This is how Russell discovered the solitary wave.
- Like in Russell's experiments, a pressure bump appears to be the most efficient way to produce a soliton. This is how most of the wave maker are working actually but it cost a lot of energy.
- Submarine earthquake can also generate solitary waves by a bottom bump. It usually has catastrophic consequences due to the stability of the wave generated.
- A moving bottom can generate periodically a succesion of solitary waves upstream the disturbance. Less energetically greedy, this is the way that has been chosen by Instant Surfing to generate solitary waves.

The last phenomenon has been identified and studied by Wu in [20]. A forcing disturbance $(x,t) \mapsto b\,(x - Ut)$ is moving steadily under a h_0 water depth in shallow water. Its speed U must be taken near the transcritical velocity $c_0 = \sqrt{gh_0}$ to observe solitons. Under this condition, the bottom b always generate periodically a succession of solitary waves advancing upstream the disturbance while a train of weakly nonlinear and weakly dispersive waves develops downstream of a region a depressed water surface trailing just behind the disturbance.

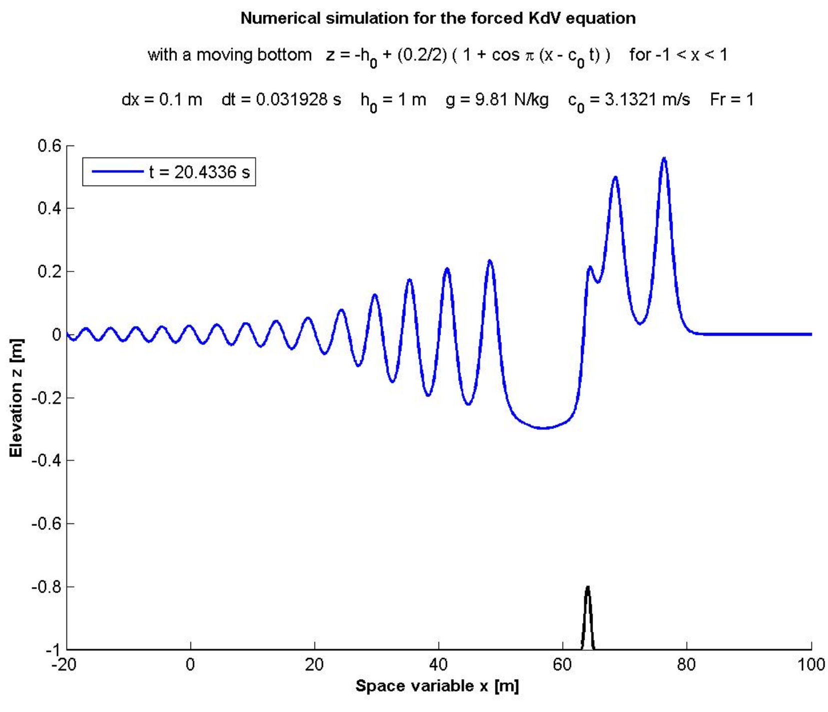

Figure 4.2: *Numerical simulation of the forced KdV equation highlighting the three regimes created by the wave maker : cnoïdal-like waves downstream, depressed water surface and solitons upstream.*

The generalized Boussinesq (gB) model is usually used for describing such bottom or pressure disturbances. Although it is simpler than the full Euler equations, it is still difficult to explore the basic mechanism underlying the phenomenon. A simpler version is develop for bidirectional waves in [3]. However, we are here in presence of unidirectional phenomena. Derived from gB by Wu in [20], it appears to be ruled by the forced Korteweg-de Vries (fKdV) equation where the wave elevation ζ satisfies the following relation in the frame of the bottom $b\left(x-Ut\right)=b\left(\xi\right)$:

$$\begin{cases} -\dfrac{1}{c_0}\dfrac{\partial\zeta}{\partial t}+\left[\left(\dfrac{U}{c_0}-1\right)-\dfrac{3}{2h_0}\zeta\right]\dfrac{\partial\zeta}{\partial\xi}-\dfrac{h_0^2}{6}\dfrac{\partial^3\zeta}{\partial\xi^3}=\dfrac{1}{2}\dfrac{\partial b}{\partial\xi} \\ \\ \zeta\left(\xi,0\right)=\zeta_0\left(\xi\right) \end{cases}$$

The initial value problem for the fKdV equation has been studied in [21] and in [9] when tension effects are taken in account. Let $T>0$ be given. For any initial data ζ_0 in $L^2\left(\mathbb{R}\right)$ and for any forcing term f in $L^2\left(-T,T;L^2\left(\mathbb{R}\right)\right)$, the previous initial value problem admits a unique solution ζ in a certain space $Y_{0,b}^T$ and the corresponding mapping is analytic.

We define for $s>-1$ and $b>\dfrac{1}{2}$ the space $Y_{s,b}$ as the completion of the space $\mathcal{S}\left(\mathbb{R}\right)$ of tempered test functions with respect to the norm

$$\|f\|_{Y_{s,b}}^2=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\left(1+|\tau-\xi^3|\right)^{2b}\left(1+|\xi|\right)^{2s}|\widehat{f}\left(\xi,\tau\right)|^2d\xi d\tau$$

where $\widehat{f}$ denotes the Fourier transform of f. For any given $T>0$, we define the following restriction to $\left(-T,T\right)$ of a function in $Y_{s,b}$. We define the equivalence relation in $Y_{s,b}$ such that $v\sim w\Leftrightarrow\forall t\in\left(-T,T\right),v(.,t)=w(.,t)$. We call $Y_{s,b}^T$ the set of equivalence classes which is the appropriate space where the solution of the forced KdV equation lives.

Chapter 5

Computation of the forced KdV equation

We present the main schemes developpped in order to solve the forced KdV equation. We show how difficulties are arising from the discretization of the third derivative and of the forcing term. We justify our choices for the scheme.

5.1 Difficulties encountered with the finite difference schemes

As it is well explained in [16], when discontinuous or near-discontinuous features are present in a physical system such as the forced KdV equation, it is possible for finite difference centered schemes to perform badly as they do not take account of the direction of propagation.

Moreover, the third order derivative of the fKdV equation implies high accuracy of the space discretization if it does not want to be polluted by numerical dispersion. Finally, the forcing term in the fKdV equation breaks all the symetries that could be exploited in the computational process such as the symplectic structure.

However, our goal here is to build a scheme that allow a fast resolution in order to incorporate it in the loop of the optimization algorithm. That is why simplicity is required for a fast computation time but also for an easier study of the properties expected from the discretization.

In [12], Zabusky and Kruskal developped a leap-frog scheme for time discretization and a finite difference scheme for space discretization that conserves mass and energy to second order. Unfortunately, the scheme is stable with a CFL condition of the form $\Delta t = O\left(\Delta\xi^3\right)$. As we want to evaluate the free surface elevation ζ for long time and incorporate the fKdV resolution in a loop of a design optimization algorithm, such a condition imposes too much calculations.

In [16], Walkley uses finite difference schemes of higher order of approximation for the space discretization. However, to avoid the CFL condition, he builds an implicit scheme for the time discretization which is very complicated for our purpose and is described in [17]. Although the fKdV equation seems to be stiff, the simplicity of the general behaviour suggests that an easier scheme exists.

5.2 Difficulties encountered with spectral methods

In [10], Johnson presents the spectral methods as a better alternative to the finite difference schemes. In [19], Fornberg and Whitham developped a clever scheme for the computation of the KdV equation, using the discrete Fourier transform.

In [15], Trefethen also took the Fourier transform of the equation and uses the method of integrating factors to allow larger time step for stability. He then set up a fourth-order Runge-Kutta method [18] for the time integration.

However, after a certain time, the cnoïdal waves leaving from the left side would re-enter to the right side and pollute the solitary waves generated. Indeed, a scheme based on Fourier transform assumed that the function is periodic on the computationnal domain.

Therefore, a big computationnal domain must be taken for long time simulation which increases drastically the computation time. That is why spectral methods or splitting methods usually used for more complex equation and described in [8] were too complicated for the problem.

5.3 Differentiate an equation to gain numerical stability

Finally, we decided to set up a scheme described in [1]. Deriving the fKdV equation in space seems to introduce a numerical stabilization as only even derivatives now appear in the equation. We discretize the following equation with a second order accurate finite difference scheme in space and a Crank-Nicholson scheme in time :

$$\frac{-2}{c_0}\frac{\partial^2\zeta}{\partial\xi\partial t} + 2(F_r - 1)\frac{\partial^2\zeta}{\partial\xi^2} - \frac{3}{2h_0}\frac{\partial^2\left(\zeta^2\right)}{\partial\xi^2} - \frac{h_0^2}{3}\frac{\partial^4\zeta}{\partial\xi^4} = \frac{\partial^2 b}{\partial\xi^2}.$$

Therefore, this scheme has many advantages :

- It does not assume the solution to be periodic in the computational domain ;
- It is an semi-implicit scheme that allow big time step, reducing the computation time ;
- Its simplicity translates the straightforwardness of the general behaviour encountered in the fKdV model.

Chapter 6

The effects of the blockage coefficient and of the shape in the generation of the solitary waves

Few results were published on this topic. In [5] and [4], Zhang and Chwang used numerical simulations of the viscuous Navier Stokes set of equations to study the generation of solitary waves with submerged objects. They confirmed the good accuracy of their simulations compared to the experiments made by Lee and al. in 1989. They also showed a very good accuracy of the forced KdV model in its domain of validity.

6.1 The effects of the blocage coefficient

They showed that a blockage coefficient, defined as the ratio of the mid-ship-sectionnal area to the cross-sectionnal area of the channel, plays a key role in the generation of solitons. In the monodimensional case, this parameter, denoted δ in the report, is simply the ratio of the bottom height d to the water depth h_0.

The amplitude increases as the blockage coefficient increases. There is a quasi-linear relation between this two parameters in good agreement with the previous experiments made on this topic. They also compared their results with the ones described by Wu and in [13]. Inviscid models [13] seems to incorporate the effects of the blockage coefficient as well as viscuous ones.

6.2 The effects of the shape

They found that the shape of a body under the free surface has a significant effect on the solitary-wave generation through viscous effects in the boundary layer of the body. In general, if a change in shape results in increasing the area of the body surface, the viscous effects will be enhanced and so will the disturbance forcing. Therefore, the amplitude of the solitary waves increases.

However, they showed that for an inviscid flow, the shape of a body under the free surface has no real effect on the generation of upstream-advancing solitary waves but has an effect on the depressed water region and trailing waves when the body length is sufficiently short. This highlights maybe a weakness of the forced KdV model.

Part III

The forced Korteweg-de Vries model

Chapter 7

Derivation of the forced KdV model

The Navier Stokes equations for a free surface called Euler equations are re-established here. The appropriate hypothesis are made to derive the fKdV model from it.

7.1 Establishment of Euler equations for a free surface

7.1.1 Hypothesis for the description of a continuous environment

Let's consider a fluid in motion, denoting ρ its density, λ the typical length, h_0 the water depth and $c_0 = \sqrt{gh_0} \sim 3$ m.s^{-1} the speed of classical gravity waves where $g \sim 10$ N.kg^{-1} is the gravitational acceleration.

Quantity	Notation	Order of magnitude
Constant density of water	ρ	~ 1000 kg.m^{-3}
Typical wavelength	λ	~ 1 m
Undisturbed water depth	h_0	~ 1 m
Gravitation acceleration constant	g	~ 10 m.s^{-2}
Characteristic speed for waves	$c_0 = \sqrt{gh_0}$	~ 3 m.s^{-1}

Of course, we will work with the model of classical physics (no relativist or quantum effects). This is justified by [1] :

$$\begin{cases} c_0 = \sqrt{gh_0} << c_{light} & \Longleftrightarrow \quad 3 \;\; \text{m.s}^{-1} << 300\ 000 \;\; \text{m.s}^{-1} \\ h_{Planck} << \rho h_0 c_0 \lambda^3 & \Longleftrightarrow \quad 6.63 \;\; 10^{-34} \;\; \text{J.s} << 3000 \;\; \text{J.s} \end{cases}$$

In the usual space, the observator is located in the galilean earth frame. Everyone can perceive the time in the same way.

The medium is considered continuous when the charateristic dimension of the flow λ is much more higher than the mean free path of a molecule of the fluid which is the case here. In these conditions, the function introduced to characterized the fluid are continuous and differentiable.

[1] The constant c_{light} is the speed of light and h_{Planck} the Planck constant.

7.1.2 A quick re-establishment of the Navier Stokes equations

Let's consider a two-dimensionnal fluid characterized by its density ρ assumed to be constant (which implies that the fluid is incompressible), its speed $\overrightarrow{u}(x,z,t) = (u(x,z,t), w(x,z,t))$ and its pressure $P(x,z,t)$.

Conservation of mass

Considering a fixed and closed volume Σ of surface $\partial\Sigma$ in our modeling, the variation of mass $\frac{d}{dt}\int_{\Sigma}\rho dV$ is only generated by a mass flow $-\int_{\partial\Sigma}\rho\overrightarrow{u}.\overrightarrow{n}dS$. As Σ is fixed and ρ differentiable, applying Green-Ostrogradski formula, we get :

$$\int_{\Sigma}\left[\frac{\partial\rho}{\partial t} + div(\rho\overrightarrow{u})\right]dV = 0.$$

As Σ is not specified, the local formulation of incompressibility can be rewriten when ρ is assumed to be constant:

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0$$

Conservation of momentum

Considering Σ, a fixed-closed volume of surface $\partial\Sigma$, the variation of momentum $\frac{d}{dt}\int_{\Sigma}\rho\overrightarrow{u}dV$ inside the volume is generated by the momentum flow $-\int_{\partial\Sigma}\rho\overrightarrow{u}(\overrightarrow{u}.\overrightarrow{n})dS$ through the surface and by the resultant of the forces applying on the volume.

The fluid is assumed to be inviscid so the forces are only composed of pressure $-\int_{\partial\Sigma}p\overrightarrow{n}dS$ and gravity $\int_{\Sigma}\rho\overrightarrow{g}dV$. As the volume is fixed and the momentum differentiable, we get after applying Green-Ostrogradski formula :

$$\int_{\Sigma}\left[\frac{\partial\left(\rho\overrightarrow{u}\right)}{\partial t} + div(\rho\overrightarrow{u}^{t}\overrightarrow{u}) + \overrightarrow{\nabla}(P) + \rho\overrightarrow{g}\right]dV = 0.$$

The volume is arbitrary so we obtain the local formulation for the equations of motion :

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} + w\dfrac{\partial u}{\partial z} + \dfrac{1}{\rho}\dfrac{\partial P}{\partial x} = 0 \\[2ex] \dfrac{\partial w}{\partial t} + u\dfrac{\partial w}{\partial x} + w\dfrac{\partial w}{\partial z} + \dfrac{1}{\rho}\dfrac{\partial P}{\partial z} + g = 0 \end{cases}$$

Conservation of energy

Neglecting the thermal exchanges in this modeling, the conservation of energy is dropped. The thermal effects are not linked anymore with the dynamical ones.

7.1.3 The boundary conditions

When there is no bottom or no mouvement, we assume that the water level is at $z = 0$ and that the bottom is at $z = -h_0$. We consider then a free surface of equation $z = \zeta(x,t) = a\eta(x,t)$ with an elevation of amplitude a and a bottom underwater characterized by the equation $z = b(x,t) = -h_0 + d\ \theta(x,t)$ and an amplitude of d.

Free surface kinematic condition

Assuming that the surface $z = \zeta(x,t)$ moves with the fluid so that it always contains the same fluid particles, we obtain the kinematic conndition for the free surface:

$$w(x,\zeta(x,t),t) = \frac{\partial \zeta}{\partial t}(x,t) + u(x,\zeta(x,t),t)\frac{\partial \zeta}{\partial x}(x,t)$$

Free Surface dynamic condition

We neglect the tension effects and assume that the pressure P on the free surface is constant to the atmospheric pressure P_a. We are now going to define another pressure p which is the perturbation to the atmospheric pressure effect. Hence, let's introduce the pressure $P(x,z,t) = P_a - \rho g z + p(x,z,t)$. The condition $P = P_a$ on the free surface becomes now :

$$p(x,\zeta(x,t),t) = \rho g \zeta(x,t).$$

Bottom kinematic condition

For an inviscid fluid, the bottom constitutes like the free surface a boundary which moves with the fluid. Hence, we get the same kinematic condition for the bottom :

$$w(x,b(x,t),t) = \frac{\partial b}{\partial t}(x,t) + u(x,b(x,t),t)\frac{\partial b}{\partial x}(x,t)$$

7.1.4 The Euler equations

Avoiding the thermal and tension effects, for an inviscid fluid of constant density ρ, the two dimensionnal Euler equations are given by :

$$\boxed{\left\{\begin{array}{l}
\dfrac{\partial u}{\partial x} + \dfrac{\partial w}{\partial z} = 0 \\[2ex]
\dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} + w\dfrac{\partial u}{\partial z} + \dfrac{1}{\rho}\dfrac{\partial p}{\partial x} = 0 \\[2ex]
\dfrac{\partial w}{\partial t} + u\dfrac{\partial w}{\partial x} + w\dfrac{\partial w}{\partial z} + \dfrac{1}{\rho}\dfrac{\partial p}{\partial z} = 0 \\[2ex]
w(x,\zeta(x,t),t) = \dfrac{\partial \zeta}{\partial t}(x,t) + u(x,\zeta(x,t),t)\dfrac{\partial \zeta}{\partial x}(x,t) \\[2ex]
p(x,\zeta(x,t),t) = \rho g \zeta(x,t) \\[2ex]
w(x,b(x,t),t) = \dfrac{\partial b}{\partial t}(x,t) + u(x,b(x,t),t)\dfrac{\partial b}{\partial x}(x,t)
\end{array}\right.}$$

7.2 Nondimensionalization and scaling of the equations

7.2.1 Nondimensionalization of the variables

We recall here that λ is a typical length distance for waves, h_0 the undisturbed water depth. The speed of the wave solution in the linear case is then $c_0 = \sqrt{gh_0}$. We can then construct and typical time for the waves which is the ratio of the length and the speed. We define new variables :

$$x^* = \frac{x}{\lambda} \qquad z^* = \frac{z}{h_0} \qquad t^* = \frac{c_0 t}{\lambda} \qquad u^*(x^*, z^*, t^*) = \frac{u(x,z,t)}{c_0}$$

The first question is how we can define a characteric speed for w. If we want to preserve the incompressibility relation with the non-dimensional variables, then we must take $w^*(x^*, z^*, t^*) = \dfrac{\lambda w(x,z,t)}{h_0 c_0}$ and it comes :

$$\frac{\partial u^*}{\partial x^*} + \frac{\partial w^*}{\partial z^*} = 0.$$

Then, if we want to preserve also the form of the first motion equation, then we must introduce $p^*(x^*, z^*, t^*) = \dfrac{p(x,z,t)}{\rho g h_0}$. A new parameter appear in the second equation of motion $\dfrac{h_0}{\lambda}$ called the long wavelength parameter

$$\begin{cases} \dfrac{\partial u^*}{\partial t^*} + u^* \dfrac{\partial u^*}{\partial x^*} + w^* \dfrac{\partial u^*}{\partial z^*} + \dfrac{\partial p^*}{\partial x^*} = 0 \\ \left(\dfrac{h_0}{\lambda}\right)^2 \left[\dfrac{\partial w^*}{\partial t^*} + u^* \dfrac{\partial w^*}{\partial x^*} + w^* \dfrac{\partial w^*}{\partial z^*}\right] + \dfrac{\partial p^*}{\partial z^*} = 0 \end{cases}$$

For the boundary conditions, we wrote the free surface equation as $z = \zeta(x,t) = a\eta(x,t)$. We introduce $\eta^*(x^*, t^*) = \eta(x,t)$ and $\zeta^*(x^*, t^*) = \dfrac{\zeta(x,t)}{h_0}$ and we do the same for the bottom of equation $z = b(x,t) = -h_0 + d\,\theta(x,t)$ with $\theta^*(x^*, t^*) = \theta(x,t)$ and $b^*(x^*, t^*) = \dfrac{b(x,t)}{h_0}$. Finally, with the new non-dimensional variables, we get a new set of equations for the boundary conditions introducing two new parameters : $\dfrac{a}{h_0}$ the free surface amplitude parameter and $\dfrac{d}{h_0}$ the bottom amplitude parameter.

$$\begin{cases} w^*\left(x^*, \dfrac{a}{h_0}\eta^*(x^*, t^*), t^*\right) = \dfrac{a}{h_0}\left[\dfrac{\partial \eta^*}{\partial t^*}(x^*, t^*) + u^*\left(x^*, \dfrac{a}{h_0}\eta^*(x^*, t^*), t^*\right)\dfrac{\partial \eta^*}{\partial x^*}(x^*, t^*)\right] \\ p^*\left(x^*, \dfrac{a}{h_0}\eta^*(x^*, t^*), t^*\right) = \dfrac{a}{h_0}\eta^*(x^*, t^*) \\ w^*\left(x^*, -1 + \dfrac{d}{h_0}\theta^*(x^*, t^*), t^*\right) = \dfrac{d}{h_0}\left[\dfrac{\partial \theta^*}{\partial t^*}(x^*, t^*) + u^*\left(x^*, -1 + \dfrac{d}{h_0}\theta^*(x^*, t^*), t^*\right)\dfrac{\partial \theta^*}{\partial x^*}(x^*, t^*)\right] \end{cases}$$

7.2.2 Scaling of the variables

The six equations above show that respectively three parameters are going to give informations on the different regimes for the motion of the waves.

Parameter	Notation	Expression
Long wave length parameter	ε	$\left(\frac{h_0}{\lambda}\right)^2$
Free surface amplitude parameter	α	$\frac{a}{h_0}$
Bottom amplitude parameter	δ	$\frac{d}{h_0}$

We can first notice that if α tends to zero, the pressure p tends to zero and so does the free surface elevation ζ. Hence there is no waves and we have to rescale the variable. We introduce :

$$\widehat{u}(x^*, z^*, t^*) = \frac{u^*(x^*, z^*, t^*)}{\alpha}, \quad \widehat{w}(x^*, z^*, t^*) = \frac{w^*(x^*, z^*, t^*)}{\alpha} \quad \text{and} \quad \widehat{p}(x^*, z^*, t^*) = \frac{p^*(x^*, z^*, t^*)}{\alpha}.$$

Then, we are going to seek solution so that $\alpha = O(\varepsilon)$ et $\delta = O(\varepsilon^2)$ [2]. Using $\varepsilon \approx \alpha$ and $\delta \approx \varepsilon^2$, the system now only depend on ε:

$$\begin{cases} \dfrac{\partial \widehat{u}}{\partial x^*} + \dfrac{\partial \widehat{w}}{\partial z^*} = 0 \\[2ex] \dfrac{\partial \widehat{u}}{\partial t^*} + \varepsilon \left[\widehat{u} \dfrac{\partial \widehat{u}}{\partial x^*} + \widehat{w} \dfrac{\partial \widehat{u}}{\partial z^*} \right] + \dfrac{\partial \widehat{p}}{\partial x^*} = 0 \\[2ex] \varepsilon \left[\dfrac{\partial \widehat{w}}{\partial t^*} + \varepsilon \left(\widehat{u} \dfrac{\partial \widehat{w}}{\partial x^*} + \widehat{w} \dfrac{\partial \widehat{w}}{\partial z^*} \right) \right] + \dfrac{\partial \widehat{p}}{\partial z^*} = 0 \\[2ex] \widehat{w}\left(x^*, \varepsilon\eta^*, t^*\right) = \dfrac{\partial \eta^*}{\partial t^*}(x^*, t^*) + \varepsilon \widehat{u}\left(x^*, \varepsilon\eta^*, t^*\right) \dfrac{\partial \eta^*}{\partial x^*}(x^*, t^*) \\[2ex] \widehat{u}\left(x^*, \varepsilon\eta^*, t^*\right) = \eta^*(x^*, t^*) \\[2ex] \widehat{w}\left(x^*, -1 + \varepsilon^2\theta^*, t^*\right) = \varepsilon \left[\dfrac{\partial \theta^*}{\partial t^*}(x^*, t^*) + \varepsilon \widehat{u}\left(x^*, -1 + \varepsilon^2\theta^*, t^*\right) \dfrac{\partial \theta^*}{\partial x^*}(x^*, t^*) \right] \end{cases}$$

[2] We recall here that a function g dominate a function f in 0, and we note $f = O(g)$, if there exists a constant M so that in the neighbourhood of 0, we have $f \leqslant Mg$.

7.2.3 A unidirectional and long-time scaling model

We are now going to assume that the bottom b is moving at a constant speed U. We place ourself in the frame of the bottom and introduce the variable $\xi = x - Ut$ which gave in non-dimensionless form $\xi^* = \dfrac{\xi}{\lambda} = x^* - F_r t^*$ where $Fr = \dfrac{U}{c_0}$ is the Froude number. Hence we get $b^*(x^*, t^*) = \overline{b}(\xi^*)$ with no time dependance anymore.

Moreover, we are going to seek the motion for long time scaling introducing $\tau = \varepsilon t$ or $\tau^* = \varepsilon t^*$ in non-dimensionless form. With these new variables (ξ^*, z^*, τ^*) the model becomes now unidirectionnal, searching for waves in the far-field of the time domain.

We introduce again a new notation for all the functions :

$$\begin{cases} \overline{u}(\xi^*, z^*, \tau^*) = \widehat{u}(x^*, z^*, t^*) \\ \overline{w}(\xi^*, z^*, \tau^*) = \widehat{w}(x^*, z^*, t^*) \\ \overline{p}(\xi^*, z^*, \tau^*) = \widehat{p}(x^*, z^*, t^*) \\ \overline{\eta}(\xi^*, \tau^*) = \eta^*(x^*, t^*) \\ \overline{\theta}(\xi^*) = \theta^*(x^*, t^*) \end{cases}$$

We can now rewrite the system of equations scaled correctly to seek for fKdV equation:

$$\boxed{\begin{cases} \dfrac{\partial \overline{u}}{\partial \xi^*} + \dfrac{\partial \overline{w}}{\partial z^*} = 0 \\ -F_r \dfrac{\partial \overline{u}}{\partial \xi^*} + \varepsilon \left[\dfrac{\partial \overline{u}}{\partial \tau^*} + \overline{u} \dfrac{\partial \overline{u}}{\partial \xi^*} + \overline{w} \dfrac{\partial \overline{u}}{\partial z^*} \right] + \dfrac{\partial \overline{p}}{\partial \xi^*} = 0 \\ \varepsilon \left[-F_r \dfrac{\partial \overline{w}}{\partial \xi^*} + \varepsilon \left(\dfrac{\partial \overline{w}}{\partial \tau^*} + \overline{u} \dfrac{\partial \overline{w}}{\partial \xi^*} + \overline{w} \dfrac{\partial \overline{w}}{\partial z^*} \right) \right] + \dfrac{\partial \overline{p}}{\partial z^*} = 0 \\ \overline{w} \left[\xi^*, \varepsilon \overline{\eta}, \tau^* \right] = -F_r \dfrac{\partial \overline{\eta}}{\partial \xi^*} + \varepsilon \left[\dfrac{\partial \overline{\eta}}{\partial \tau^*} + \overline{u} \left(\xi^*, \varepsilon \overline{\eta}, \tau^* \right) \dfrac{\partial \overline{\eta}}{\partial \xi^*} \right] \\ \overline{p} \left[\xi^*, \varepsilon \overline{\eta}, \tau^* \right] = \overline{\eta} \\ \overline{w} \left[\xi^*, -1 + \varepsilon^2 \overline{\theta}, \tau^* \right] = \varepsilon \left[-F_r \dfrac{\partial \overline{\theta}}{\partial \xi^*} + \varepsilon \overline{u} \left(\xi^*, -1 + \varepsilon^2 \overline{\theta}, \tau^* \right) \dfrac{\partial \overline{\theta}}{\partial \xi^*} \right] \end{cases}}$$

7.3 From asymptotic development to fKdV model

7.3.1 Asymptotic development as $\varepsilon \longrightarrow 0$

Incompressibility relation

We are now using the last equations and develop all the functions in power of ε stopping at the second order precision. If we write :

$$\begin{cases} \overline{u}(\xi^*, z^*, \tau^*) = u_0(\xi^*, z^*, \tau^*) + \varepsilon u_1(\xi^*, z^*, \tau^*) + O(\varepsilon^2) \\ \overline{w}(\xi^*, z^*, \tau^*) = w_0(\xi^*, z^*, \tau^*) + \varepsilon w_1(\xi^*, z^*, \tau^*) + O(\varepsilon^2) \end{cases}$$

then, the incompressibility condition becomes :

$$\left[\frac{\partial u_0}{\partial \xi^*} + \frac{\partial w_0}{\partial z^*}\right] + \varepsilon \left[\frac{\partial u_1}{\partial \xi^*} + \frac{\partial w_1}{\partial z^*}\right] + O(\varepsilon^2) = 0.$$

Equations of motion

We introduce also $\overline{p}(\xi^*, z^*, \tau^*) = p_0(\xi^*, z^*, \tau^*) + \varepsilon p_1(\xi^*, z^*, \tau^*) + O(\varepsilon^2)$. We will suppose that the bottom is moving at a critical regime so that $F_r = 1 + \varepsilon\sigma + O(\varepsilon^2)$. The equations of motion now becomes :

$$\begin{cases} \left[-\frac{\partial u_0}{\partial \xi^*} + \frac{\partial p_0}{\partial \xi^*}\right] + \varepsilon\left[-\sigma\frac{\partial u_0}{\partial \xi^*} - \frac{\partial u_1}{\partial \xi^*} + \frac{\partial u_0}{\partial \tau^*} + u_0\frac{\partial u_0}{\partial \xi^*} + w_0\frac{\partial u_0}{\partial z^*} + \frac{\partial p_1}{\partial \xi^*}\right] + O(\varepsilon^2) = 0 \\ \frac{\partial p_0}{\partial z^*} + \varepsilon\left[-\frac{\partial w_0}{\partial \xi^*} + \frac{\partial p_1}{\partial z^*}\right] + O(\varepsilon^2) = 0 \end{cases}$$

Free surface conditions

We especially have to be careful with the boundary conditions. We are going to make a Taylor development around zero for u_0 and for w_0. We also expand asymptotically according to ε the free surface : $\overline{\eta}(\xi^*, \tau^*) = \eta_0(\xi^*, \tau^*) + \varepsilon\eta_1(\xi^*, \tau^*) + O(\varepsilon^2)$. It comes then :

$$\begin{cases} \left(w_0|_{z=0} + \frac{\partial \eta_0}{\partial \xi^*}\right) + \varepsilon\left(\frac{\partial w_0}{\partial z^*}|_{z=0}\eta_0 + w_1|_{z=0} + \sigma\frac{\partial \eta_0}{\partial \xi^*} + \frac{\partial \eta_1}{\partial \xi^*} - \frac{\partial \eta_0}{\partial \tau^*} - u_0|_{z=0}\frac{\partial \eta_0}{\partial \xi^*}\right) + O(\varepsilon^2) = 0 \\ [p_0|_{z=0} - \eta_0] + \varepsilon\left[\eta_0\frac{\partial p_0}{\partial z^*}|_{z=0} + p_1|_{z=0} - \eta_1\right] + O(\varepsilon^2) = 0 \end{cases}$$

Bottom condition

Finally, we expand the bottom asymptotically $\overline{\theta}(\xi^*, \tau^*) = \theta_0(\xi^*, \tau^*) + \varepsilon\theta_1(\xi^*, \tau^*) + O(\varepsilon^2)$ and we also use a Taylor development of u_0 and w_0 around -1. We get the equation :

$$w_0|_{z=0} + \varepsilon\left[w_1|_{z=-1} + \frac{\partial \theta_0}{\partial \xi^*}\right] + O(\varepsilon^2) = 0$$

7.3.2 The emergence of the fKdV model

Solving the system of zero-order approximation

We have the system as follow to integrate :

$$\begin{cases} \dfrac{\partial p_0}{\partial z^*} = 0 \quad \text{and} \quad p_0|_{z=0} = \eta_0 \\ \\ \dfrac{\partial u_0}{\partial \xi^*} = \dfrac{\partial p_0}{\partial \xi^*} \\ \\ \dfrac{\partial w_0}{\partial z^*} = -\dfrac{\partial u_0}{\partial \xi^*} \quad \text{and} \quad w_0|_{z=-1} = 0 \\ \\ w_0|_{z=0} = -\dfrac{\partial \eta_0}{\partial \xi^*} \end{cases}$$

By integrating the first equation using the dynamic boundary condition, we get $p_0 = \eta_0$ for all $z \in [-1, 0]$. Integrating the second equation and assuming that there is no motion and no pressure at infinity, we get also $u_0 = p_0 = \eta_0$ for all $z \in [-1, 0]$. Finally, we get the final relation by integrating the last equation and using the bottom condition $w_0 = -(z^* + 1)\dfrac{\partial \eta_0}{\partial \xi^*}$ compatible with the last relation. Hence, we get :

$$\forall z^* \in [-1, 0], \quad \begin{cases} u_0 = p_0 = \eta_0 \\ \\ w_0 = -(z^* + 1)\dfrac{\partial \eta_0}{\partial \xi^*} \end{cases}$$

Solving the system of first order approximation

We are going to do the same thing. Using the previous results, we have to integrate the system:

$$\begin{cases} \dfrac{\partial p_1}{\partial z^*} = \underbrace{\dfrac{\partial w_0}{\partial \xi^*}}_{-(z^*+1)\frac{\partial^2 \eta_0}{\partial \xi^{*2}}} \quad \text{and} \quad p_1|_{z=0} = \eta_1 - \eta_0 \underbrace{\dfrac{\partial p}{\partial z^*}|_{z=0}}_{0} \\ \\ \dfrac{\partial w_1}{\partial z^*} = -\dfrac{\partial u_1}{\partial \xi^*} = \sigma \underbrace{\dfrac{\partial u_0}{\partial \xi^*}}_{\frac{\partial \eta_0}{\partial \xi^*}} - \underbrace{\dfrac{\partial u_0}{\partial \tau^*}}_{\frac{\partial \eta_0}{\partial \tau^*}} - \underbrace{u_0}_{\eta_0} \underbrace{\dfrac{\partial u_0}{\partial \xi^*}}_{\frac{\partial \eta_0}{\partial \xi^*}} - w_0 \underbrace{\dfrac{\partial u_0}{\partial \xi^*}}_{0} - \dfrac{\partial p_1}{\partial \xi^*} \\ \\ w_1|_{z=-1} = -\dfrac{\partial \theta_0}{\partial \xi^*} \quad \text{and} \quad w_1|_{z=0} = -\underbrace{\dfrac{\partial w_0}{\partial z^*}|_{z=0}}_{-\frac{\partial \eta_0}{\partial \xi^*}} \eta_0 - \sigma \dfrac{\partial \eta_0}{\partial \xi^*} - \dfrac{\partial \eta_1}{\partial \xi^*} + \dfrac{\partial \eta_0}{\partial \tau^*} + \underbrace{u_0|_{z=0}}_{\eta_0} \dfrac{\partial \eta_0}{\partial \xi^*} \end{cases}$$

Integrating the first equation and using the kinamatic condition, we can get p_1 that we inserted into the second relation. Then, we integrate this last between -1 and 0 using the two last boundary conditions. We can see that appears then the equation which is fKdV in a dimensionless form:

$$-2\frac{\partial \eta_0}{\partial \tau^*} + 2\sigma \frac{\partial \eta_0}{\partial \xi^*} - 3\eta_0 \frac{\partial \eta_0}{\partial \xi^*} - \frac{1}{3}\frac{\partial^3 \eta_0}{\partial {\xi^*}^3} = \frac{\partial \theta_0}{\partial \xi^*}$$

Getting back to the dimensional form

We recall the notation : $\xi^* = \dfrac{\xi}{\lambda}$ and $\tau^* = \dfrac{c_0 \varepsilon}{\lambda} t$. Then, we have $\sigma = \dfrac{F_r - 1}{\varepsilon}$. The last equation can then be rewriten after simplifying by λ and multiplying by ε^2 :

$$-\frac{2\varepsilon}{c_0}\frac{\partial \eta_0}{\partial t} + 2\varepsilon \left(F_r - 1\right) \frac{\partial \eta_0}{\partial \xi} - 3\varepsilon^2 \eta_0 \frac{\eta_0}{\partial \xi} - \frac{\lambda^2 \varepsilon^2}{3}\frac{\partial^3 \eta_0}{\partial \xi^3} = \varepsilon^2 \frac{\partial \theta_0}{\partial \xi}$$

Then, we use the approxiamtions made between the parameters to obtain the original scaling of the free surface elevation. This gives $\varepsilon \eta_0 \approx \dfrac{a\eta_0}{h_0} = \dfrac{\zeta_0}{h_0}$ and $\lambda^2 \varepsilon^2 \eta_0 \approx h_0^2 \zeta_0$. For the bottom elevation, we obtain $\varepsilon^2 \theta_0 \approx \dfrac{d\theta_0}{h_0} = 1 + \dfrac{b_0}{h_0}$. After simplifying by h_0, the equation finally became the forced Korteweg and de Vries one for the free surface elevation ζ_0 in the frame of the bottom (ξ, t) near the critical regime $F_r \approx 1$:

$$\boxed{\frac{-2}{c_0}\frac{\partial \zeta_0}{\partial t} + 2(F_r - 1)\frac{\partial \zeta_0}{\partial \xi} - \frac{3}{h_0}\zeta_0 \frac{\partial \zeta_0}{\partial \xi} - \frac{h_0^2}{3}\frac{\partial^3 \zeta_0}{\partial \xi^3} = \frac{\partial b_0}{\partial \xi}}$$

Chapter 8

Computation of the forced Korteweg and de Vries equation

We discretize the following equation with a second order accurate finite difference scheme in space and a Crank-Nicholson scheme in time :

$$\frac{-2}{c_0}\frac{\partial^2\zeta}{\partial\xi\partial t}+2(F_r-1)\frac{\partial^2\zeta}{\partial\xi^2}-\frac{3}{2h_0}\frac{\partial^2\left(\zeta^2\right)}{\partial\xi^2}-\frac{h_0^2}{3}\frac{\partial^4\zeta}{\partial\xi^4}=\frac{\partial^2 b}{\partial\xi^2}.$$

Then, we explain how we decided to treat the non-linearity and the boundary conditions.

8.1 A Crank-Nicholson scheme for time

The fKdV equation is written in the form :

$$\frac{\partial^2\zeta}{\partial t\partial\xi}(\xi,t)=\Psi(\xi,t).$$

Let's choose a uniform time step Δt so that $[0,T]\approx\{t_0,...,t_M\}$ with $t_j=j\Delta t$ for all $j\in\{0,...,M\}$ and $M=\text{Ent}\left[\frac{T}{\Delta t}\right]$. We recall the backward and forward finite difference in time :

$$\begin{cases}\Psi(\xi,t)=\dfrac{\partial^2\zeta}{\partial t\partial\xi}(\xi,t)=\dfrac{\dfrac{\partial\zeta}{\partial\xi}(\xi,t+\Delta t)-\dfrac{\partial\zeta}{\partial\xi}(\xi,t)}{\Delta t}+O(\Delta t^2)\\[2ex]\Psi(\xi,t+\Delta t)=\dfrac{\partial^2\zeta}{\partial t\partial\xi}(\xi,t+\Delta t)=\dfrac{\dfrac{\partial\zeta}{\partial\xi}(\xi,t+\Delta t)-\dfrac{\partial\zeta}{\partial\xi}(\xi,t)}{\Delta t}+O(\Delta t^2)\end{cases}$$

The Crank-Nicholson scheme is a mean of the forward and backward one. Hence, if we add the two previous equations, we get the formula :

$$\frac{\dfrac{\partial\zeta}{\partial\xi}(\xi,t+\Delta t)-\dfrac{\partial\zeta}{\partial\xi}(\xi,t)}{\Delta t}=\frac{1}{2}\left[\Psi(\xi,t+\Delta t)+\Psi(\xi,t)\right].$$

8.2 A finite difference scheme for space

We use a finite difference scheme in space. Let's use the Taylor development of ζ around the point (ξ, t) :

$$\begin{cases} \zeta(\xi+\Delta\xi,t) = \zeta(\xi,t) + \Delta\xi\dfrac{\partial\zeta}{\partial\xi}(\xi,t) + \dfrac{\Delta\xi^2}{2}\dfrac{\partial^2\zeta}{\partial\xi^2}(\xi,t) + \dfrac{\Delta\xi^3}{6}\dfrac{\partial^3\zeta}{\partial\xi^3}(\xi,t) + O(\Delta\xi^4) \\ \zeta(\xi+\Delta\xi,t) = \zeta(\xi,t) - \Delta\xi\dfrac{\partial\zeta}{\partial\xi}(\xi,t) + \dfrac{\Delta\xi^2}{2}\dfrac{\partial^2\zeta}{\partial\xi^2}(\xi,t) - \dfrac{\Delta\xi^3}{6}\dfrac{\partial^3\zeta}{\partial\xi^3}(\xi,t) + O(\Delta\xi^4) \end{cases}$$

Let's add the first equation with the second and we obtain an approximation for the second derivative of second order accurate :

$$\frac{\partial^2\zeta}{\partial\xi^2}(\xi,t) = \frac{\zeta(\xi+\Delta\xi,t) - 2\zeta(x,t) + \zeta(\xi-\Delta\xi,t)}{\Delta\xi^2} + O(\Delta\xi^2).$$

If we now substract the two equations, we obtain an approximation for the first derivative of second order accurate :

$$\frac{\partial\zeta}{\partial\xi}(\xi,t) = \frac{\zeta(\xi+\Delta\xi,t) - \zeta(\xi-\Delta\xi,t)}{2\Delta\xi} + O(\Delta\xi^2).$$

If we iterate this process for the fourth derivative, we can obtain also the fourth order derivative of second order accurate :

$$\begin{aligned} \frac{\partial^4\zeta}{\partial\xi^4}(\xi,t) = & \frac{\dfrac{\partial^2\zeta}{\partial\xi^2}(\xi+\Delta\xi,t) - 2\dfrac{\partial^2\zeta}{\partial\xi^2}(\xi,t) + \dfrac{\partial^2\zeta}{\partial\xi^2}(\xi-\Delta\xi,t)}{\Delta\xi^2} + O(\Delta\xi^2) \\ = & \frac{\zeta(\xi+2\Delta\xi,t) - 4\zeta(\xi+\Delta\xi,t) + 6\zeta(\xi,t) - 4\zeta(\xi-\Delta\xi,t) + \zeta(\xi-2\Delta\xi,t)}{\Delta\xi^4} + O(\Delta\xi^2) \end{aligned}$$

8.3 A discretization of the non-linearity adapted to the scheme

For the nonlinear term, we are going to use a trick which will appear adapted to the Crank-Nicholson scheme.

$$\begin{aligned} \zeta(\xi,t+\Delta t)^2 &= \left[\zeta(\xi,t) + \Delta t\frac{\partial\zeta}{\partial t}(\xi,t) + O(\Delta t^2)\right]^2 \\ &= \zeta(\xi,t)^2 + 2\zeta(x,t)\Delta t\left[\frac{\partial\zeta(\xi,t+\Delta t) - \zeta(\xi,t)}{\Delta t} + O(\Delta t)\right] + O(\Delta t^2) \\ &= 2\zeta(\xi,t)\zeta(\xi,t+\Delta t) - \zeta(\xi,t)^2 + O(\Delta t^2) \end{aligned}$$

Hence, we have a formula for the non-linear term of second order accurate in time:

$$\zeta(\xi,t+\Delta t)\zeta(\xi,t) = \frac{1}{2}\left(\zeta(\xi,t)^2 + \zeta(\xi,t+\Delta t)^2\right) + O(\Delta t^2)$$

8.4 Resolution of the system with a sparse matrix

Let's choose a space step $\Delta\xi$. We will in a first time assume that the numerical space domain is sufficiently large so that the wave can't reach the border and that the bottom is centered in zero. Let's say $[-L, L] \approx \{\xi_0, ..., \xi_{2N}\}$ where $\xi_i = (i-N)\Delta\xi$ and $N = \text{Ent}\left[\dfrac{L}{\Delta\xi}\right]$. Hence, the value of the functions taken at the extra points such as ξ_{-1}, ξ_{-2} or ξ_{N+1}, ξ_{N+2} will be taken to zero as L is sufficiently large. We will use the notation :

$$\forall (i,j) \in \{0, ..., N\} \times \{0, ..., M\}\,,\ \zeta(\xi_i, t_j) = \zeta_i^j.$$

We can now rewrite our system discretized in time and space.

$$\begin{aligned}
-\frac{2}{c_0\Delta t}\left[\frac{\zeta_{i+1}^{j+1}-\zeta_{i-1}^{j+1}}{2\Delta\xi} - \frac{\zeta_{i+1}^{j}-\zeta_{i-1}^{j}}{2\Delta\xi}\right] + 2\left(F_r-1\right)\frac{1}{2}\left[\frac{\zeta_{i+1}^{j+1}-2\zeta_{i}^{j+1}+\zeta_{i-1}^{j+1}}{\Delta\xi^2} + \frac{\zeta_{i+1}^{j}-2\zeta_{i}^{j}+\zeta_{i-1}^{j}}{\Delta\xi^2}\right]& \\
-\frac{3}{2h_0}\left[\frac{\zeta_{i+1}^{j+1}\zeta_{i+1}^{j}-2\zeta_{i}^{j+1}\zeta_{i}^{j}+\zeta_{i-1}^{j+1}\zeta_{i-1}^{j}}{\Delta\xi^2}\right]& \\
-\frac{h_0^2}{3}\frac{1}{2}\left[\frac{\zeta_{i+2}^{j+1}-4\zeta_{i+1}^{j+1}+6\zeta_{i}^{j+1}-4\zeta_{i-1}^{j+1}+\zeta_{i-2}^{j+1}}{\Delta\xi^4} + \frac{\zeta_{i+2}^{j}-4\zeta_{i+1}^{j}+6\zeta_{i}^{j}-4\zeta_{i-1}^{j}+\zeta_{i-2}^{j}}{\Delta\xi^4}\right]& \\
=& \\
\frac{b_{i+1}-2b_i+b_{i-1}}{\Delta\xi^2}&
\end{aligned}$$

We can rearrange this in a system of the form $A(\zeta^j).\zeta^{j+1} = B(\zeta^j)$. Hence, we have a system of N equations to solve at each time step. The matrix A is sparse as only the five diagonals of the center are non zeros. After rearrranging the term we get the following system of N equations.

$$
\begin{aligned}
\left(-\frac{h_0^2}{6\Delta\xi^4}\right)\zeta_{i+2}^{j+1} \quad & + \quad \left(-\frac{1}{c_0\Delta t\Delta\xi}+\frac{F_r-1}{\Delta\xi^2}-\frac{3\zeta_{i+1}^j}{2h_0\Delta\xi^2}+\frac{2h_0^2}{3\Delta\xi^4}\right)\zeta_{i+1}^{j+1} \\
& + \quad \left(-\frac{2\left(F_r-1\right)}{\Delta\xi^2}+\frac{6\zeta_i^j}{2h_0\Delta\xi^2}-\frac{h_0^2}{\Delta\xi^4}\right)\zeta_i^{j+1} \\
& + \quad \left(\frac{1}{c_0\Delta t\Delta\xi}+\frac{F_r-1}{\Delta\xi^2}-\frac{3\zeta_{i-1}^j}{2h_0\Delta\xi^2}+\frac{2h_0^2}{3\Delta\xi^4}\right)\zeta_{i-1}^{j+1} \\
& + \quad \left(-\frac{h_0^2}{6\Delta\xi^4}\right)\zeta_{i-2}^{j+1} \\
& = \quad \frac{b_{i+1}-2b_i+b_{i-1}}{\Delta\xi^2} \\
& + \quad \left(\frac{h_0^2}{6\Delta\xi^2}\right)\zeta_{i+2}^j \\
& - \quad \left(\frac{1}{c_0\Delta t\Delta\xi}+\frac{F_r-1}{\Delta\xi^2}+\frac{2h_0^2}{3\Delta\xi^4}\right)\zeta_{i+1}^j \\
& + \quad \left(\frac{2\left(F_r-1\right)}{\Delta\xi^2}+\frac{h_0^2}{\Delta\xi^4}\right)\zeta_i^j \\
& + \quad \left(\frac{1}{c_0\Delta t\Delta\xi}-\frac{F_r-1}{\Delta\xi^2}-\frac{2h_0^2}{3\Delta\xi^4}\right)\zeta_{i-1}^j \\
& + \quad \left(\frac{h_0^2}{6\Delta\xi^4}\right)\zeta_{i-2}^j
\end{aligned}
$$

8.5 A filter to assume the correct boundary conditions

Although this scheme of the forced KdV equation seems to be quite simple and stable for big time step, it assumes that the waves produced by the moving bottom do not reach the boundaries of the computational domain. Therefore, if we want to simulate for a long time the phenomenon, a sufficiently large domain must be chosen while a reasonable computation time needs the smallest one.

In order to solve this problem, we decided to apply a filter $\mathcal{F}$ to the solution ζ at each time step. The filter is zero at the left boundary $\xi = -L$ and one after $\xi = -L + 20$. A cosinus profile is then used to link the two lines.

$$\forall \xi \in [-L, L]\,, \quad \mathcal{F}(\xi) = \frac{\mathbf{1}_{[-L,-L+20]}(\xi)}{2}\left(1 + \cos\left(\pi \frac{\xi + L - 20}{20}\right)\right) \quad + \quad \mathbf{1}_{]-L+20,L]}(\xi)$$

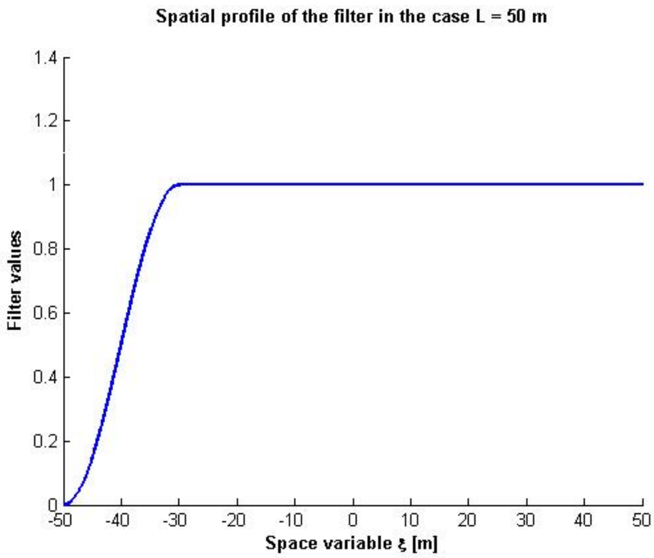

Figure 8.1: *The spatial profile of the filter.*

At each time step, we multiply ζ by $\mathcal{F}$ in order to kill the cnoidal waves. It turns out that this procedure works well without changing the rest of the free surface elevation profile. Therefore, we have a scheme that allow big time step with a small computational domain.

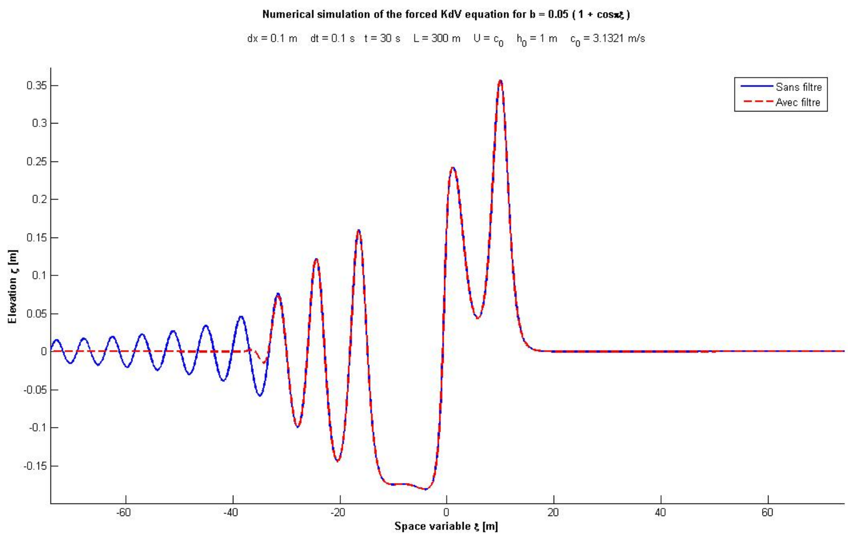

Figure 8.2: *The filter does not interact at all after a long time with the elevation.*

Chapter 9

Validation of the numerical resolution of the forced KdV equation

We first are going to check if the code is working by using the exact solution of the KdV equation. Then, we will compare its performance with other algorithms implemented on Matlab.

9.1 The exact analytic form of the solitary waves generated

We want to find one of the exact solution of the Korteweg and de Vries equation ($F_r = 1$ et $b = 0$) that is a solitary wave :

$$\begin{cases} \dfrac{1}{c_0}\dfrac{\partial \zeta}{\partial t} + \dfrac{3}{2h_0}\zeta\dfrac{\partial \zeta}{\partial \xi} + \dfrac{h_0^2}{6}\dfrac{\partial^3 \zeta}{\partial \xi^3} = 0 \\ \lim_{\xi\to\pm\infty} \zeta(\xi, t) = 0 \\ \lim_{\xi\to\pm\infty} \dfrac{\partial \zeta}{\partial \xi}(\xi, t) = 0 \\ \lim_{\xi\to\pm\infty} \dfrac{\partial^2 \zeta}{\partial \xi^2}(\xi, t) = 0 \end{cases}$$

As we seek for a solitary wave, we search for a travelling wave of the form $\zeta(\xi, t) = f(\chi) = f(\xi - ct)$ where c is the speed of the wave and $\chi = \xi - ct$ is the charateristic on which the waves travels. The equation became then :

$$-\frac{c}{c_0}\frac{df}{d\chi}(\chi) + \frac{3}{4h_0}\frac{d(f^2)}{d\chi}(\chi) + \frac{h_0^2}{6}\frac{d^3 f}{d\chi^3}(\chi) = 0$$

We integrate this equation using the zero boundary conditions and we then multiply again by f' the equation obtained so as to integrate again. We then have :

$$f'^2(\chi) = -\frac{3}{h_0^3} f^2(\chi) \left[f - \underbrace{\frac{2h_0 c}{c_0}}_{:=a} \right]$$

The right member of this relation is a trinom that must be positive. This impose that the wave elevation will be between 0 and $a = \frac{2h_0 c}{c_0}$. We denote χ_0 the point where the wave will achieve its maximum. Hence we have $f(\chi_0) = a$. We can pass to the root and now separate the variable and integrate between χ_0 and χ. We get :

$$h_0 \sqrt{\frac{h_0}{3}} \int_{\chi_0}^{\chi} \frac{f'(\varsigma)}{f(\varsigma)\sqrt{a - f(\varsigma)}} d\varsigma = \int_{\chi_0}^{\chi} 1 d\varsigma \;\Leftrightarrow\; -2h_0 \sqrt{\frac{h_0}{3a}} \tanh^{-1}\left(\sqrt{\frac{a - f(\chi)}{a}} \right) = \chi - \chi_0$$

$$\Leftrightarrow\; f(\chi) = a \ \mathrm{sech}^2 \left[\frac{1}{2h_0} \sqrt{\frac{3a}{h_0}} (\chi - \chi_0) \right]$$

To get back to the wave elevation ζ let's say that χ_0 is the point ξ_0 at $t = 0$ so that $f(\chi_0) = a = \zeta(\xi_0, 0)$. We then have the formula remembering the relation $2ch_0 = c_0 a$ [1] :

$$\zeta(\xi, t) = a \ \mathrm{sech}^2 \left[\frac{1}{2h_0} \sqrt{\frac{3a}{h_0}} \left(\xi - \xi_0 - c_0 \frac{a}{2h_0} t \right) \right]$$

Finally, in order to get back to the physical space variable x, we recall that we assume during the fKdV model derivation that $\xi = x - c_0 t$. Let's choose ξ_0 the point x_0 at $t = 0$. As above, we have $\zeta(\xi_0) = \zeta(x_0) = a$. Rearranging the term, we obtain the final formulation for the wave elevation ζ :

$$\boxed{\zeta(x, t) = a \ \mathrm{sech}^2 \left[\frac{1}{2h_0} \sqrt{\frac{3a}{h_0}} \left(x - x_0 - c_0 \left(1 + \frac{a}{2h_0} \right) t \right) \right]}$$

Therefore, we can see that the wave amplitude is the only degree of freedom in the formulation of the wave elevation. In particular, the speed of the wave is only determined by the amplitude with the relation found by Scott Russell :

$$c = c_0 \left(1 + \frac{a}{2h_0} \right) \approx \sqrt{g(h_0 + a)}$$

[1]We recall here that $sech = \frac{1}{cosh} = \frac{2}{exp() + exp(-\ )}$ and the identity $1 - tanh^2 = \frac{1}{cosh^2} = sech^2$.

9.2 Description of the KdV algorithm

I implemented four different discretizations of the folowing dimensional KdV equation and compared them to its analytical solution found in the previous section.

$$\begin{cases} \dfrac{1}{c_0}\dfrac{\partial \zeta}{\partial t} + \dfrac{3}{2h_0}\zeta\dfrac{\partial \zeta}{\partial \xi} + \dfrac{h_0^2}{6}\dfrac{\partial^3 \zeta}{\partial \xi^3} = 0 \\ \zeta(x,0) = a \ \mathrm{sech}^2\left(\dfrac{1}{2h_0}\sqrt{\dfrac{3a}{h_0}}x\right) \end{cases}$$

All the Matlab programs can be found in the last part of the report entitled Annexes. It mainly consists in :

- the historical scheme developped by Zabusky and Kruskal [12] that uses a finite difference method and a leap-frog discretization ;
- a scheme developped by Fornberg and Whitham [19] that involves a clever disctretization using the Fourier transform and a leap-frog method ;
- a scheme implemented by Trefethen [15] that employs a fourth order Runge-Kutta algorithm and the method of integrating factor in the pseudo-spectral discretization ;
- the scheme used in the optimization algorithm [1] and described above that discretizes the derivated KdV equation with a Cranck-Nicholson and a finite difference method.

We give below the main data used in the four algorithms. First, we will confront the L^2-norm error between the analytical solution ζ_{ref} and each numerical solution ζ :

$$\forall j \in \{1, \ldots M\}, \ \|\zeta - \zeta_{ref}\|(t_j) = \sqrt{\frac{1}{2N+1}\sum_{i=0}^{2N}\left(\zeta_i^j - \zeta_{ref_i}^j\right)^2}$$

Then, we will compare the running time, the stability condition and the treatment of boundary conditions for each algorithm.

Parameter	**Notation**	**Value**
Computational domain	$[-L, L]$	$[-12 \text{ m}, 12 \text{ m}]$
Water depth	h_0	1 m
Speed of reference	$c_0 = \sqrt{gh_0}$	3.1 m.s^{-1}
Amplitude of the wave	a	0.2 m
Speed of the wave in the bottom frame	$c = c_0 a \backslash 2h_0$	0.3 m.s^{-1}
Space step	$\Delta\xi$	0.1 m
Computational time	T	10 s
Time step	Δt	depends on the CFL condition

9.3 Performance of the KdV algorithms

First, the error commited via the three discretizations described for comparison are of 4 order of magnitude. Our algorithm is of the same order but is twice less precise. However, this is still physically negligeable. The sudden increase of the error for our discretization comes from the fact that the soliton interferes a bit with the zero boundary condition hypothesis.

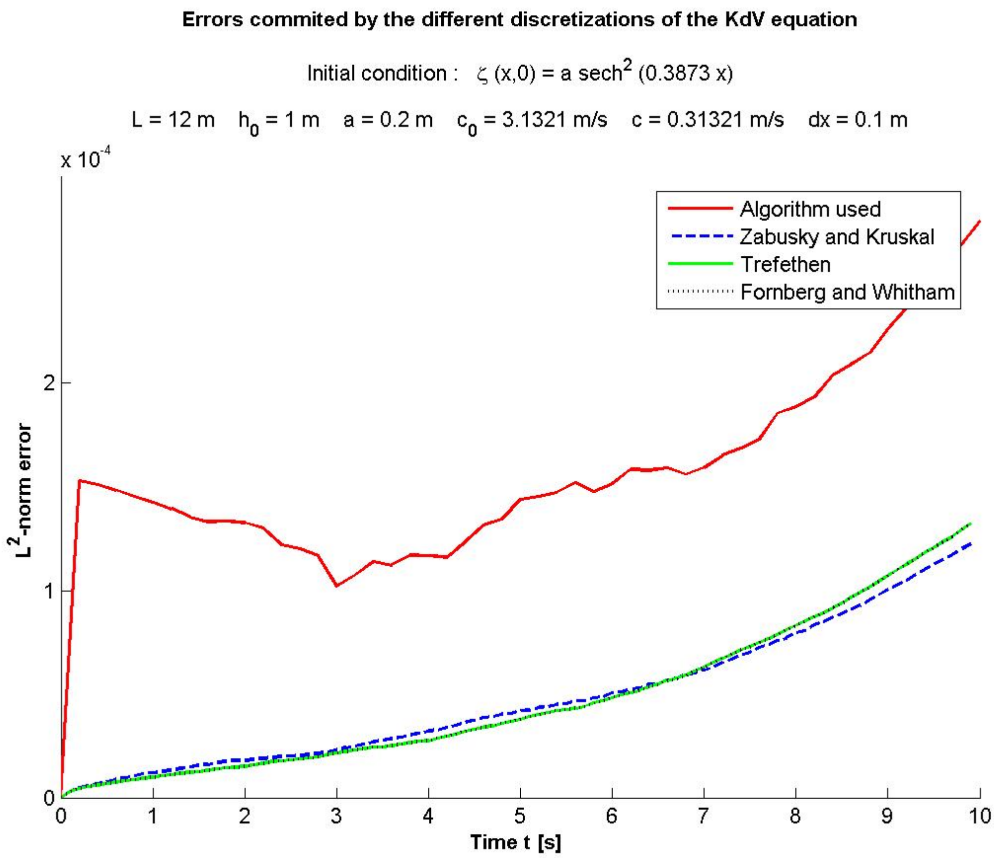

Figure 9.1: *Our algorithm seems to perform badly compared to other methods. Indeed, the L^2-norm error is the double of others one.*

However, the loss of precision is compensated by a significant gain of time in the running process. Indeed, all the previous algorithms are subject to a strong stability condition of the type $\Delta t = O\left(\Delta x^3\right)$ whereas the algorithm chosen has not. This really justifies the choice made: a fast algorithm that will be incorporated in an optimization loop.

Algorithm	CFL condition	Value	Running time
Our discretization	none	$\Delta t = 0.1$ s	0.4 s
Zabusky and Kruskal	$\Delta t \leqslant \dfrac{\Delta\xi^3}{4 + a\Delta\xi^2}$	$\Delta t = 2.5\ 10^{-4}$ s	2.7 s
Trefethen	$\Delta t \leqslant 10\dfrac{3\Delta\xi^3}{2\pi^2}\left(\dfrac{\pi}{L}\right)^3$	$\Delta t = 2.7\ 10^{-5}$ s	14 min 13 s
Fornberg and Whitham	$\Delta t \leqslant \dfrac{3\Delta\xi^3}{2\pi^2}\left(\dfrac{\pi}{L}\right)^3$	$\Delta t = 2.7\ 10^{-6}$ s	15 min 28 s

Finally, when a second member is added in the KdV equation, all the algorithms presented became less efficient than the one chosen. Indeed, the forcing term seems to really introduce numerical instabilities. Spectral methods and leap-frog scheme is known for enhancing them [19].

To conclude, the Fourier methods always suppose that functions are periodic on the spatial domain. This imposes a large computational domain or the introduction of a filter to ensure the zero boundary conditions. The last suggestion really reduces the running time but has not been used for the comparison between the different algorithm performances : a sufficiently large spatial domain was chosen in order to ensure that the boundaries were not interfering with the solution.

Part IV

The shape optimization problem

Chapter 10

Formulation of the design optimization problem

First, we will clearly describe the optimization problem in mathematical term : maximizing a functionnal in an admissible set of solutions. Then, we will use the adjoint formulation of the problem to evaluate the gradient of the functionnal.

10.1 Find the bottom that creates the highest wave

With the forced KdV equation, it has been shown that solitons can be generated from a moving bottom underwater. in order to optimize the wave maker, one interesting degree of freedom is the choice of shape.

This design optimization process has been the main subject of this report : under some conditions, is it possible to find numerically an optimal bottom that could create the highest wave ? Using the fKdV model, how does the shape affect the wave height ?

Therefore, the goal here is to find $\max a\,(b)$ where the amplitude a of the generated soliton depends on the bottom b and in particular on its shape.

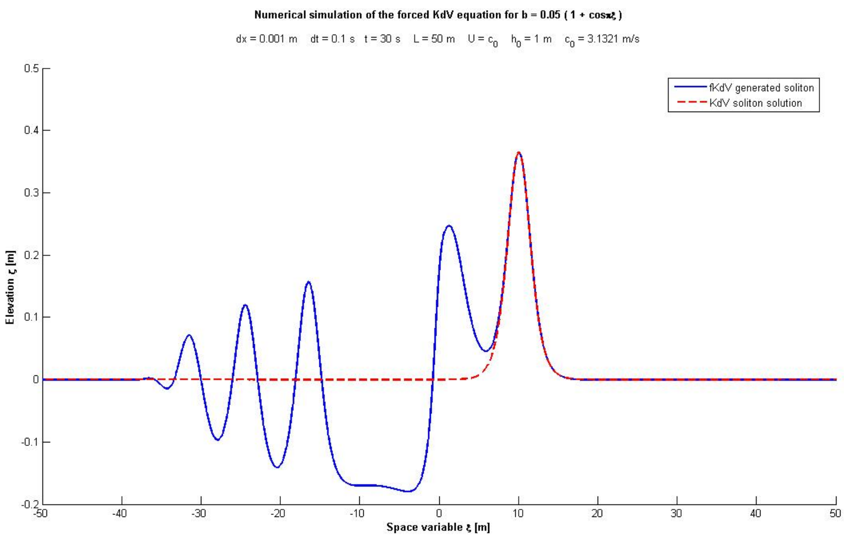

Figure 10.1: *The fKdV equation generates an autonomous solitary wave only determined by its amplitude a which depends on the bottom b. In a first approach, we want to find* $\max_b a$.

In [20], Wu studied how the energy of the forcing disturbance is shared out. He obtained relations between the period of soliton generation τ, the energy $\mathcal{E}$ received by the trailing wavetrain from the bottom and the amplitude a of the solitary wave :

$$\tau = \frac{64}{(3a)^{3\backslash 2}} \quad \text{and} \quad \frac{d\mathcal{E}}{dt} = \left(\frac{a}{2}\right)^3.$$

Therefore, maximizing the amplitude of the solitary wave is equivalent to maximize the whole surface elevation and will reduce the period of generation. With this remark in mind, we decided to choose a simpler functional to maximize :

$$J(b) = \frac{1}{2}\int_0^T \int_{-\infty}^{+\infty} \zeta_b(\xi,t)^2 d\xi dt.$$

where b is the bottom and ζ_b is the corresponding surface elevation solution of the fKdV equation with the initial condition $\zeta_b(\xi, 0) = 0$.

Finally, an admissible set of solutions has to be chosen. We decided to impose the bottom to have its support in $[-S, S]$ with $S > 0$, to be positive and to have its L^2-norm bounded by a fixed constant M. It means that the set of admissible bottoms is :

$$\mathcal{B} = \left\{ b \in L^2(\mathbb{R}), \quad \text{supp}\, b \subseteq [-S, S], \quad b \geqslant 0 \ \text{ and } \int_{\mathbb{R}} b^2(\xi)\, d\xi \leqslant M^2 \right\}.$$

Hence, the problem is to find $\max\limits_{b\in\mathcal{B}} J(b) = -\min\limits_{b\in\mathcal{B}} -J(b)$.

10.2 The dual continuous approach to evaluate the gradient of the functionnal

Let's recall here the problem in term of minimum where the support condition has been inserted into the partial differential equation. We want to find $\min\limits_{b\in\mathcal{B}} J(b)$ with :

$$\left\{ \begin{array}{l} J(b) = -\dfrac{1}{2}\displaystyle\int_0^T \int_{-\infty}^{+\infty} \zeta_b(\xi,t)^2 d\xi dt \\[2ex] \mathcal{B} = \left\{ b \in L^2(\mathbb{R}), \quad b \geqslant 0 \ \text{ and } \displaystyle\int_{\mathbb{R}} b^2(\xi)\, d\xi \leqslant M^2 \right\} \end{array} \right.$$

where ζ_b is solution of the partial differential equation

$$\left\{ \begin{array}{l} -\dfrac{1}{c_0}\dfrac{\partial \zeta_b}{\partial t} + \left[\left(\dfrac{U}{c_0} - 1\right) - \dfrac{3}{2h_0}\zeta_b\right]\dfrac{\partial \zeta_b}{\partial \xi} - \dfrac{h_0^2}{6}\dfrac{\partial^3 \zeta_b}{\partial \xi^3} = \dfrac{1}{2}\dfrac{\partial\left(b\mathbf{1}_{[-S,S]}\right)}{\partial \xi} \\[2ex] \zeta_b(\xi, 0) = 0 \end{array} \right.$$

In order to solve this problem, a descent method needs the gradient of the functionnal. Therefore, a formulation of an adjoint problem is needed : the sensitivity of the functionnal will be ruled by an other partial differential equation.

10.2.1 Searching for a shape derivative of the functionnal

We want to find the Gâteau derivative of the functional so as to find our minimum with a gradient method. We make a small perturbation of the bottom b in a direction h. Thus, we have :

$$\begin{aligned}\frac{J(b+\varepsilon h)-J(b)}{\varepsilon} &= -\frac{1}{2\varepsilon}\int_0^T\int_{-\infty}^{+\infty}\left[\zeta_{b+\varepsilon h}^2-\zeta_b^2\right]dxdt \\ &= -\frac{1}{2}\int_0^T\int_{-\infty}^{+\infty}\frac{\zeta_{b+\varepsilon h}-\zeta_b}{\varepsilon}\left[\zeta_{b+\varepsilon h}+\zeta(b)\right]dxdt\end{aligned}$$

Assume that we can pass to the limit when $\varepsilon \longrightarrow 0$. Denoting $\langle d\zeta(b)|h\rangle = \overline{\zeta}$, then we obtain :

$$\langle dJ(b)|h\rangle = -\int_0^T\int_{-\infty}^{+\infty}\overline{\zeta}\zeta dxdt$$

10.2.2 The partial differential equation followed by $\overline{\zeta}$

Let's choose $\varepsilon > 0$ and h a direction of perturbation for the bottom b. The two wave elevations $\zeta_{b+\varepsilon h}$ and ζ_b are respectively solution of the partial differential equations :

$$\begin{cases} -\dfrac{1}{c_0}\dfrac{\partial\zeta_b}{\partial t}+\left[\left(\dfrac{U}{c_0}-1\right)-\dfrac{3}{2h_0}\zeta_b\right]\dfrac{\partial\zeta_b}{\partial\xi}-\dfrac{h_0^2}{6}\dfrac{\partial^3\zeta_b}{\partial\xi^3}=\dfrac{1}{2}\dfrac{\partial\left(b\mathbf{1}_{[-S,S]}\right)}{\partial\xi} \\ \zeta_b(\xi,0)=0 \end{cases}$$

$$\begin{cases} -\dfrac{1}{c_0}\dfrac{\partial\zeta_{b+\varepsilon h}}{\partial t}+\left[\left(\dfrac{U}{c_0}-1\right)-\dfrac{3}{2h_0}\zeta_{b+\varepsilon h}\right]\dfrac{\partial\zeta_{b+\varepsilon h}}{\partial\xi}-\dfrac{h_0^2}{6}\dfrac{\partial^3\zeta_{b+\varepsilon h}}{\partial\xi^3}=\dfrac{1}{2}\dfrac{\partial\left(b+\varepsilon h\mathbf{1}_{[-S,S]}\right)}{\partial\xi} \\ \zeta_{b+\varepsilon h}(\xi,0)=0 \end{cases}$$

Let's substrack the last with the first equation and divide by ε and then pass to the limit when ε tends to 0. We have to take care with the non-linear term in this way :

$$\zeta_{b+\varepsilon h}\frac{\partial\zeta_{b+\varepsilon h}}{\partial\xi}-\zeta_b\frac{\partial\zeta_b}{\partial\xi}=\left[\zeta_{b+\varepsilon h}-\zeta_b\right]\frac{\partial\zeta_{b+\varepsilon h}}{\partial\xi}+\zeta_b\frac{\partial}{\partial\xi}\left[\zeta_{b+\varepsilon h}-\zeta_b\right]$$

Hence, we get the partial differential equation for $\overline{\zeta}$:

$$\begin{cases} -\dfrac{1}{c_0}\dfrac{\partial\overline{\zeta}}{\partial t}+\left(\dfrac{U}{c_0}-1\right)\dfrac{\partial\overline{\zeta}}{\partial\xi}-\dfrac{3}{2h_0}\left[\overline{\zeta}\dfrac{\partial\zeta_b}{\partial\xi}+\zeta_b\dfrac{\partial\overline{\zeta}}{\partial\xi}\right]-\dfrac{h_0^2}{6}\dfrac{\partial^3\overline{\zeta}}{\partial\xi^3}=\dfrac{1}{2}\dfrac{\partial\left(h\mathbf{1}_{[-S,S]}\right)}{\partial\xi} \\ \overline{\zeta}(\xi,0)=0 \end{cases}$$

10.2.3 The emergence of the adjoint

Let's now multiply the last equation by an unknown function v and let's integrate by part enough time to put the derivatives of $\overline{\zeta}$ on v. After a tedious calculus we obtain :

$$\int_0^T \int_{-\infty}^{+\infty} \overline{\zeta} \;\left[\frac{1}{c_0}\frac{\partial v}{\partial t} - \left(\frac{U}{c_0} - 1\right)\frac{\partial v}{\partial \xi} + \frac{3}{2h_0}\zeta_b \frac{\partial v}{\partial \xi} + \frac{h_0^2}{6}\frac{\partial^3 v}{\partial \xi^3}\right] = -\frac{1}{2}\int_0^T \int_{-\infty}^{+\infty} h\mathbf{1}_{[-1,1]}\frac{\partial v}{\partial \xi}$$
$$-\frac{1}{c_0}\int_{-\infty}^{+\infty} \left[v\overline{\zeta}\,\right]_0^T$$
$$\underbrace{+\int_0^T \left[\frac{1}{2}hv\mathbf{1}_{[-S,S]} - \left(\frac{U}{c_0} - 1\right)v\overline{\zeta} + \frac{3}{2h_0}v\overline{\zeta}\zeta_b + \frac{h_0^2}{6}\left(v\frac{\partial^2 \overline{\zeta}}{\partial \xi^2} - \frac{\partial v}{\partial \xi}\frac{\partial \overline{\zeta}}{\partial \xi} + \overline{\zeta}\frac{\partial^2 v}{\partial \xi^2}\right)\right]_{-\infty}^{+\infty}}_{=0}$$

We now can see the adjoint equation appearing in the left member of the equation. We want to calculate

$$\langle dJ(b)|h\rangle = -\int_0^T \int_{-\infty}^{+\infty} \overline{\zeta}\zeta dxdt.$$

and if we define v as the solution of this adjoint partial differential equation :

$$\boxed{\begin{cases} \dfrac{1}{c_0}\dfrac{\partial v}{\partial t} - \left(\dfrac{U}{c_0} - 1\right)\dfrac{\partial v}{\partial \xi} + \dfrac{3}{2h_0}\zeta_b\dfrac{\partial v}{\partial \xi} + \dfrac{h_0^2}{6}\dfrac{\partial^3 v}{\partial \xi^3} = \zeta_b \\ v(\xi, T) = 0 \end{cases}}$$

then, if we remember that we worked for imposing the zero boundary conditions via a filter, our last relation will simplify into :

$$\langle dJ(b)|h\rangle = -\int_0^T \int_{-\infty}^{+\infty} \overline{\zeta}\zeta dxdt = \frac{1}{2}\int_0^T \int_{-\infty}^{+\infty} h\mathbf{1}_{[-S,S]}\frac{\partial v}{\partial \xi}dxdt = \left\langle \frac{1}{2}\mathbf{1}_{[-S,S]}\int_0^T \frac{\partial v}{\partial \xi}|h\right\rangle$$

In other word, we have found the shape derivative of our functionnal :

$$\boxed{dJ(b)|_\xi = \frac{1}{2}\mathbf{1}_{[-S,S]}(\xi)\int_0^T \frac{\partial v}{\partial \xi}(\xi, t)dt.}$$

where v is solution of the adjoint problem depending on ζ_b solution of the fKdV equation.

10.3 The computation of the adjoint problem

In computationnal fluids dynamics, a huge amount of work has been done to obtain very efficient algorithms for solving very difficult partial differential equations such as the ones in the full Navier Stokes model.

However, with the development of design optimization, new types of equations emerged from adjoint formulations of problems. Moreover, the computation of these equations turns out to be very difficult. The lack of physical intuition makes it even more difficult.

We decided to use exactly the same scheme for the adjoint equation than for the fKdV one. This means that a derivation according to the space variable is applied on the adjoint equation. Then, the space discretization uses the classical finite difference method and the time one employs the Crank-Nicholson scheme.

$$
\begin{aligned}
\left(-\frac{h_0^2}{12\Delta\xi^4}\right) v_{i-2}^j \quad & + \quad \left(-\frac{1}{2c_0\Delta t\Delta\xi} + \frac{F_r-1}{2\Delta\xi^2} + \frac{3\left(\zeta_{i+1}^j - \zeta_{i-1}^j\right)}{16h_0\Delta\xi^2} - \frac{3\zeta_i^j}{8h_0\Delta\xi^2} + \frac{h_0^2}{3\Delta\xi^4}\right) v_{i-1}^j \\
& + \quad \left(-\frac{F_r-1}{\Delta\xi^2} + \frac{3\zeta_i^j}{2h_0\Delta\xi^2} - \frac{h_0^2}{2\Delta\xi^4}\right) v_i^j \\
& + \quad \left(\frac{1}{2c_0\Delta t\Delta\xi} + \frac{F_r-1}{\Delta 2\xi^2} - \frac{3\left(\zeta_{i+1}^j - \zeta_{i-1}^j\right)}{16h_0\Delta\xi^2} - \frac{3\zeta_i^j}{4h_0\Delta\xi^2} + \frac{h_0^2}{3\Delta\xi^4}\right) v_{i+1}^j \\
& + \quad \left(-\frac{h_0^2}{12\Delta\xi^4}\right) v_{i+2}^j \\
& = \quad -\frac{\zeta_{i+1}^{j+1} + \zeta_{i+1}^j - \zeta_{i-1}^{j+1} - \zeta_{i-1}^j}{4\Delta\xi} \\
& + \quad \left(\frac{h_0^2}{12\Delta\xi^4}\right) v_{i-2}^{j+1} \\
& - \quad \left(\frac{1}{2c_0\Delta t\Delta\xi} + \frac{F_r-1}{2\Delta\xi^2} + \frac{3\left(\zeta_{i+1}^{j+1} - \zeta_{i-1}^{j+1}\right)}{16h_0\Delta\xi^2} - \frac{3\zeta_i^{j+1}}{4h_0\Delta\xi^2} + \frac{h_0^2}{3\Delta\xi^4}\right) v_{i-1}^{j+1} \\
& + \quad \left(\frac{F_r-1}{\Delta\xi^2} - \frac{3\zeta_i^{j+1}}{2h_0\Delta\xi^2} + \frac{h_0^2}{2\Delta\xi^4}\right) v_i^{j+1} \\
& + \quad \left(\frac{1}{2c_0\Delta t\Delta\xi} - \frac{F_r-1}{2\Delta\xi^2} + \frac{3\left(\zeta_{i+1}^{j+1} - \zeta_{i-1}^{j+1}\right)}{16h_0\Delta\xi^2} + \frac{3\zeta_i^{j+1}}{4h_0\Delta\xi^2} - \frac{h_0^2}{3\Delta\xi^4}\right) v_{i+1}^{j+1} \\
& + \quad \left(\frac{h_0^2}{12\Delta\xi^4}\right) v_{i+2}^{j+1}
\end{aligned}
$$

However, in the adjoint equation, the evolution of time is reversed, so we must solve a system of the form $A\left(\zeta^{j}\right)v^{j}=B\left(v^{j+1},\zeta^{j+1},\zeta^{j}\right)$ where the unknown vector is v^{j}.

Calculations are not detailled here but they are exactly the same as for the forced KdV equation and the algorithm turns out to work well. The same filter on the left and right side is employed to ensure a zero boundary condition at every time. This allows a small spatial domain and reduces the running time.

$$\begin{aligned}\forall\xi\in[-L,L]\,,\quad \mathcal{F}(\xi) &= \frac{\mathbf{1}_{[-L,-L+20]}(\xi)}{2}\left(1+\cos\left(\pi\frac{\xi+L-20}{20}\right)\right) + \mathbf{1}_{]-L+20,L]}(\xi)\\ &+ \frac{\mathbf{1}_{[L-20,L]}(\xi)}{2}\left(1+\cos\left(\pi\frac{\xi-L+20}{20}\right)\right)\end{aligned}$$

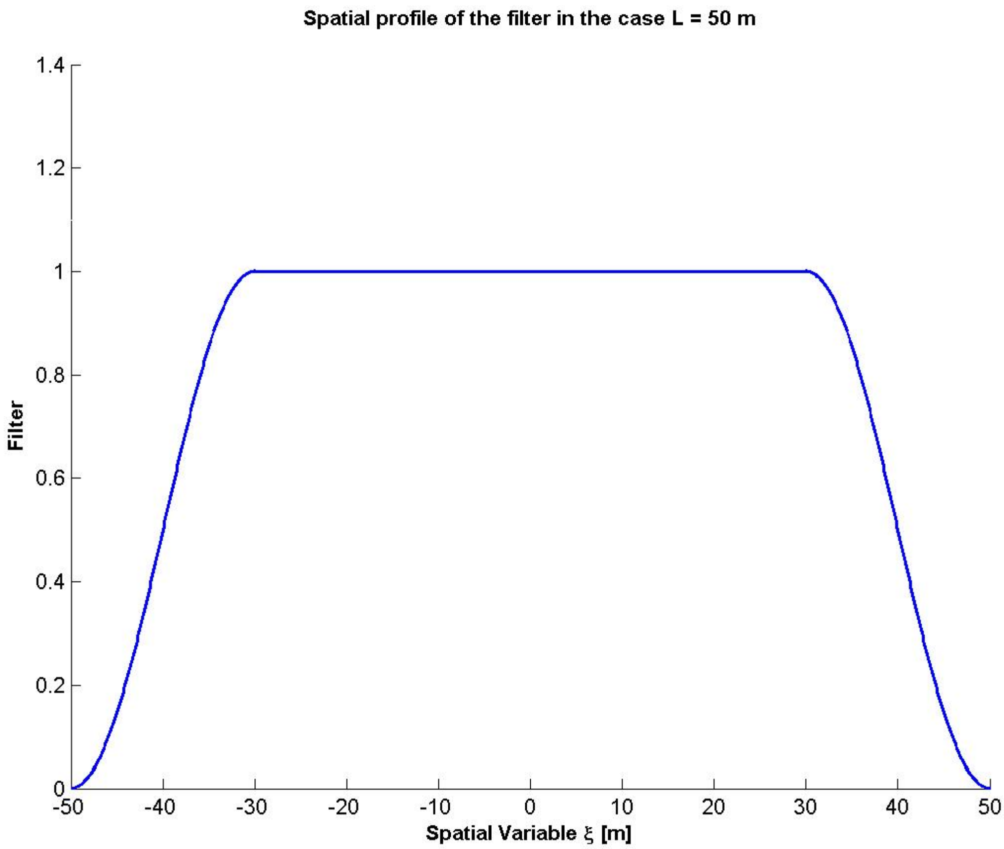

Figure 10.2: *Spatial profile of the adjoint filter in the case* $L=50$ m.

On the figure below, we can notice that the filter interacts a bit with the adjoint solution v after a long time whereas this was not the case in the fKdV equation. However, we will assume this difference negligible in order to perform better the optimization algorithm.

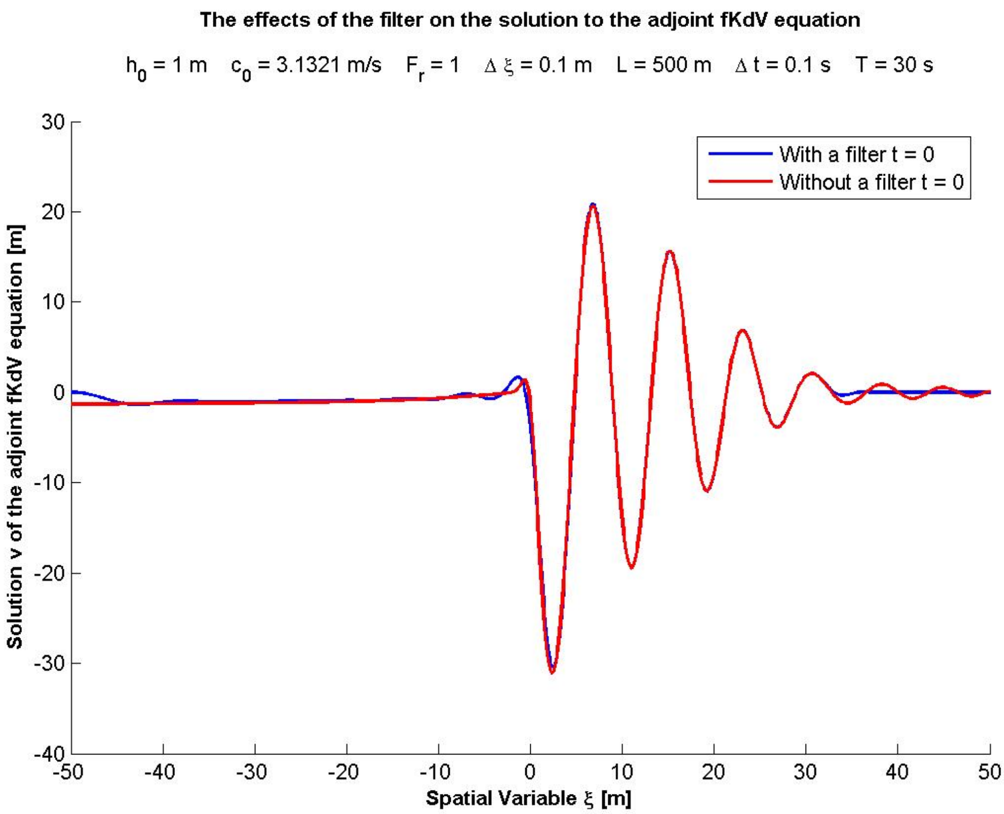

Figure 10.3: *We compared the adjoint solution obtained with a* $[-50\text{ m}, 50\text{ m}]$*-filter on fKdV and the adjoint, with the one taken on a large computationnal domain* $L = 500m$ *without any filter.*

Chapter 11

The numerical approach using Usawa algorithm

Now that we can evaluate the derivative of our functionnal $dJ(b)$ solving each time two partial differential equations, let's see the numerical point of view. After the discretization, b is not anymore a function in infinite dimension space. Usawa algorithm is a way to replace a constrainted minimization problem by a sequence of unconstrainted minimization problem.

11.1 Introduction of the Lagrangian

Let's denote $\Delta\xi$ the step discretization. As the support of b is included in $[-S, S]$, b will be characterized by $R = \text{Ent}\left[\frac{2S}{\Delta\xi}\right] + 1$ points where Ent[.] denotes here the floor function. Thus, $b = (b_0, ..., b_{R-1})^T \in \mathbb{R}^R$. We then introduce the lagrangian :

$$\begin{array}{rccl} \mathcal{L}: & \mathbb{R}^R \times (\mathbb{R}_+)^{R+1} & \longrightarrow & \mathbb{R} \\ & (b, \lambda) & \longmapsto & J(b) - \displaystyle\sum_{n=0}^{R-1} \lambda_n b_n + \lambda_R \left(\int_{-S}^{S} b^2 - M^2 \right) \end{array}$$

where the integral has been approximated by the Simson rule[1].

The theory of the lagrangian then gives us a way to solve equivalently our minimization problem [6].

$$\text{Find} \qquad \inf_{b \geqslant 0,\ \int b^2 \leqslant M^2} J(b) \Longleftrightarrow \begin{cases} \text{Define} & G(\lambda) = \displaystyle\inf_{b \in \mathbb{R}^R} \mathcal{L}(b, \lambda) \\ \\ \text{Find} & \displaystyle\sup_{\lambda \in \mathbb{R}_+^{R+1}} G(\lambda) \end{cases}$$

[1]We recall that if the interval is slipt up in $2N$ subintervals of same legnth $[a, b] \approx \{a, x_1, \ldots, x_{2N-1}, b\}$, then we have the approximation $\displaystyle\int_a^b f(\xi)\, d\xi \approx \frac{\Delta\xi}{3} \left[f(a) + 4 \sum_{j=1}^{N} f(x_{2j-1}) + 2 \sum_{j=1}^{N-1} f(x_{2j}) + f(b) \right]$.

The Usawa algorithm combine a gradient method and a projected gradient method. Hence, we need a derivative for the lagrangian. Numerically, if we take the limit of the quantity $\dfrac{\mathcal{L}(b+\varepsilon h,\lambda)-\mathcal{L}(b,\lambda)}{\varepsilon}$ when $\varepsilon \to 0$ and write it in the form $\langle d\mathcal{L}(b)|h\rangle$, we get :

$$\begin{pmatrix} d\mathcal{L}(b)_0 \\ \vdots \\ d\mathcal{L}(b)_{R-1} \end{pmatrix} = \begin{pmatrix} dJ(b)_0 \\ \vdots \\ dJ(b)_{R-1} \end{pmatrix} - \underbrace{\begin{pmatrix} \lambda_0 \\ \vdots \\ \lambda_{R-1} \end{pmatrix}}_{\lambda^*} + 2\lambda_R \begin{pmatrix} b_0 \\ \vdots \\ b_{R-1} \end{pmatrix}$$

which we can write introducing the vector $\lambda^* \in \mathbb{R}^R$ in a synthetic way :

$$\boxed{d\mathcal{L}(b) = \frac{1}{2}\mathbf{1}_{[-S,S]}\int_0^T \frac{\partial v_{\zeta_b}}{\partial \xi} - \lambda^* + 2\lambda_R b}$$

11.2 Behaviour expected from the algorithm

During the initialization, we choose an initial bottom that satisfies all the conditions required to be in the admissible set of solutions. It means that $\lambda = 0$ and so $\mathcal{L}(\lfloor,\lambda) = \mathcal{J}(\lfloor)$. Therefore, at the beginning, the evolution of the bottom shape will only be ruled by the functionnal and not the constraints.

As we know that the wave height is very sensitive to the bottom one and that the support is fixed, we expect from the algorithm to increase the bottom maximum elevation d. However, after some iterations, the bottom won't be anymore in the admissible set which means that $\lambda > 0$. Then, the constraints will begin to act more and more on the evolution of the bottom shape in order to bring the bottom back to the admissible set.

Therefore, we expect from the algorithm oscillations of the funtionnal around an equilibrium between the will of the functionnal and the constraints. This is exactly the meaning of the saddle point $\sup\limits_{\lambda\in\mathbb{R}_+^{R+1}} \inf\limits_{b\in\mathbb{R}^R} \mathcal{L}(b,\lambda)$.

The convergence of the algorithm will mainly depends on how quickly the constraints will intervene or not in the process respectively if the constraints are not satisfied or if they are. This will be ruled by a numerical parameter κ really difficult to tune.

11.3 Description of the algorithm

11.3.1 Initialization

We choose a bottom $b^0 \in \mathbb{R}^R$ verifying the constrains $b^0 \geqslant 0$, supp $b^0 \subseteq [-S,S]$ and $\displaystyle\int_{\mathbb{R}} \left(b^0\right)^2 \leqslant M^2$. We also choose a $\lambda^0 \in (\mathbb{R}_+)^{R+1}$. In the algorithm, we fix $\lambda^0 = 0$.

In order to test the performances of the algorithm, we implemented it for various initial bottom. We choose :

- a cube profile $b^0(\xi) = d\mathbf{1}_{[-S,S]}(\xi)$;
- a cosinus profile $b^0(\xi) = \dfrac{d}{2}\left[1 + \cos\left(\dfrac{\pi\xi}{S}\right)\right]\mathbf{1}_{[-S,S]}(\xi)$;
- a triangle profile $b^0(\xi) = d\left(\dfrac{S+\xi}{2S}\right)\mathbf{1}_{[-S,S]}(\xi)$;
- an inverse triangle profile $b^0(\xi) = d\left(\dfrac{S-\xi}{2S}\right)\mathbf{1}_{[-S,S]}(\xi)$;
- a bowl profile $b^0(\xi) = d\left(\dfrac{\xi}{S}\right)^4\mathbf{1}_{[-S,S]}(\xi)$;
- a double semi-elliptic profile of the form

$$b^0(\xi) = d\left[\mathbf{1}_{[-S,S/2]}(\xi)\sqrt{1-\left(\frac{4\xi}{S}+3\right)^2} \quad + \quad \mathbf{1}_{[S/2,S]}(\xi)\sqrt{1-\left(\frac{4\xi}{S}-3\right)^2}\right].$$

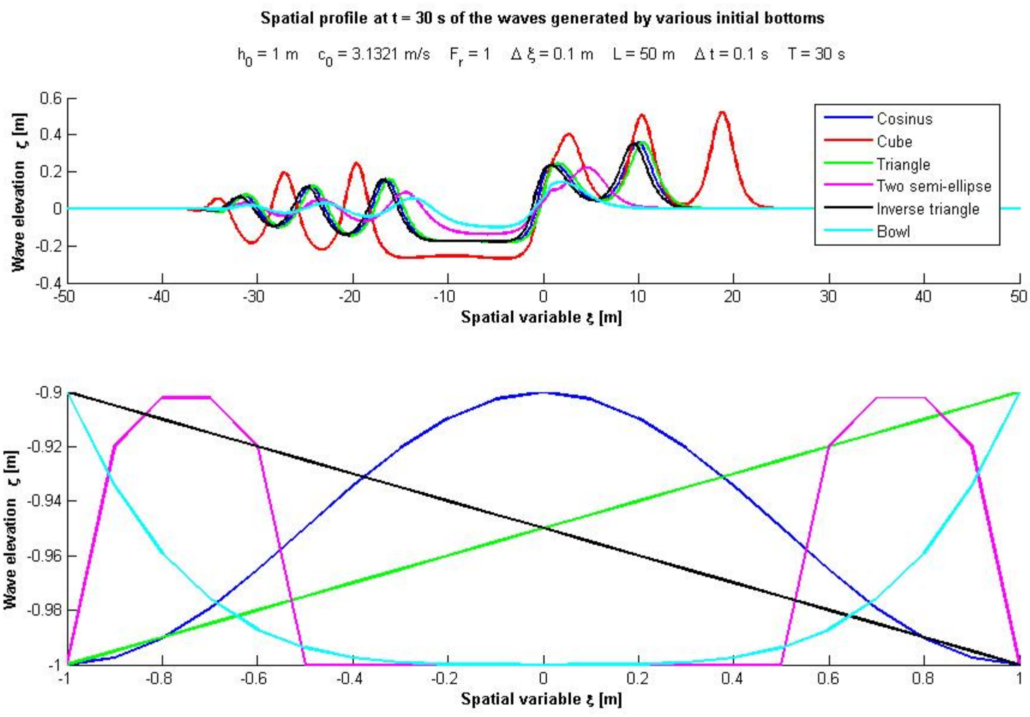

Figure 11.1: *We compared the results obtained by the fKdV model on various initial bottoms and oberve the differences between the waves generated.*

11.3.2 Step $k+1$: re-initialization of b^k

We suppose we know $b^k \in \mathbb{R}^R$ and $\lambda^k \in (\mathbb{R}_+)^{R+1}$. Then, a gradient descent is implemented. We introduce a new bottom capable of decreasing sufficiently the functionnal $b \to \mathcal{L}(b, \lambda^k)$ in the optimal local direction $d\mathcal{L}(b^k, \lambda^k)$ [2].

$$b^{k+1} = b^k - \gamma d\mathcal{L}(b^k, \lambda^k)$$

We have to choose a correct $\gamma \in \mathbb{R}$ and that is an hard task. The ideal case would be the minimum of the function $\Psi: \ \gamma \to \mathcal{L}\left(b^k - \gamma d\mathcal{L}(b^k, \lambda^k), \lambda^k\right)$ on the entire set $\mathbb{R}$ but the evaluation of the function at one point needs each time to solve two partial differential equations.

Moreover, if the γ is negative and too important, the bottom will become negative and will leave the admissible set. If it is positive and too big, the bottom will be too high and will violate the physical limit of the fKdV model. Therefore, we must restrain the choice of γ in a small interval and find the minimum of Ψ on it.

11.3.3 Choice of γ

We can first see the γ as a rescaling between $d\mathcal{L}(b^k, \lambda^k)$ compared to b^k. Hence, we first choose the following value :

$$\gamma_0 = 10^{\mathrm{Ent}\left[\log_{10}(b^k)\right] - \mathrm{Ent}\left[\log_{10}\left(d\mathcal{L}(b^k, \lambda^k), \lambda^k\right)\right] - 2}$$

which basically impose that $\gamma d\mathcal{L}(b^k, \lambda^k), \lambda^k)$ is two orders of magnitude less than b^k. I then evaluate $\Psi(-\gamma_0)$ and $\Psi(\gamma_0)$. As I know three point with $\Psi(0)$, an interpolation with a polynom of degree two is done on Ψ and the minimum γ_{opt} is found on the interval $[-\gamma_0, \gamma_0]$.

This evaluation procedure is justified in the case of our problem. Indeed, we studied precisely the function Ψ during many phases of the algorithm and it turns out that Ψ profile is close to a parabola. The interpolation is then cheap in time and efficient in this case.

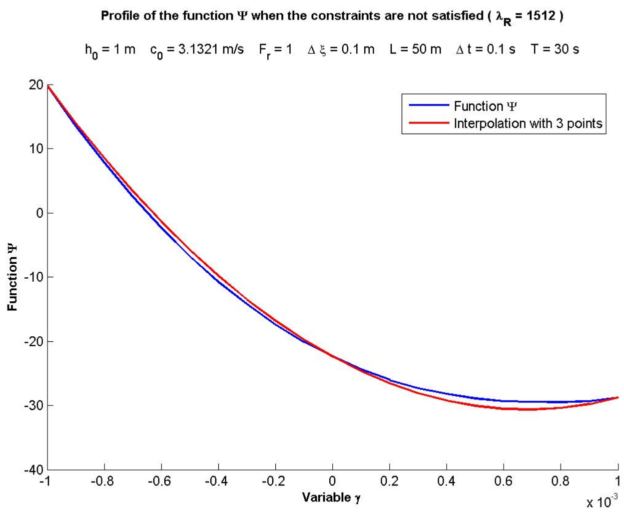

Figure 11.2: *Profile of the function Ψ and its interpolation on a large interval.*

[2] We recall here that the functionnal is an integral evaluated by the Simson rule described above.

11.3.4 Step $k+1$: re-initialization of λ^k

We then compute easily the projection of the solution in the space $(\mathbb{R}_+)^{R+1}$. This gives in term of vectors:

$$\left(\lambda^{k+1}\right)^* = \max\left(0\ ,\ \left(\lambda^k\right)^* - \kappa b^{k+1}\right) \qquad \text{and} \qquad \lambda_R^{k+1} = \max\left[0\ ,\ \lambda_R^k + \kappa\left(\int_{\mathbb{R}} b^{k+1\ 2} - M^2\right)\right]$$

where κ is a parameter that the user has to fix so as to make the two term of the same order of magnitude. This parameter rules how much you want the constrains to penalize the functionnal in the optimization process.

11.3.5 Choice of κ

This is the hardest part of the algorithm because it can only converge for an appropriate κ which is a computational parameter, completly independant of the physical problem. Moreover, we observed that the lagrangian and the functionnal are very sensitive to this parameter.

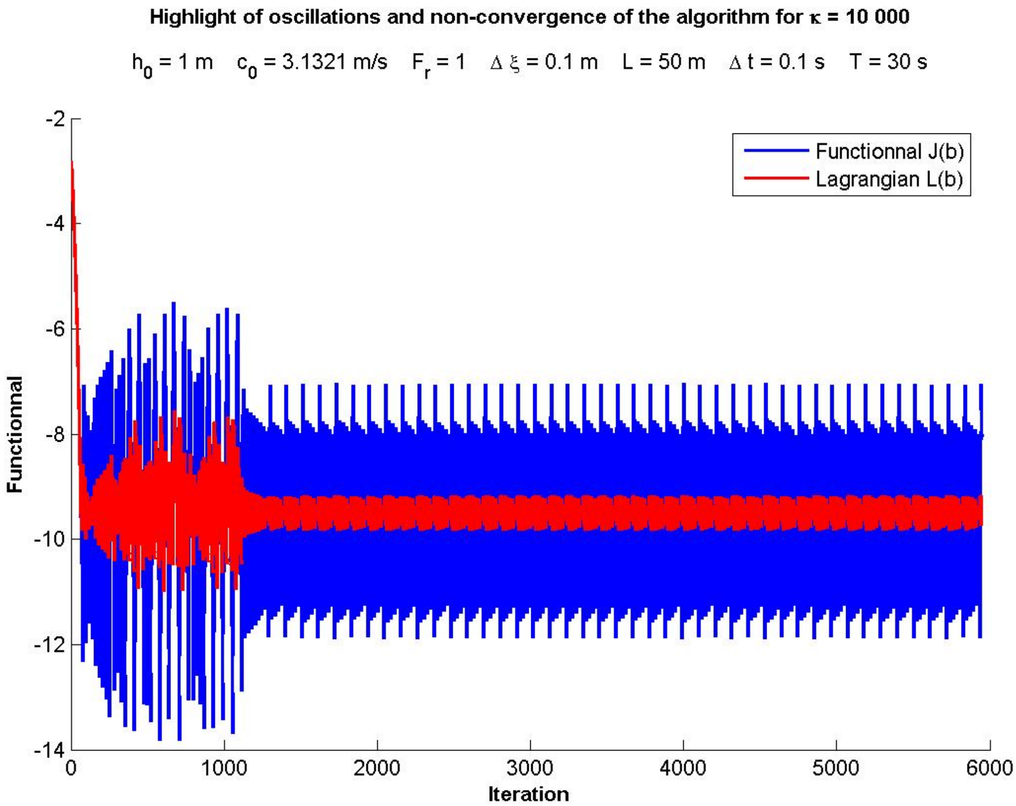

Figure 11.3: *Profile of the functionnal during many iterations of the algorithm. We can observe the highly oscillating behaviour of the functionnal J and the non-convergence of the algorithm. A periodicity is even visible here for $\kappa = 10\ 000$.*

Indeed, on the one hand, if the κ is too small, the constraints will act very late in the process of penalization which means high oscillations and no chance of convergence for the algorithm.

On the other hand, if the value of κ is too high, the constraints will be immediatly significant leading to an instability of the program. During one iteration $\lambda = 0$ and during the next one $\lambda > 0$ will be very high leading again to the case $\lambda = 0$. In conclusion, oscillations will be again observed.

In our context, the right κ values are located around 30 000. To make the algorithm converge, its value depends on the initial bottom b^0 which make the task even more difficult. In the table below, we put the accurate value of κ found for each of our initial bottoms.

Initial bottom	**Value of κ**
Two semi-ellipse	30 100
Triangle	30 200
Cube	30 100
Bowl	29 000
Inverse triangle	29 000
Cosinus	30 000

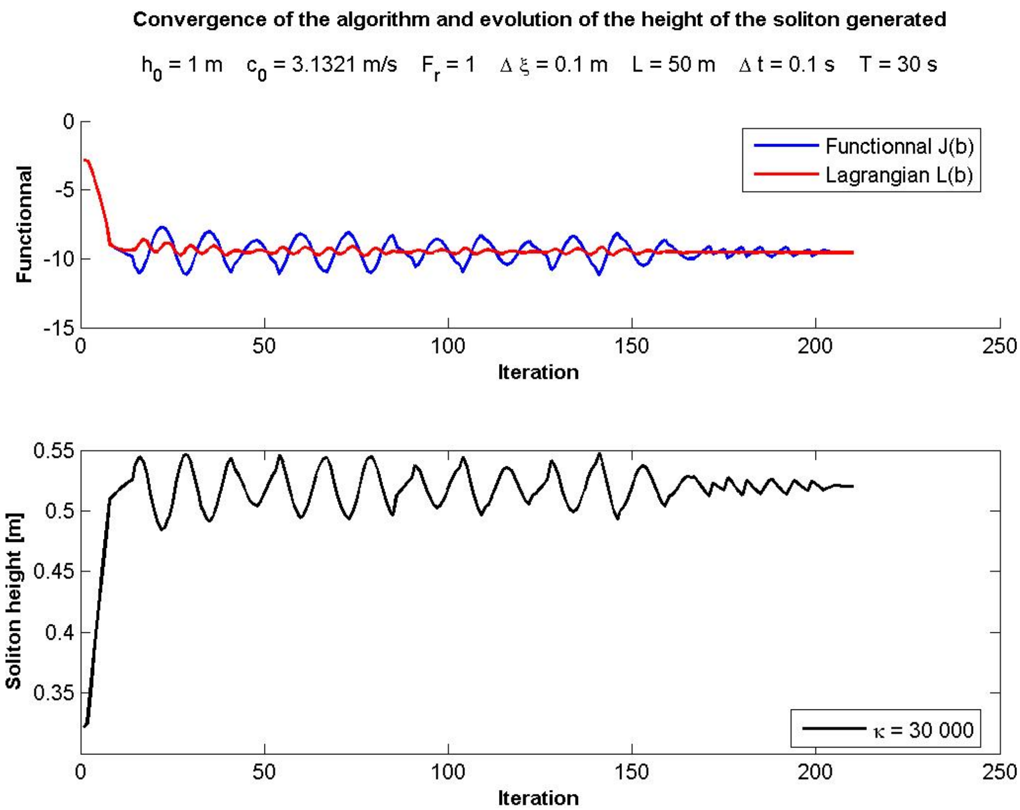

Figure 11.4: *Example of convergence for a cosinus initial profile.*

Finally, an interpretation of the algorithm behaviour can be found. We know that the functionnal is very sensitive to the bottom height. A little perturbation of the first generates high one of the second. That is why the convergence is difficult to obtain and oscillations are omnipresent in the algorithm.

11.3.6 Stop criteria

The stop criteria leads to another precision parameter to fix denoted ι. It depends on the precision on the bottom you want to impose. We can stop where these three conditions are fulfilled :

$$\begin{cases} b^{k+1} \in \mathcal{B} \\ \\ \max\limits_{0 \leqslant i \leqslant N-1} \left(b_i^{k+1} - b_i^k \right) < 10^{-\iota_b} \\ \\ \max\limits_{0 \leqslant i \leqslant N-1} \left(\lambda_i^{k+1} - \lambda_i^k \right) < 10^{-\iota_\lambda} \end{cases}$$

11.4 Results obtained from the algorithm

We present in the table below all the values taken for the parameters in the algorithm.

Parameter	**Notation**	**Value**
Water depth	h_0	1 m
Gravity acceleration	g	9.81 m.s^{-2}
Froude number	F_r	1
L^2-upper bound for bottom	M^2	0.02 m^4
Penalization parameter	κ	$\sim$ 30 000
Bottom support	$[-S, S]$	$[-1\ m, 1\ \text{m}]$
Space step	$\Delta\xi$	0.1 m
Spatial domain	$[-L, L]$	$[-50\ \text{m}, 50\ \text{m}]$
Time step	Δt	0.1 s
Final time	T	30 s

The graphic above sums up the situation. Our algorithm has been tried on many initial bottoms and it converges almost to the same optimal bottom. This numerically proves the existence and uniqueness of a solution to our optimization problem.

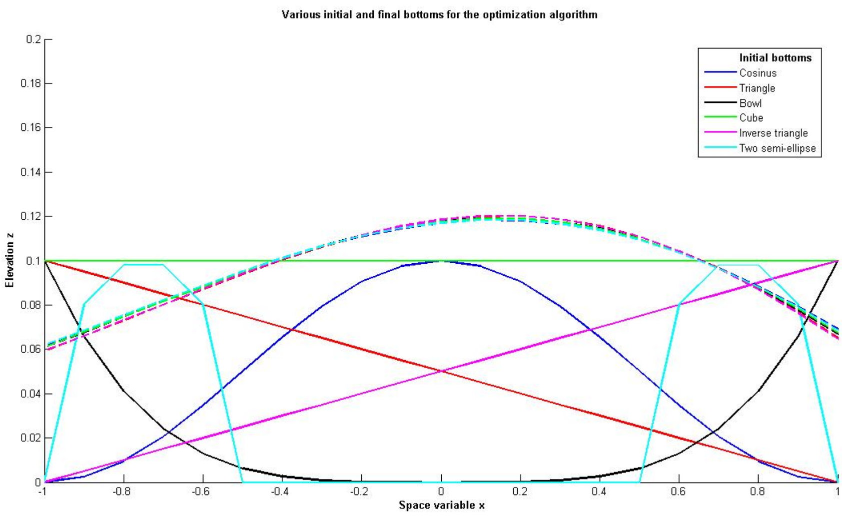

Figure 11.5: *Results obtained via the optimization algorithm.*

Chapter 12

The influence of parameters

We study here the influence of various parameters that where fixed before in the algorithm. We improve the efficiency of the wave maker by a tuning of the Froude number. We then show the limit of our model and try to minimize the energy necessary to create the wave. Finally, we study the influence of the admissible set of solution on the optimal shape.

12.1 The role of the Froude number

The Froude number is defined as the ratio between the speed U of the bottom to the reference speed $c_0 = \sqrt{gh_0}$ of the waves. Some questions arise about why this ratio must be taken close to the unity. Indeed, let's recall its influence in the case of a monodimensional linear hydrostatic flow described in [2] .

12.1.1 Physical meaning of the Froude number

Incompressible, homogeneous inviscid fluid

We recall the equations for the motion of an incompressible, homogeneous inviscid fluid with constant density ρ :

$$\begin{cases} \dfrac{D\overrightarrow{u}}{Dt} = -\dfrac{1}{\rho}\overrightarrow{\nabla}P + \overrightarrow{g} \\ \\ \overrightarrow{\nabla}.\overrightarrow{u} = 0 \end{cases}$$

where D/Dt is the material derivative, $\overrightarrow{u} = (u, v, w)^T$ with components in the cartesian directions (x, y, z) respectively. We assume that some bottom topography is present and has the form $z = -h_0 + b(x, y)$, with a base level at $z = -h_0$. The boundary condition of zero velocity normal to this surface may then be expressed as :

$$w = \overrightarrow{u}.\overrightarrow{\nabla}h \text{ on } z = -h_0 + b(x, y)$$

The fluid has an upper free surface at the mean level $z = 0$, with a displacement of equation $z = \zeta(x, y, t)$. The mass of any fluid located above is negligible so that the pressure at the surface is constant. Consequently, we have :

$$\begin{cases} P = P_a \\ \\ w = \dfrac{D\zeta}{Dt} \end{cases} \quad \text{on } z = \zeta(x, y, t)$$

Hydrostatic flow

The curl of the equation of motion gives the vorticity equation $D\overrightarrow{\omega}/Dt = \overrightarrow{w}.\overrightarrow{\nabla}\overrightarrow{u}$, and if $\overrightarrow{\omega} = \overrightarrow{\nabla} \times (\overrightarrow{u}) = 0$ initially it remains so, and the motion is irrotational throughout. If the vertical accelerations Dw/Dt are everywhere much less than gravity, the motion is said hydrostatic, so that :

$$-\frac{1}{\rho}\frac{\partial P}{\partial z} - g = 0$$

If the horizontal scale of the fluid is λ, simple scale analysis shows that this approximation holds if $\varepsilon = (h_0/\lambda)^2 << 1$. We then have $P = P_a + \rho g (\zeta - z)$ within the fluid, and the horizontal pressure gradient is independant of z. Hence, if u and v are initially independant of z, they will remain so. This independance and the formula for P applied on the equations of motion gives :

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} + w\underbrace{\dfrac{\partial u}{\partial z}}_{=0} = -\dfrac{1}{\rho}\underbrace{\dfrac{\partial P}{\partial x}}_{=\rho g(\partial\zeta/\partial x)} \\ \dfrac{\partial v}{\partial t} + u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} + w\underbrace{\dfrac{\partial v}{\partial z}}_{=0} = -\dfrac{1}{\rho}\underbrace{\dfrac{\partial P}{\partial y}}_{=\rho g(\partial\zeta/\partial y)} \\ \underbrace{w|_{z=\zeta}}_{=(D\zeta/Dt)} - \underbrace{w|_{z=-h_0+b}}_{=\overrightarrow{u}.\overrightarrow{\nabla}b} = \displaystyle\int_{-h_0+b}^{\zeta}\frac{\partial w}{\partial z}dz = -\underbrace{\int_{-h_0+b}^{\zeta}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)dz}_{=(\partial u/\partial x\ +\ \partial v/\partial y)(\zeta+h_0-b)} \end{cases}$$

which can be rewriten in a synthetic form. The suffices denote derivatives and $h = h_0+\zeta-b$ the thickness of the fluid layer. We then obtain the equations of motion for a classical hydrostatic flow :

$$\begin{cases} u_t + uu_x + vy_y = -g\zeta_x \\ v_t + uv_x + vv_y = -g\zeta_y \\ \zeta_t + (hu)_x + (hv)_y = 0 \end{cases}$$

Linear flow

We now consider a uniform stream of velocity U in the x-direction, and insert a long obstacle with small height of the form $z = b(x, y)$ into this stream, by uplift from below. Linearising the previous equation about this initial state gives $u = U + \delta u$ and :

$$\begin{cases} (\delta u)_t + U(\delta u)_x = -g\zeta_x \\ v_t + Uv_x = -g\zeta_y \\ \zeta_t + U\zeta_x + h_0((\delta u)_x + v_y) = Ub_x \end{cases}$$

One-dimensional flow

Finally, if we assume the monodimensionnal hypothesis (b independant of y), then the resulting flow will have $v = 0$ everywhere. Eliminating δu and v gives the evolution equation for ζ and with the initial data, the problem takes the form :

$$\begin{cases} \left[\frac{\partial}{\partial t} + \left(U - \sqrt{gh_0}\right)\frac{\partial}{\partial x}\right]\left[\frac{\partial}{\partial t} + \left(U + \sqrt{gh_0}\right)\frac{\partial}{\partial x}\right]\zeta = U^2 \frac{\partial^2 b}{\partial x^2} \\ \zeta\left(x, y, 0\right) = b\left(x, y\right) \\ \frac{\partial \zeta}{\partial t}\left(x, y, 0\right) = 0 \end{cases}$$

The sudden introduction of the obstacle results in an instantaneous potential flow, which causes a deformation of the interface but does not immediatly alter the fluid velocity. This system has one dimensionless parameter : the Froude number F_r for the undisturbed flow, defined by :

$$F_r = \frac{U}{\sqrt{gh_0}}$$

which is the ratio of the flow speed U to the speed of long gravity waves on stationary fluid of depth h_0. The solution to the problem above can be found by the standard method of characteristics.

The role of the Froude number

We can immediatly see that one has to distinguish between two cases : if $F_r = 1$ or not. We have :

$$\zeta\left(x, t\right) = \frac{F_r^2}{F_r^2 - 1} b\left(x\right) + \frac{1}{2}\left[\frac{b\left(x - \left(U + \sqrt{gh_0}\right)t\right)}{F_r + 1} - \frac{b\left(x - \left(U - \sqrt{gh_0}\right)t\right)}{F_r - 1}\right]$$

provided that $F_r \neq 1$. The solution consists of a steady component over the obstacle and two waves, the one with larger amplitude porpagating against the stream and the other with smaller amplitude with it. All the three terms have the form of the obstacle itself and the two wave terms are functions of the characteristic variables. Therefore, the larger wave appears on the upstream side if $F_r < 1$ and on the downstream side if $F_r > 1$.

The solution becomes singular as $F_r \to 1$, and cannot be valid near this point because the linearisation assumption of small amplitude is violated. For $F_r = 1$, the solution is given by :

$$\zeta\left(x, t\right) = \frac{Ut}{2}\frac{\partial b}{\partial x} + \frac{1}{4} b\left(x - 2Ut\right) + \frac{3}{4} b\left(x\right)$$

Here, the flow over the obstacle grows linearly with time, but the downstream-propagating wave is unaffected. This singular behaviour arises because the forcing by the obstacle resonates with the upstream-propagating wave. For this resonant case, there is an associated drag force on the obstacle, whereas there is no drag in the steady state for $F_r \neq 1$. That is why we search F_r around the unity in the derivation of the forced KdV equation.

12.1.2 The supercritical case to enhance the efficieny of the wave maker

In the figure below, the behaviour of the solitary wave height generated by the optimal bottom is studied according to the Froude number F_r. We can clearly see that in the supercritical case, there is an maximum around $F_r \approx 1.2$, a value that is still in the physical limit of the fKdV model. Therefore, a tuning of the bottom speed has to be done so as to obtain a higher wave and this value is around 1.2 c_0.

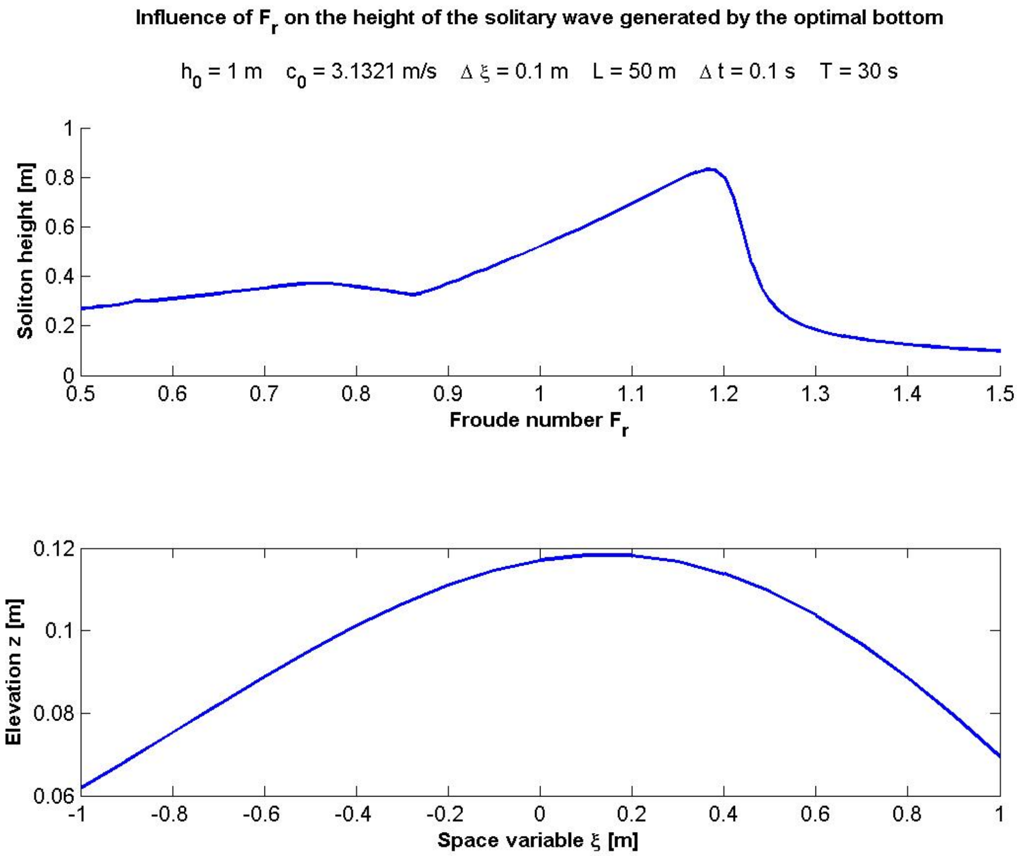

Figure 12.1: *The dependancy of the optimal wave height according to the Froude number. he optimal value seems to be for* $F_r = 1.2$.

12.2 Some energetic considerations and weakness of the fKdV model

12.2.1 A comparison between the cube and the optimal profile

When the height of the wave produced by the optimal bottom and by the cube are compared, we can immediatly that from an mathematical point of view, the optimal bottom generates a higher wave than the cube profile.

However, from an engineering point of view and taking in account all the errors commited during the discretization process, the difference between them is negligible. Moreover, a cube is much more faster to built than the complicated profile of the optimal bottom. Therefore, some other argument have to be found.

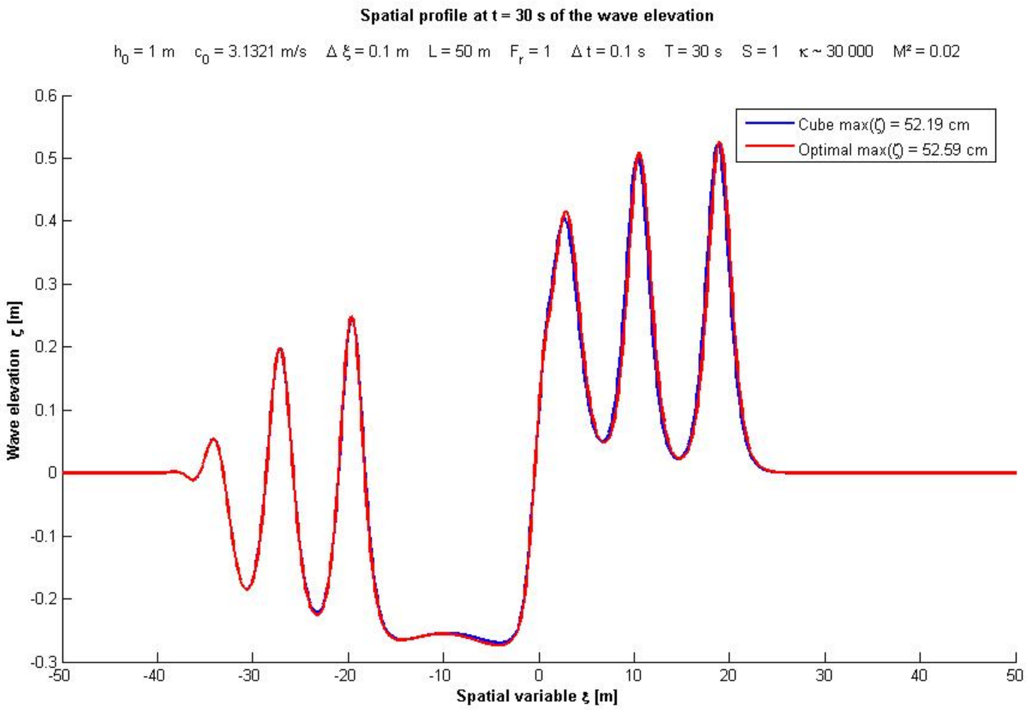

Figure 12.2: *The comparison between the wave elevation of the cube and the one of the optimal bottom. The difference is physically negligible.*

12.2.2 The weakness of the model : the inviscid assumption

From an engineering point of view, if two bottoms are able to produce the same waves, their preference will go to the one that can generate it with the less energy as possible. Therefore, the most aerodynamical shape will be chosen that is our potimal bottom.

However, our model assume that the fluid is inviscid. Most of the energy necessary to the wave generation comes from the adherence of the moving bottom on the viscuous fluid (water). Therfore, this inviscid assumption does not allow the access to this energy and the viscuous effects are crudely taken in account.

This highlights maybe one of the weakness of the fKdV equation. Although it is known for modeling very well the real phenomena over a wide domain of physical validity, this model does not capture the viscuous effects. Consequently, the effects of the shape is less taken in account than it should, especially in the optimization process.

12.2.3 The influence of the shape on the drag coefficient

Nevertheless, a certain part of this energy is accessible via the model. A drag which corresponds to the resistance due to unsteady wave making can be evaluated. Denoted D_w by Wu in [20], the drag experienced by the moving disturbance is given by the formula :

$$D_w(t) = \int_{-S}^{S} b(\xi) \frac{\partial \zeta}{\partial \xi}(\xi, t)\, d\xi$$

This definition is essentially based on the fKdV rule of equivalence between a pressure disturbance $p(\xi) = p(x - Ut)$ and a bottom one $b(\xi)$. Indeed, the fKdV equation treats the two cases in the same way, included in the forcing term of the equation, whereas the generalized Boussinesq model treats them differently in its equations.

Then, in his article, Wu defined the associated drag coefficient C_{D_w} and shows that for positive forcing, the drag coefficient oscillates nearly sinusoidally about a positive mean value $\overline{C_{D_w}}$ and get an estimation of it via the soliton height a :

$$C_{D_w}(t) = \frac{D_w(t)}{2\rho g h_0 S} \quad \text{and} \quad \overline{C_{D_w}} = \frac{a^3}{4}$$

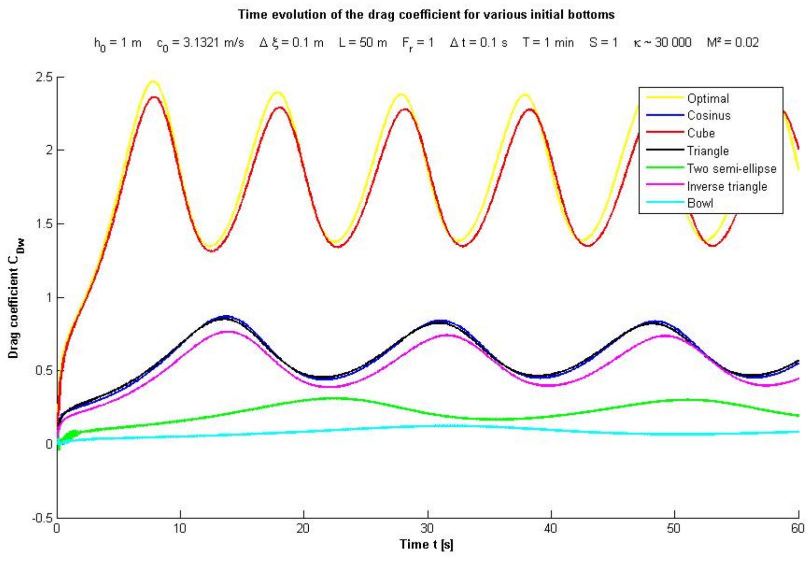

Figure 12.3: *The time evolution of the drag coefficient for various bottoms. We can see that the initial bottoms that generate a higher wave will have a higher drag mean.*

This traduces a physical observation : if you want a higher wave, you will need more energy. If this coefficient captures this phenomena, one can ask how well the model capture in this drag coefficient the shape influence on the total energy used to operate the wave maker. Unfortunately, the graphic below shows that there is no real shape influence on the drag coefficient. However, it is clear that the optimal shape is more aerodynamical than the cube.

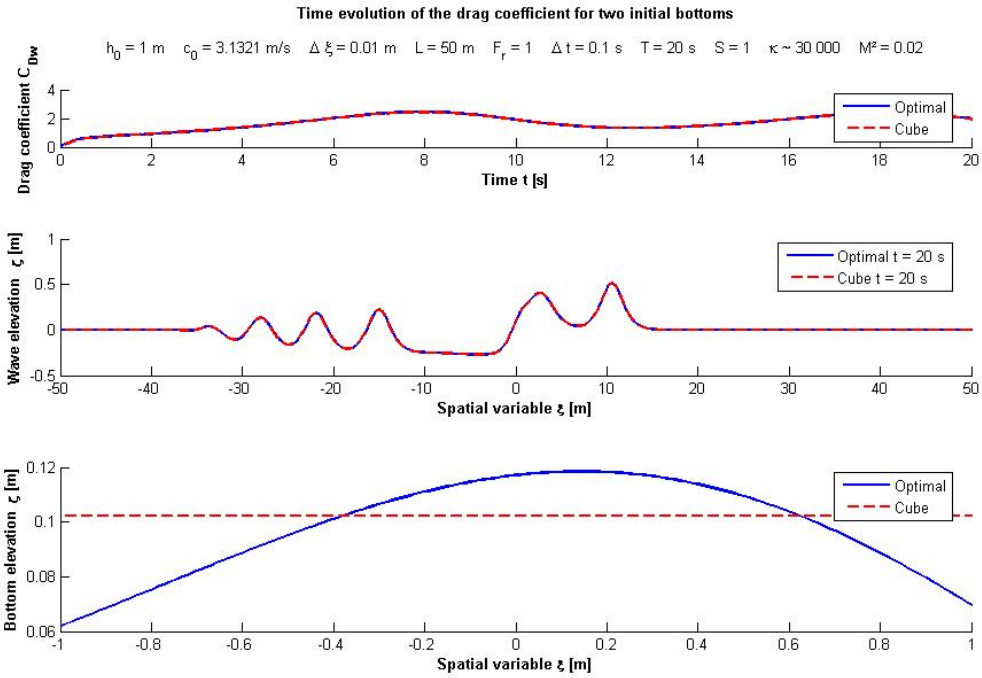

Figure 12.4: *The comparison between the drag coefficient of the optimal bottom and the one of a cube that gives the same wave profile. The difference is physically negligible.*

12.3 The influence of the admissible set of solutions

The remarks made in the previous section shows that in a certain way, our optimization problem seems to be mathematically well-posed because the algorithm converges to a unique solution. However, is this optimal bottom physically more efficient than a cube ?

The graphic below reassures us about the physical pertinence of the shape obtained by the algorithm. We compared the shape derivative of the functionnal with the shape of the optimal bottom. We can clearly see that there are really close.

The shape derivative of the functionnal obtained via the adjoint formulation is a continuous approach which is rigorous theoretically. This gives us the local directions where the bottom wants to be modified in order to minimize the functionnal. The similarities between the two graphs is a good sign for the physical efficiency of the optimal bottom.

Nevertheless, if the optimal bottom b looks like the shape derivative of J, it means that the optimization has been done in a space where the constraints were adapted in the sense that they were just acting in order to maintain the bottom height to a reasonnable elevation.

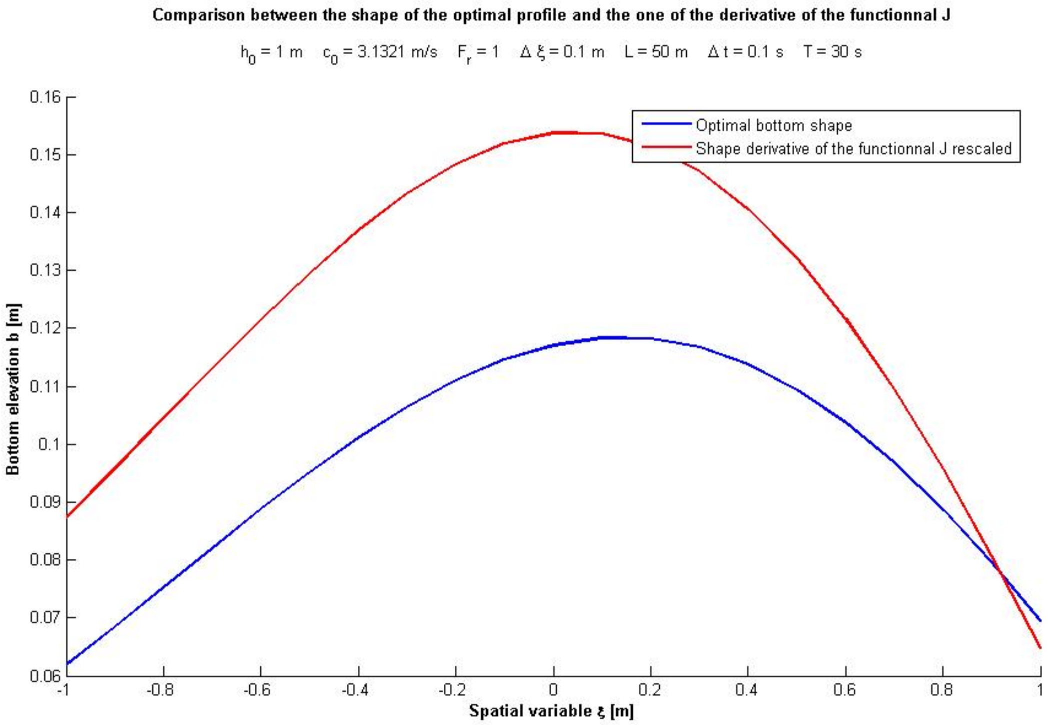

Figure 12.5: *The comparison between the shape of the optimal bottom $b(\xi)$ and the shape derivative of the functionnal $dJ(b)$ rescaled by $10^{\mathrm{Ent}[\log_{10}(\max b)]-\mathrm{Ent}[\log_{10}(\max dJ)]}$ in order to be of the same order of magnitude.*

Therefore, the admissible set of solutions has been of first importance. Let's now define a new admissible set of solutions. First, the positive assumption cannot be neglected for practical reason : a negative bottom can't be physically and safely translated in a pool.

Then, the support has always been fixed to $[-1, 1]$. This localization cuts the optimal bottom shape in a crude way. However, a tuning of the support can be done in order to get a smooth bottom profile, more aerodynamic, and that will improve the wave height. To find this accurate support, it can be usefull to look at the graph of the shape derivative of the function dJ as its profile is close to the one of the optimal bottom.

Finally, the norm can be changed so as to see how much the set influence the optimal shape. We can distinguish two opposite points of view :

1. The L^1-norm will bound the area of the bottom. Therefore, as the wave elevation is more sensitive to the bottom height than its shape, the dirac will be the optimal bottom found for this norm.
2. The L^∞-norm will bound the bottom pointwisely. Therefore, as we have already seen, the optimal bottom will be a cube.

The L^2-norm is a balance between these two extrema and it seems to be the right one in physical term. On the graphic below, we plotted the optimal bottom for three intermediate space : $L^{1.5}, L^2$ and L^4. This illustrates perfectly the influence of the admissible space on the shape of the optimal bottom.

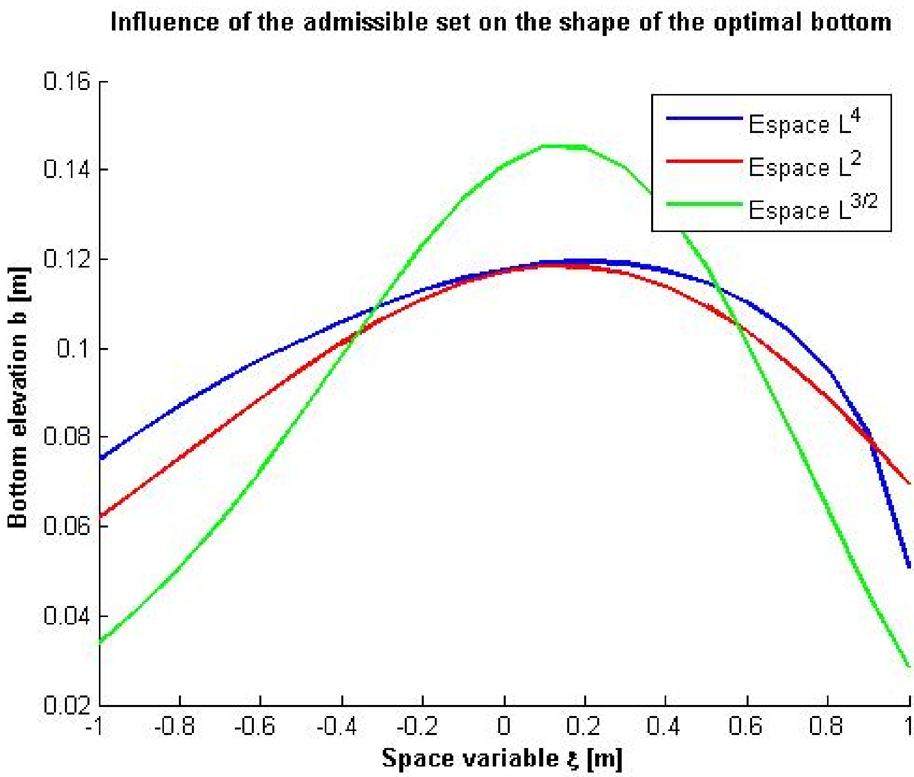

Figure 12.6: *The influence of the space on the shape of the optimal bottom.*

Conclusion

The project Instant Surfing led to the development of a new wave maker : a bottom is translating underwater in a pool. Invited by BCAM for a workshop on computational fluid dynamics, the prototype was presented. During my internship, I studied the forced Korteweg-de Vries equation to model the observed phenomena and I obtained an optimal shape for the bottom that would create the highest wave.

The accuracy of the forced KdV model

First, the fKdV equation seems to be the most adapted model to study the wave maker. Indeed, it uses the fact that the steady translation of the bottom is unidirectionnal and unidimensionnal. It takes in account the non-linear and dispersive effects to a certain extend. Moreover, besides its simplicity, it provides results very close from what have been observed during experiments and over a wide domain of validity ($h_0^2 = 0.5\lambda^2$).

Then, the model shows that the solitary wave produced upstream the disturbance is a soliton. It means a very stable wave that can travel over very large distances without changing its initial form. Moreover, given a certain water depth h_0, its shape is only ruled one degree of freedom : its height a.

Finally, we know its speed from the relation $c^2 \sim g\,(h_0 + a)$. After some time, the soliton becomes autonomous and goes faster than the bottom. We have an estimation for the period of generation $\tau = 64\backslash\,(3a)^{1.5}$. Therefore, after the time τ, it is possible to stop the bottom : the wave is generated. Moreover, higher is the wave and less time is needed to produce it.

The influence of parameters

First, Instant surfing prototype seems to consume less energy than a classical wave maker based on the drop of accelerated water in a pool. However, like every inviscid model, the fKdV model does not allow an access to the energy needed for the translation of the bottom. Indeed, most of the dissipated energy comes from the adherence of the moving bottom to the water. We can only evaluate a non-significant part of this energy with a drag D_w experienced by the moving bottom. It corresponds to the resistance due to the unsteady wave making. However, this drag coefficient seems not to be influenced by the shape of the bottom in the fKdV model whereas a more aerodynamical shape of the bottom would reduce significantly the energy needed for the wavemaker operation.

Then, the height of the generated wave is very sensitive to the bottom one. Indeed, there is a quasi-linear dependance between them. Therefore, in order to stay in the validity of the fKdV model, the pool must be as deep as it can, taking in account the economic constraints due to the water and maintenance. Therefore, if the water depth is bigger, the fKdV model ensures a higher limit for the bottom height implying a higher wave produced.

Finally, the solitary wave can be produce only if the bottom is moving steadily at a transcritical speed $c_0 = \sqrt{gh_0}$. The Froude number F_r defined by the ratio of the bottom speed to c_0 must be close to the unity to generate a soliton. This parameter modifies the repartition of the bottom energy in the three regimes observed from the fKdV model : cnoïdal-like waves downstream, depressed water surface and solitons upstream. Taking F_r a bit greater than 1 will dedicate more energy to the solitary wave, producing a higher one. However, if it is too far from 1, the fKdV model is not valid anymore and the wave height decreases. Therefore, there is an optimal value for F_r close to the 1.2 that produce higher wave than the classical case $F_r = 1$.

The influence of shape

The Usawa algorithm depends on a numerical parameter κ that must be close to 30 000 in order to ensure the convergence of the program. However, its value depends of the initial bottom chosen. We obtained a unique optimal shape from a wide range of initiam bottoms. This reinforces this idea that there exists a unique global optimal bottom for our optimization problem and that it is the one found by the algorithm.

However, the optimal shape depends a lot on the admissible set of bottoms. Indeed, we decided to find it between the L^2 integrable and positive bottoms whose L^2-norm is upper bounded. A change of the admissible space would completly modify the optimal shape. A space closer to L^1 will tend to produce a dirac shape whereas a space closer to L^∞ would give a cube. This highlights that our optimal shape is a solution of our mathematical problem but maybe physically far from the real desired shape. Fortunately, the L^2-norm is the intuitive one related to the physical energy of the system. Therefore, some experiments should be done to see if this shape really enhances the performance of the wave maker in term of energy and of wave height.

Finally, in the algorithm, we decided to localize the bottom to a certain support that really cut the shape. An appropriate support has to be found so as to get smoothly to the pool ground. Furthermore, we showed that a cube that has a L^2-norm negligibly bigger than the constraints imposed on the admissible set would create the same wave height than the optimal bottom found. Mathematically, this bottom is not in the admissible space but physically would be easier to build. Only economic argument based on the energy needed to produce the wave could decide between those two. We hope the optimal shape we found has a physical reality and not only the denomination of solution to our mathematical problem. Only experiments could tell.

To conclude, the fKdV model is accurate for such a wave maker. It gives a lot of answers about the comprehension of the observed phenomena. If we solved our mathematical problem, the main default of the model is the inviscid-flow hypothesis that reduces a lot the shape influence on the generation of the solitary wave. In order to better the results obtained in this report, only a viscuous model could improve them and gives some answers to what experiments could tell.

Part V

Annexes

Chapter 13

The Zabusky and Kruskal historical simulation of KdV equation

We reproduced the work of N.J. Zabusky and M.D. Kruskal they published in 1965 in a paper named *Interaction of solitons in a collisionless plasma and the recurrence of initial states.* They observed unusual nonlinear interaction in the numerical solution of the Korteweg-de Vries equation :

$$u_t + uu_x + \delta^2 u_{xxx} = 0.$$

13.1 Description of the phenomena

We have a nonlinear process in which interacting localized pulses do not scatter irreversibly. As the third dispersive term is small at the beginning because $\delta = 0.022$, the equation is almost the Burger's one and u tends to become discontinuous at $T_b \sim \frac{1}{\pi}$: the breakdown time. Then, the third term becomes stronger and appears a series of solitons that are going to travel independently. Finally, at the recurrence time $T_r \sim 30.4T_b$, all the solitons almost reconstruct the initial state.

13.2 Description of the Matlab code

We are using a leap-frog scheme in time and a finite difference scheme in space that conserved mass and energy to second order. First, we discretize first the space variable x and the time variable t according to a CFL condition adapted to the scheme we are using.

$$\begin{cases} N = 100, \quad \Delta x = \dfrac{1}{N} \\ [0,2[\approx \{x_0, ..., x_i, ..., x_{2N-1}\} \\ \forall i \in \{0, ..., 2N-1\}, \quad x_i = i\Delta x \end{cases} \quad \text{and} \quad \begin{cases} \Delta t \leqslant \dfrac{\Delta x^3}{4 + \Delta x^2 \max u} \\ [0,+\infty[\approx \{t_0, ..., t_j, ...\} \\ \forall j \in \mathbb{N}, \quad t_j = j\Delta t \end{cases}$$

Then, we computed this scheme on Matlab denoting $u_i^j = u(x_i, t_j)$, using a trick as a second numerical initial state is needed.

$$\begin{cases} u_i^0 = \cos\left(\pi x_i\right) \\ \\ u_i^1 = \cos\left[\pi\left(x_i - u_i^0 \Delta t\right)\right] \\ \\ \dfrac{u_i^{j+1} - u_i^{j-1}}{2\Delta t} \; + \; \dfrac{u_{i+1}^j + u_i^j + u_{i-1}^j}{3} \dfrac{u_{i+1}^j - u_{i-1}^j}{2\Delta x} \; + \; \delta^2 \dfrac{u_{i+2}^j - 2u_{i+1}^j + 2u_{i-1}^j - u_{i-2}^j}{2\Delta x^3} \; = \; 0 \\ \\ u_i^j = u_{i+2N}^j \end{cases}$$

Finally, as the scheme was becoming unstable after a certain time, every two time steps, we were using Robert-Asselin time filter. If we write the scheme as $u^{j+1} = u^{j-1} + F(u^j)$, then we have :

$$\begin{cases} \widehat{u^1} = u^1, \;\; \widehat{u^{2j+1}} = u^{2j+1} + 0.01\left(u^{2j+2} - 2u^{2j+1} + u^{2j}\right) \\ \\ u^{2j} = u^{2j-2} + F\left(\widehat{u^{2j-1}}\right) \\ \\ u^{2j+1} = \widehat{u^{2j-1}} + F(u^{2j}) \end{cases}$$

13.3 Matlab code

function zabusky_kruskal()

clear all;
close all;

Running_time=cputime;

Data

$N = 100;$
$\delta = 0.022;$
$T_b = \dfrac{1}{\pi};$
$T_r = 30.4 T_b;$
$T_{\max} = 1.1 T_r;$

Spatio-temporal vectors

$\Delta x = \dfrac{1}{N};$
$X = 0 : \Delta x : 2 - \Delta x;$
$\text{Long}X = \text{length}(X);$
$\Delta t = \dfrac{\Delta x^3}{4 + 10\Delta x^2};$

Initialization

$U_0 = \cos(\pi X)$;
Compteur = 1;

figure;
hold on;
plot(X, U_0);
plot($X, 5 + 0.X$,'k');
plot($X, -3 + 0.X$,'k');
xlabel('**Space variable** x ');
ylabel('**Elevation** u ');
legend('$t = 0$ h');
title({'**Numerical simulation realized by N.J. Zabuski and M.D. Kruskal in 1965**',' ','**Interaction of solitons in a collisionless plasma and the recurrence of initial states**',' ',['N = ',num2str(N),' δ = ',num2str(δ),' T_b = ',num2str(T_b),' T_r = ',num2str(T_r),' CFL = Δt = ',num2str(Δt)],' '});
saveas(gcf,['soliton',num2str(Compteur),'.jpg'],'jpg');
close;

U=zeros(Long$X$, 2);
Boucle=[U_0', $\cos(\pi(X - U_0 \Delta t))$'];

Temporal loop

for j = 1:floor$\left(\dfrac{T_{max}}{\Delta t}\right)$
U_{i-2} = [Boucle(end-1:end,2);Boucle(1:end-2,2)];
U_{i-1} = [Boucle(end,2);Boucle(1:end-1,2)];
U_{i+1} = [Boucle(2:end,2);Boucle(1,2)];
U_{i+2} = [Boucle(3:end,2);Boucle(1:2,2)];
U(:,1) = Boucle(:,1) - $\dfrac{\Delta t}{3\Delta x}$($U_{i-1}$+ Boucle(:,2) +$U_{i+1}$) .* $(U_{i+1} - U_{i-1})$

- $\dfrac{\delta^2 \Delta t}{\Delta x}$ $(U_{i+2} - 2U_{i+1} + 2U_{i-1} - U_{i-2})$;

U_{i-2} = [U(end-1:end,1);U(1:end-2,1)];
U_{i-1} = [U(end,1);U(1:end-1,1)];
U_{i+1} = [U(2:end,1);U(1,1)];
U_{i+2} = [U(3:end,1);U(1:2,1)];
U(:,2) = Boucle(:,2) + 0.01(U(:,1) - 2 Boucle(:,2) + Boucle(:,1))

$\dfrac{\Delta t}{3\Delta x}$($U_{i-1}$ + U(:,1) +U_{i+1}) .* $(U_{i+1} - U_{i-1})$

- $\dfrac{\delta^2 \Delta t}{\Delta x}$ $(U_{i+2} - 2U_{i+1} + 2U_{i-1} - U_{i-2})$;

Boucle = U;

if mod$\left[j, \text{floor}\left(\dfrac{0.1\ T_b}{2\Delta t}\right)\right]$ == 0
Compteur = Compteur + 1;

```
        figure;
        hold on;
        plot(X,U(:,2)');
        plot(X,5+0.X,'k');
        plot(X,-3+0.X,'k');
        xlabel('Space variable x ');
        ylabel('Elevation u ');
        legend(['t =',num2str(2jΔt)]);
        title({'Numerical simulation realized by N.J. Zabuski and M.D. Kruskal
        in 1965',' ','Interaction of solitons in a collisionless plasma and the recurrence
        of initial states',' ',['N =',num2str(N),' δ = ',num2str(δ),' T_b = ',num2str(T_b),' T_r =',
        num2str(T_r),' CFL = Δt = ',num2str(Δt)],' '});
        saveas(gcf,['soliton',num2str(Compteur),'.jpg'],'jpg');
        close;
    end;

    if max(max(U)) ⩾ 100;
        break;
    end;

end;
Temps2calcul = cputime - Running_time;
```

Chapter 14

The Whitham and Fornberg simulation of the KdV equation

We reproduced the work of B. Fornberg and G.B. Whitham they published in 1977 in a paper named *A numerical and theoretical study of certain nonlinear wave phenomena*. They implemented a numerical spectral method for the 2π-periodic initial value problem pf the Korteweg-de Vries equation :

$$u_t + uu_x + u_{xxx} = 0$$

14.1 Description of the Matlab code

We are using a leap-frog scheme in time and a spectral method for space. First, we discretize first the space variable x and the time variable t according to a CFL condition adapted to the scheme we are using.

$$\begin{cases} N = 10, \quad \Delta x = \dfrac{\pi}{N} \\ [0, 2\pi[\; \approx \{x_0, ..., x_i, ..., x_{2N-1}\} \\ \forall i \in \{0, ..., 2N-1\}, \quad x_i = i\Delta x \end{cases} \quad \text{and} \quad \begin{cases} \Delta t \leqslant \dfrac{3\Delta x^3}{2\pi^2} \\ [0, +\infty[\; \approx \{t_0, ..., t_j, ...\} \\ \forall j \in \mathbb{N}, \quad t_j = j\Delta t \end{cases}$$

Then, we computed this scheme on Matlab denoting $u_i^j = u(x_i, t_j)$, using a explicit Euler method for the second numerical initial state.

$$\begin{cases} u_i^0 = a\,\mathrm{sech}^2\left[\dfrac{1}{2}\sqrt{\dfrac{a}{3}}\left(x - \dfrac{a}{3}t\right)\right] \\ u_i^1 = u_i^0 \; - \; i\Delta t u_i^0 \mathcal{F}^{-1}\left[\nu \mathcal{F}(u)\right]_i^0 + i\mathcal{F}^{-1}\left[\sin(\nu^3 \Delta t)\,\mathcal{F}(u)\right]_i^0 \\ u_i^{j+1} = u_i^{j-1} \; - \; 2i\Delta t u_i^j \mathcal{F}^{-1}\left[\nu \mathcal{F}(u)\right]_i^j + 2i\mathcal{F}^{-1}\left[\sin(\nu^3 \Delta t)\,\mathcal{F}(u)\right]_i^j \\ u_i^j = u_{i+2N}^j \end{cases}$$

14.2 Matlab code

function whitham_fornberg()

close all;
clear all;
Temps2calcul = cputime;

Data

L = 12;
g = 9.81;
$h_0 = 1$;
$c_0 = \sqrt{gh_0}$;
a = 0.2;
$c = \dfrac{c_0 a}{2h_0}$;

Spatio-temporal discretization

$\Delta x = 0.1$;
X = -L:Δx:$L - \Delta x$;
LongX = length(X);

$T_{\max} = 10$;
$\Delta t = \dfrac{3}{2\pi^2}\left(\dfrac{\pi \Delta x}{L}\right)^3$;
$T_{graph} = 0.1$;
$j_{graph} = \text{floor}\left(\dfrac{T_{graph}}{\Delta t}\right) + 1$;

Initialization

Compteur = 1;
b = $\dfrac{1}{2h_0}\sqrt{\dfrac{3a}{h_0}}$;
$U_0 = a\text{sech}^2\,(bX))$;
$U_0 = U_0'$;
fftU_0 = fft(U_0);
N = $\dfrac{1}{2}$length(U_0);
ν = [0:N,-N + 1:-1];
$\nu = \nu'$;

$U_1 = U_0 - \dfrac{3i\Delta t\pi c_0}{Lh_0}U_0$.* ifft(fftU_0 .* ν) + $\dfrac{3i\pi^3 h_0^2 c_0}{3L^3}$ifft(fft$U_0$.* sin($\Delta t\nu$.^ 3));
U_1 = real(U_1);
Udouble = [U_0 U_1];
Udata=zeros(0,0);

Temporal loop

for j = 3:floor$\left(\frac{T_{\max}}{\Delta t}\right)+1$

fftUdouble=fft(Udouble(:,2));

$$U = \text{Udouble}(:,1) \quad -\frac{3i\Delta t\pi c_0}{Lh_0}\text{Udouble}(:,2) \text{ .* ifft(fftUdouble .* } \nu)$$

$$+ \frac{3i\pi^3 h_0^2 c_0}{3L^3}\text{ifft(fftUdouble .* sin}(\Delta t\nu \text{ .\^{} } 3 \text{));}$$

U = real(U);

Udouble = [Udouble(:,2) U];

if mod(j,j_{graph}) == 0

Compteur=Compteur+1

Udata = [Udata U];

if max(max(Udata))¿100

break;

end;

end;

end;

Temps2calcul=cputime-Temps2calcul;

Chapter 15

The Matlab code of the optimization algorithm

We give here the optimization algorithm described in the report. It is divided into height parts :

1. the fKdV solver $\zeta(b)$;
2. the adjoint solver $v(\zeta_b)$;
3. the evaluation of the functionnal $J(b)$;
4. the evaluation of the lagrangian $\mathcal{L}(b, \lambda)$;
5. the computation of the shape derivative $dJ(b)$;
6. the computation of the lagrangian derivative $d\mathcal{L}(b, \lambda)$;
7. the projecteur sur l'espace des λ ;
8. le solveur du problme d'optimisation.

15.1 The fKdV solver

function Udata=fKdV(F,h_0,c_0,F_r,Δx,Δt,$T_{\max}$,X,longX,L)

U = 0.X';
Udata=U;
$F_{xx} = \dfrac{1}{\Delta x^2}$ [F(2), - 2 F(1) diff(F,2), - 2 F(end) + F(end-1)];

for j=1:floor$\left(\dfrac{T_{\max}}{\Delta t}\right)$

$A_{i\pm2} = -\dfrac{h_0^2}{6\Delta x^4}$;
$A_{i-2} = [A_{i\pm2}$ ones(longX-2,1);0;0];
$A_{i+2} =$ [0;0;$A_{i\pm2}$ ones(longX-2,1)];
$A_{i-1} = \dfrac{1}{c_0 \Delta t \Delta x} + \dfrac{F_r - 1}{\Delta x^2} - \dfrac{3}{2h_0\Delta x^2}$ U(1:end-1) $+ \dfrac{2h_0^2}{3\Delta x^4}$;

A_{i-1} = [A_{i-1};0];
$A = -\dfrac{2\,(F_r-1)}{\Delta x^2} + \dfrac{6U}{2h_0\Delta x^2} - \dfrac{h_0^2}{\Delta x^4}$;
$A_{i+1} = -\dfrac{1}{c_0\Delta t\Delta x} + \dfrac{F_r-1}{\Delta x^2} - \dfrac{3}{2h_0\Delta x^2}$ U(2:end) $+\dfrac{2h_0^2}{3\Delta x^4}$;
A_{i+1} = [0;A_{i-1}];

C = spdiags(A_{i-2},-2,longX,longX);
C = C + spdiags(A_{i-1},-1,longX,longX);
C = C + spdiags(A,0,longX,longX);
C = C + spdiags(A_{i+2},1,longX,longX);
C = C + spdiags(A_{i+2},2,longX,longX);

B = F_{xx}';
B = B + $\dfrac{h_0^2}{6\Delta x^4}$[0;0;U(1:end-2)];
B = B + $\dfrac{1}{c_0\Delta t\Delta x} - \dfrac{F_r-1}{\Delta x^2} - \dfrac{2h_0^2}{3\Delta x^4}$[0;U(1:end-1)];
B = B + $\dfrac{2\,(F_r-1)}{\Delta x^2} + \dfrac{h_0^2}{\Delta x^4}$U;
B = B - $\left[\dfrac{1}{c_0\Delta t\Delta x} + \dfrac{F_r-1}{\Delta x^2} = \dfrac{2h_0^2}{3\Delta x^4}\right]$[U(2:end);0];
B = B + $\dfrac{h_0^2}{6\Delta x^4}$[U(3:end);0;0];

U = C \ B;
U = U .* ($\dfrac{1}{2}\left[1+\cos\left(\dfrac{X'+L-20}{20}\right)\right]$.*(X'⩾-L).*(X'⩽-L+20) + (X' ¿ -L+20).*(X'⩽L));
Udata = [Udata U];

if max(max(U))⩾ 1000
    break;
end;

end;

15.2 The adjoint solveur

function Vdata=adjoint(Udata,h_0,c_0,F_r,Δx,X,LongX,Δt,L)

$j_{\max}$ = size(Udata);
$j_{\max} = j_{max}(2)$;
V=0.X';
Vdata=V;
$U_x = \dfrac{1}{2\Delta x}$[Udata(2,:);Udata(3:end,:)-Udata(1:end-2,:);-Udata(end-1,:)];
for j = 1:$j_{\max}$ - 1
    $A_{i\pm2} = -\dfrac{h_0^2}{12\Delta x^4}$;
    A_{i-2} = [$A_{i\pm2}$.ones(longX-2,1);0;0];
    A_{i+2} = [0;0;$A_{i\pm2}$.ones(longX-2,1)];

$$A_{i-1} = \frac{-1}{2c_0\Delta t\Delta x} + \frac{F_r - 1}{2\Delta x^2} + \frac{3}{8h_0\Delta x} U_x(\text{2:end},j_{\max}\text{-j}) - \frac{3}{4h_0\Delta x^2}\ \text{Udata(2:end},j_{\max}\text{-j)} + \frac{h_0^2}{3\Delta x^4};$$

A_{i-1} = [A_{i-1};0];

$$A = -\frac{F_r - 1}{\Delta x^2} + \frac{3}{2h_0\Delta x^2}\ \text{Udata(:},j_{\max}\text{-j)} - \frac{h_0^2}{2\Delta x^4};$$

$$A_{i+1} = \frac{0.5}{c_0\Delta t\Delta x} + \frac{F_r - 1}{2\Delta x^2} - \frac{3}{8h_0\Delta x} U_x(\text{1:end-1},j_{\max}\text{-j}) - \frac{0.75}{h_0\Delta x^2}\ \text{Udata(1:end-1},j_{\max}\text{-j)} + \frac{h_0^2}{3\Delta x^4};$$

A_{i+1} = [0;A_{i+1}];

C = spdiags(A_{i-2},-2,LongX,LongX);
C = C + spdiags(A_{i-1},-1,LongX,LongX);
C = C + spdiags(A,0,LongX,LongX);
C = C + spdiags(A_{i+1},1,LongX,LongX);
C = C + spdiags(A_{i+2},2,LongX,LongX);

$$\text{B} = -\frac{1}{2}\left(U_x(:, j_{\max} - j) + U_x(:, j_{\max} - j + 1)\right);$$

$$\text{B} = \text{B} + \frac{h_0^2}{12\Delta x^4}[\text{0;0;V(1:end-2)}];$$

$$\text{B} = \text{B} + \left[-\frac{1}{2c_0\Delta t\Delta x} - \frac{F_r - 1}{2\Delta x^2} - \frac{3}{8h_0\Delta x}U_x(:, j_{\max} - j + 1)\right.$$

$$\left. + \frac{3}{4h_0\Delta x^2}\ \text{Udata}(:, j_{\max} - j + 1) - \frac{h_0^2}{3\Delta x^4}\right] .\!*\ [\text{0;V(1:end-1)}];$$

$$\text{B} = \text{B} + \left[\frac{F_r - 1}{\Delta x^2} - \frac{3}{2h_0\Delta x^2}\ \text{Udata}(:, j_{\max} - j + 1) + \frac{h_0^2}{2\Delta x^4}\right] .\!*\ \text{V};$$

$$\text{B} = \text{B} + \left[\frac{1}{2c_0\Delta t\Delta x} - \frac{F_r - 1}{2\Delta x^2} + \frac{3}{8h_0\Delta x}U_x(:, j_{\max} - j + 1)\right.$$

$$\left. + \frac{3}{4h_0\Delta x^2}\ \text{Udata}(:, j_{\max} - j + 1) - \frac{h_0^2}{3\Delta x^4}\right] .\!*\ [\text{V(2:end);0}];$$

$$\text{B} = \text{B} + \frac{h_0^2}{12\Delta x^4}[\text{V(3:end);0;0}];$$

V = C\B;

$$\text{V} = \quad \text{V} .\!*\ \frac{1}{2}\left[1 + \cos\left(\frac{\pi(X' + L - 20)}{20}\right)\right] .*(X' \geqslant -L).*(X' \leqslant -L + 20)$$

$$+(X' > -L + 20).*(X' < L - 20)$$

$$+\frac{1}{2}\left[1 + \cos\left(\frac{\pi(X' - L + 20)}{20}\right)\right] .*(X' >= L - 20).*(X' <= L));$$

Vdata = [V Vdata];

if max(max(V)) $\geqslant$ 1000
    break;
end;

end;

15.3 Evaluation of the functionnal

function Jb = functionnal(Udata,$T_{\max}$,Δt,Δx,LongX)

S = zeros(1,floor$\left(\frac{T_{\max}}{\Delta t}\right)$ + 1);
S(1) = 1;
S(2:2:end-1) = 4 ones(1,$\frac{1}{2}$floor$\left(\frac{T_{\max}}{\Delta t}\right)$);
S(3:2:end-2) = 2 ones(1,$\frac{1}{2}$floor$\left(\frac{T_{\max}}{\Delta t}\right)$-1);
S(end) = 1;
S = $\left(\frac{\Delta t}{3}\right)$ ones(longX,1) S;

I = S .* Udata .^ 2;
I=sum(I');

SS = zeros(1,LongX);
SS(1) = 1;
SS(2:2:end-1) = 4 ones(1,floor$\left(\frac{LongX-1}{2}\right)$;
SS(3:2:end-2) = 2 ones(1,floor$\left(\frac{LongX-1}{2}\right)$) - 1);
SS(end) = 1;
SS = $\frac{\Delta x}{3}$ SS;

Jb = SS .* I;
Jb = sum(Jb);
Jb = - $\frac{1}{2}$ Jb;

15.4 Evaluation of the lagrangian

function Lag=lagrangien(F,Jb,λ,L,Δx,M,$supp$)

S = zeros(floor$\left(\frac{L+supp}{\Delta x}\right)$ - floor$\left(\frac{L-supp}{\Delta x}\right)$ + 1,1);
S(1) = 1;
S(2:2:end-1) = 4 ones(floor((floor$\left(\frac{L+supp}{\Delta x}\right)$) - $\frac{1}{2}$floor$\left(\frac{L-supp}{\Delta x}\right)$ + 1),1);
S(3:2:end-2) = 2 ones(floor((floor($\left(\frac{L+supp}{\Delta x}\right)$ - $\frac{1}{2}$floor$\left(\frac{L-supp}{\Delta x}\right)$ - 1,1);
S(end) = 1;
S = $\frac{\Delta x}{3}$ S;

I = F(floor$\left(\frac{L-supp}{\Delta x}\right)$ + 1:floor$\left(\frac{L+supp}{\Delta x}\right)$ + 1) .^ 2 S;
I = sum(I);

Lag = Jb - F(floor$\left(\dfrac{LongX - supp}{\Delta x}\right)$ + 1:floor$\left(\dfrac{LongX + supp}{\Delta x}\right)$) λ(1:end-1) + λ(end)*(I-M);

15.5 Computation of the shape derivative of the functionnal

function dJ = d_J(V_x,Δt,$T_{\max}$,X,LongX,$supp$)

S = zeros(1,floor$\left(\dfrac{T_{\max}}{\Delta t}\right)$ + 1);
S(1) = 1;
S(2:2:end-1) = 4 ones(1,$\frac{1}{2}$floor$\left(\dfrac{T_{\max}}{\Delta t}\right)$);
S(3:2:end-2) = 2 ones(1,$\frac{1}{2}$floor$\left(\dfrac{T_{\max}}{\Delta t}\right)$ - 1);
S(end) = 1;
S = $\dfrac{\Delta t}{3}$ ones(LongX,1) S;

I = S .* V_x;
I = sum(I');

dJ = $\frac{1}{2}$I .* (X $\geqslant -supp$) .* (X $\leqslant supp$);
dJ = dJ';

15.6 Computation of the lagrangian derivative

function dL=d_L(dJ,F,λ,L,Δx,$supp$)
dL = dJ - [zeros(floor$\left(\dfrac{L - supp}{\Delta x}\right)$,1);$\lambda$(1:end-1);zeros(floor$\left(\dfrac{L - supp}{\Delta x}\right)$,1)] + 2*$\lambda$(end) F';

15.7 Computation of the positive projector

function llambda=lambda_projete(FF,λ,L,Δx,M,κ,$supp$)
S = zeros(floor$\left(\dfrac{L + supp}{\Delta x}\right)$ - floor$\left(\dfrac{L - supp}{\Delta x}\right)$ + 1,1);
S(1) = 1;
S(2:2:end-1) = 4 ones(floor((floor$\left(\dfrac{L + supp}{\Delta x}\right)$) - $\frac{1}{2}$floor$\left(\dfrac{L - supp}{\Delta x}\right)$ + 1),1);
S(3:2:end-2) = 2 ones(floor((floor($\left(\dfrac{L + supp}{\Delta x}\right)$ - $\frac{1}{2}$floor$\left(\dfrac{L - supp}{\Delta x}\right)$ - 1,1);
S(end) = 1;
S = $\dfrac{\Delta x}{3}$ S;

I=FF(floor$\left(\dfrac{L - supp}{\Delta x}\right)$ + 1:floor$\left(\dfrac{L + supp}{\Delta x}\right)$ + 1) .^ 2 S;

I = sum(I);

llambda = max(0,λ(1:end-1)-κ FF(floor$\left(\dfrac{L - supp}{\Delta x}\right)$ + 1:floor$\left(\dfrac{L + supp}{\Delta x}\right)$ + 1)');
llambda = [llambda;max(0,λ(end) + κ (I-M))];

15.8 Optimization algorithm

Results=function optimisation()

clear all;
close all;
Running_time=cputime;

15.8.1 Initialization part

Data

h_0 = 1;
g = 9.81;
$c_0 = \sqrt{gh_0}$;
F_r = 1;

Constraints

M = 0.02;
κ = 30000;
supp = 1;

Spatial vector

Δx = 0.1;
L = 50;
X = -L:Δx:L;
LongX = length(X);

Temporal vector

Δt = 0.1;
$t_{\max}$ = 30;
T = 0:Δt:$T_{\max}$;
LongT = length(T);

Initial bottom

Fini = $\dfrac{0.1}{2}\left[1 + \cos(\pi X)\right]$.* (X $\geqslant -supp$) .* (X $\leqslant supp$);
F = Fini;
λ = zeros(floor$\left(\dfrac{L + supp}{\Delta x}\right)$ -floor$\left(\dfrac{L - supp}{\Delta x}\right)$ + 2,1);
Results=zeros(4,0);

Simson matrix

S = zeros(floor$\left(\dfrac{L+supp}{\Delta x}\right)$ - floor$\left(\dfrac{L-supp}{\Delta x}\right)$ + 1,1);
S(1) = 1;
S(2:2:end-1) = 4 ones(floor((floor$\left(\dfrac{L+supp}{\Delta x}\right)$) - $\dfrac{1}{2}$floor$\left(\dfrac{L-supp}{\Delta x}\right)$ + 1),1);
S(3:2:end-2) = 2 ones(floor((floor($\left(\dfrac{L+supp}{\Delta x}\right)$ - $\dfrac{1}{2}$floor$\left(\dfrac{L-supp}{\Delta x}\right)$ - 1,1);
S(end) = 1;
S = $\dfrac{\Delta x}{3}$ S;

Stop criteria

Stop = 0;
Compteur = 0;

15.8.2 Temporal loop

Udata = fKdV(F,h_0,c_0,F_r,Δx,Δt,$T_{\max}$,X,LongX,L);
while Stop==0
  Compteur=Compteur+1;

Computation of the lagrangian

Vdata = adjoint(Udata,h_0, c_0, F_r, Δx,X,LongX,Δt,L);
V_x = $\dfrac{1}{2\Delta x}$[Vdata(2,:);Vdata(3:end,:)-Vdata(1:end-2,:);-Vdata(end-1,:)];
Jb = functionnal(Udata,$T_{\max}$, Δt, Δx,LongX);
Lag = lagrangien(F,Jb,λ,L,Δx,M,$supp$);

Resolution of the minimization problem without constraints

dJ = d_J(V_x, Δt, $T_{\max}$,X,LongX,$supp$);
dL = d_L(dJ,F,λ,L,Δx,$supp$);
ε_{ini} = 10 ^ (floor(log10(mean(abs(F))))-floor(log10(mean(abs(dL)))) - 2);
if max(λ) == 0
  ε_{ini} = 10 ^ (floor(log10(mean(abs(F))))-floor(log10(mean(abs(dL)))) - 1);
end;

F2 = (F - ε_{ini} dL') .* (X $\geqslant -supp$) .* (X$\leqslant$ $supp$);
Udata2 = fKdV(F2,h_0,c_0,F_r,Δx,Δt,$T_{\max}$,X,LongX,L);
Jb2 = functionnal(Udata2,$T_{\max}$, Δt, Δx,LongX);
Lag2 = lagrangien(F2,Jb2,λ,L,Δx,M,$supp$);

F3 = (F + ε_{ini} dL') .* (X $\geqslant -supp$) .* (X$\leqslant$ $supp$);
Udata3 = fKdV(F3,h_0,c_0,F_r,Δx,Δt,$T_{\max}$,X,LongX,L);
Jb3 = functionnal(Udata3,$T_{\max}$, Δt, Δx,LongX);
Lag3 = lagrangien(F3,Jb3,λ,L,Δx,M,$supp$);

P = polyfit([-ε_{ini},0,ε_{ini}],[Lag3,Lag,Lag2],2);

if P(1) ¿ 0

$\varepsilon_{opt} = -\dfrac{P(2)}{2P(1)}$;

if $(\varepsilon_{opt} \geqslant -\varepsilon_{ini})$&&$(\varepsilon_{opt} \leqslant \varepsilon_{ini})$

$\varepsilon = \varepsilon_{opt}$;

end;

if $(\varepsilon_{opt} < -\varepsilon_{ini})$

$\varepsilon = -\varepsilon_{ini}$;

end;

if $(\varepsilon_{opt} > \varepsilon_{ini})$

$\varepsilon = \varepsilon_{ini}$;

end;

end;

if P(1) ¡ 0

if Lag3 $\leqslant$ Lag2

$\varepsilon = -\varepsilon_{ini}$;

end;

if Lag3 ¿ Lag2

$\varepsilon = \varepsilon_{ini}$;

end;

end;

FF = (F - ε dL') .* (X $\geqslant -supp$) .* (X$\leqslant$ $supp$);

UUdata=fKdV(FFh_0,c_0,F_r,Δx,Δt,$T_{\max}$,X,LongX,L);

JJb=functionnal(UUdata,$T_{\max}$, Δt, Δx,LongX);

LLag=lagrangien(FF,JJb,λ,L,Δx,M,$supp$);

Projection on the positive space

llambda=lambda_projete(FF,λ,L,Δx,M,κ, $supp$);

Results=[Results [Jb;Lag;λ(end);ε;max(max(Udata))]];

Stop criteria

I = (FF$\left(\dfrac{L - supp}{\Delta x}\right)$+1:floor$\left(\dfrac{L + supp}{\Delta x}\right)$.^ 2) S;

I = sum(I);

if (max(abs(FF-F)) $\leqslant 10^{-5}$) & & (max(abs(λ - llambda)) $\leqslant 10^{-1}$) & & (I $\leqslant$ M) & & (min(FF) $\geqslant$ 0)

Stop = 1;

end;

Actualization

F=FF;

Udata=UUdata;

λ=llambda;

end;

Running_time=cputime-Running_time;

Bibliography

[1] L.H. Wiryanto & A. Achirul. An implicit finite difference method for a forced KdV equation.

[2] P. G. Baines. *Topographic Effects in Stratified Flows.* Cambridge Monographs on Mechanics, 1998.

[3] M. Chen. Equations for bi-directional waves over an uneven bottom. *Mathematics and Computers in Simulation*, 62(1-2):3–9, February 2003.

[4] D. Zhang & A.T. Chwang. Numerical study of nonlinear shallow water waves produced by a submerged moving disturbance in viscous flow. *Physics of Fluids*, 8(1):147–155, January 1996.

[5] D. Zhang & A.T. Chwang. On solitary waves forced by underwater moving objects. *Journal of Fluids Mechanics*, 389:119–135, 1999.

[6] F. Conrad. Cours d'optimisation. Ecole Nationale Supérieure des Mines de Nancy.

[7] E.M. de Jager. On the origin of the Korteweg-de Vries equation. Korteweg-de Vries Institute, University of Amsterdam, February 2006.

[8] G.M. Muslu & H.A. Erbay. A split-step Fourier method for the complex modified Korteweg-de Vries equation. *Computers and Mathematics with Applications*, 45:503–514, February 2003.

[9] T. Igushi. A mathematical justification of the forced Korteweg-de Vries equation for capillary-gravity waves. *Kyushu Journal of Mathematics*, 60(2):267–303, 2006.

[10] P.G. Drazin & R.S. Johnson. *Solitons : an introduction.* Cambridge University Press, 1996.

[11] R.S. Johnson. *A Modern Introduction to the Mathematical Theory of Water Waves.* Cambridge University Press, 1997.

[12] N.J. Zabusky & M.D. Kruskal. Interaction of solitons in a collisionless plasma and the recurrence of initial states. *Physical Review letters*, 15(6):240–243, August 1965.

[13] Y. Cao & R.F. Beck & W.W. Schultz. Numerical computations of two-dimensional solitary waves generated by moving disturbances. *International Journal for Numerical Methods in Fluids*, 17(10):905–920, November 1993.

[14] D. Smets. Méthodes mathématiques pour la mécanique des fluides. Notes de cours pour les étudiants de l'Ecole Normale Supérieure, January 2010.

[15] L.N. Trefethen. *Spectral methods in Matlab*, chapter 10, pages 110–113. Society for Industrial and Applied Mathematics, 2000.

[16] M.A. Walkley. *A Numerical Method for Extended Boussinesq Shallow-Water Waves Equations*. PhD thesis, The University of Leeds, School of Computer Studies, September 1999.

[17] E. Hairer & G. Wanner. *Solving Ordinary Differential Equations, Stiff and Differential-Algebraic Problems*, chapter 4, pages 2–253. 1991.

[18] E. Hairer & S.P. Norsett & G. Wanner. *Solving Ordinary Differential Equations, Nonstiff Problems*, chapter 2, pages 132–142. 2008.

[19] B. Fornberg & G.B. Whitham. A numerical and theoretical study of certain nonlinear wave phenomena. *Philosophical Transactions*, 289(1361):373–404, May 1978.

[20] T.Y. Wu. Generation of upstream advancing solitons by moving disturbances. *Journal of Fluid Mechanics*, 184:75–99, 1987.

[21] J.L. Bona & B. Zhang. The initial-value problem for the forced Korteweg-de Vries equation. September 1994.