



UNIVERSITÉ DE TECHNOLOGIE DE COMPIÈGNE

RAPPORT DE STAGE DE FIN D'ÉTUDES

MODÉLISATION ET SIMULATION NUMÉRIQUE DES VAGUES NON LINÉAIRES

Simon Chabot

GI06

Sous la direction de :

M. DIDIER CLAMOND	Univ. Nice-Sophia Antipolis	Tuteur
M. STÉPHANE MOTTELET	U.T.C.	Enseignant suiveur

Février 2014 - Juillet 2014

Publié sous licence *Creative-Commons BY*



JOSEPH CHAN © 2020

published under *Creative-Commons BY* license

Vous êtes autorisé à :

- *partager* — copier, distribuer et communiquer le matériel par tous moyens et sous tous formats
- *adapter* — remixer, transformer et créer à partir du matériel pour toute utilisation, y compris commerciale.

Selon les conditions suivantes :

- *attribution* — vous devez créditer l'Œuvre, intégrer un lien vers la licence et indiquer si des modifications ont été effectuées à l'Œuvre. Vous devez indiquer ces informations par tous les moyens possibles mais vous ne pouvez pas suggérer que l'Offrant vous soutient ou soutient la façon dont vous avez utilisé son Œuvre.
- *pas de restrictions supplémentaires* — vous n'êtes pas autorisé à appliquer des conditions légales ou des mesures techniques qui restreindraient légalement autrui à utiliser l'Œuvre dans les conditions décrites par la licence.

Notes — Vous n'êtes pas dans l'obligation de respecter la licence pour les éléments ou matériel appartenant au domaine public ou dans le cas où l'utilisation que vous souhaitez faire est couverte par une exception. Aucune garantie n'est donnée. Il se peut que la licence ne vous donne pas toutes les permissions nécessaires pour votre utilisation. Par exemple, certains droits comme les droits moraux, le droit des données personnelles et le droit à l'image sont susceptibles de limiter votre utilisation.

You are free to :

- *share* — copy and redistribute the material in any medium or format
- *adapt* — remix, transform, and build upon the material, for any purpose, even commercially.

Under the following terms:

- *attribution* — you must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in a way that suggests the licensor endorses you or your use.
- *no additional restrictions* — you may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices — You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation. No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

REMERCIEMENTS

Si j'ai pu réaliser ce stage, c'est grâce à plusieurs personnes que j'aimerais ici remercier.

Je tiens tout d'abord à remercier Didier Clamond pour m'avoir accepté en stage au sein du laboratoire mathématiques de l'université de Nice. Didier a été tout au long de ce stage à mes côtés, à répondre à mes questions et à me faire partager sa passion pour le domaine des vagues. Je le remercie aussi pour m'avoir fait assister à de multiples séminaires, *workshops* et soutenances (thèses / HDR).

Je remercie également Stéphane Mottelet qui a, une fois de plus, accepté de me suivre. Je le remercie également pour avoir fait le voyage de Compiègne à Nice pour venir me voir et me permettre de réaliser une soutenance préliminaire, nécessaire à ma candidature en thèse.

Et enfin, je remercie Daniel et mon amie, Anaïs, pour m'avoir relu =)

CONTENTS

CONTENTS	x
INTRODUCTION	1
1 NONLINEAR WATER-WAVES EQUATIONS	3
1.1 MATHEMATICAL MODEL	3
1.1.1 Notations	4
1.2 SOME CLASSICAL NONLINEAR EQUATIONS	4
1.2.1 Korteweg-de Vries equation	4
1.2.2 Saint-Venant equations (or Shallow-Waters equations)	6
1.2.3 The Serre Green Naghdi equations	6
1.3 SUM UP	7
2 NUMERICAL RESOLUTION OF THE WATER-WAVES EQUATIONS	9
2.1 PSEUDO-SPECTRAL METHODS	9
2.1.1 Motivation	9
2.1.2 Why “pseudo” ?	10
2.1.3 The Korteweg-de Vries equation in the Fourier space	11
2.2 RUNGE-KUTTA METHODS	11
2.2.1 The adaptive time-step	12
2.2.2 Dense Output	14
2.2.3 An example with the Korteweg-de Vries equation	14
2.3 THE INTEGRATING FACTOR	15
2.3.1 with the Korteweg-de Vries equation	15
2.4 THE MODIFIED INTEGRATING FACTOR	16
2.4.1 Choosing the best $P(t)$ function	16
2.4.2 Calculating the time derivatives	17
2.4.3 Substituting y back	17
2.5 SOME RESULTS	18
3 DERIVATION OF NONLINEAR WATER-WAVE EQUATIONS WITH A VARIABLE BOTTOM	21
3.1 EULER EQUATIONS	21
3.2 THE DEPTH-AVERAGED EULER EQUATIONS	22
3.2.1 Integrating the continuity equation	22
3.2.2 Integrating the horizontal momentum equation	22
3.2.3 Integrating the vertical momentum equation	23
3.2.4 Final depth-averaged equations	24
3.3 MODELS	24
3.3.1 Saint Venant equations (or Shallow-Water equations)	24
3.3.2 Serre Green-Naghdi equations	25

3.4 AN EXAMPLE OF A LINEARIZATION	28
CONCLUSION	31
A THE PROOF OF SOME IDENTITIES	33
A.1 THE DIVERGENCE OF A TENSOR PRODUCT	33
A.2 A LEIBNIZ RULE-LIKE FOR A TOTAL DERIVATIVE	33
BIBLIOGRAPHY	35
LIST OF FIGURES	35

INTRODUCTION

EN FRANÇAIS

Les vagues extrêmes (tsunamis, vagues scélérates, etc) jouent un rôle important dans l’environnement et les risques naturels. La compréhension de ces phénomènes passe, entre autres, par la simulation numérique. Les tsunamis se caractérisent généralement par des vagues ayant des longueurs d’onde entre 10 et 500 km et des périodes de plus d’une heure. La profondeur des océans étant typiquement de 3 000 à 4 000 mètres, ces vagues extrêmes peuvent être décrites par la théorie dite *des vagues en eaux peu profondes*, comme nous le verrons dans ce rapport.

Au lieu de génération du tsunami, en océan profond, le niveau de l’eau s’élève de quelques mètres à peine ce qui explique que la perturbation n’est généralement pas ressentie par les navires à ce niveau là, étant donné la longueur d’onde du phénomène. Au fur et à mesure que l’onde se déplace – à une vitesse très élevée¹ – vers les côtes, la profondeur diminue ce qui augmente la hauteur des vagues.

La propagation de ces ondes se fait sur des centaines voire des milliers de kilomètres avec une perte d’énergie infime. Afin de capturer au mieux cette physique nous devons utiliser des méthodes très précises qui permettent de conserver cette énergie tout au long de la simulation. Nous devons ainsi avoir des erreurs numériques aussi faibles que possibles et une dissipation numérique quasiment nulle. Pour ces raisons, nous faisons appel aux méthodes spectrales qui vont nous permettre de réaliser des simulations tout en tenant compte de ces contraintes fortes.

Ce rapport expose une partie des travaux que j’ai été amenés à faire lors de mon stage au laboratoire de mathématiques Jean Alexandre Dieudonné de l’université de Nice. Il est divisé en trois parties. La première vise à présenter différentes équations classiques décrivant le mouvement des vagues. Ces équations seront présentées sous leur forme la plus simple, c’est à dire avec l’hypothèse d’un fond horizontal. Dans la seconde partie, nous nous intéresserons aux méthodes numériques permettant les simulations précises dont nous avons besoin. Nous exposerons notamment le principe des méthodes spectrales et les travaux de Brice Eichwald sur le Facteur Intégrant Modifié. Enfin la troisième partie sera consacrée à la dérivation des équations des vagues avec un fond variable en temps *et* en espace. Les équations de Serre – Green-Nagdhi seront ainsi généralisées.

¹ $c \approx \sqrt{g \times d} \approx \sqrt{9.81 \times 3000} \approx 617$ km/h en plein océan.

IN ENGLISH

Extreme waves (tsunamis, rogue waves, etc) play an important role in the environment and natural hazards. Their understanding requires, *inter alia*, numerical simulation. Tsunamis are usually characterized by waves having a wavelength between 10 and 500 km and a period over an hour. The ocean's depth is about 3 000 to 4 000 meters, therefore those extreme waves can be described by the so-called *shallow-water theory*, as we will see in this report.

At the place where the tsunami occurs, in the deep ocean, the water level increases of a few meters only, explaining why ships, at this same place, usually do not feel the perturbation, given the wavelength of the phenomena. As the wave goes – at a very high speed² – towards the coasts, the ocean's depth decreases and the wave-height increases.

The wave propagates over hundreds even thousands kilometers with a very tiny energy loss. To capture this physics as best as possible, very precise methods have to be used so that this energy is conserved all over the simulation. Numerical error must be as small as possible and numerical dissipation nearly zero. For those reasons, spectral methods are used, enabling us to perform simulations under those strong constraints.

This report relates a part of the work I have conducted during my internship at the laboratory of mathematics Jean Alexandre Dieudonné of the university of Nice. It is divided into three parts. The first one introduces some classical equations describing extreme waves. The equations are given in their simplest form, that is to say a flat bottom is assumed. The second part shows the numerical methods we use for the simulations. The spectral methods and the work of Brice Eichwald on the Modified Integrating Factor are explained in particular. Finally, the last part is dedicated to the derivation of the wave equations with a variable bottom in space *and* time. Serre – Green-Nagdhi equations are thus generalized.

² $c \approx \sqrt{g \times d} \approx \sqrt{9.81 \times 3000} \approx 617 \text{ km/h}$ in deep ocean.

NONLINEAR WATER-WAVES EQUATIONS

1

This short chapter aims to introduce different classical nonlinear water-waves equations which are manipulated in the report. But first, we explain the mathematical model of the water-wave and the notations we use in the whole report.

1.1 MATHEMATICAL MODEL

We consider an ideal incompressible fluid of constant density ρ . The fluid is bounded by a free surface and the bottom of the sea. We use a Cartesian coordinate system, such that the free surface $y = 0$ corresponds to the still water level. The horizontal independent variables are denoted by $x = (x_1, x_2)$ and the upward vertical one by y . The free surface is given by $y = \eta(x, t)$, and the bottom is defined by $y = -d(x, t)$. Finally, we denote by u the horizontal component of the fluid's velocity and by v the vertical one.

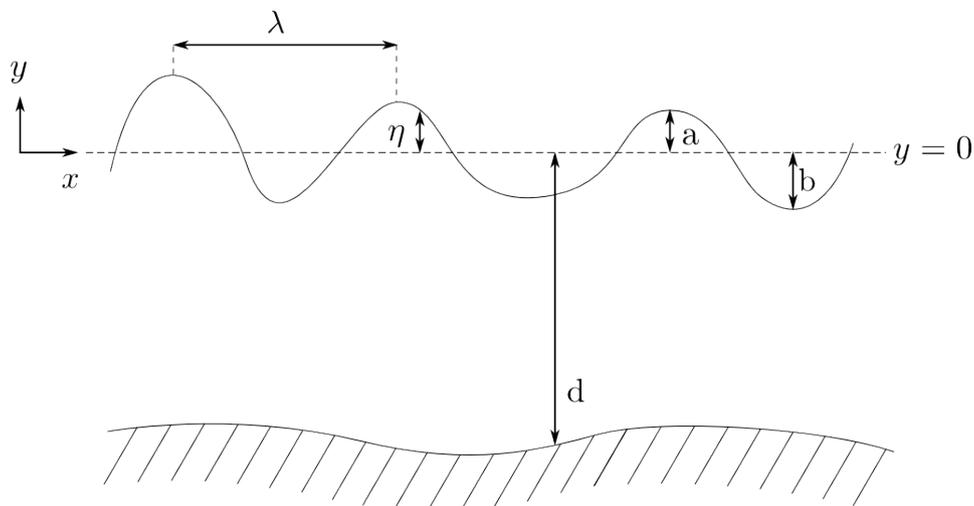


Figure 1.1 – domain definition

As illustrated on the sketch 1.1, we call :

- λ the wavelength of the wave ;
- a the amplitude from the wave crest to the still level ;
- b the amplitude from the still level to the troughs of the wave ;

- $H = a + b$ the amplitude from the wave crest to the troughs ;
- $h = d + \eta$ the height of the fluid. It is assumed to be positive for all times, $h > 0, \forall t$.

1.1.1 Notations

In all the present document, we will use the following notations :

- \bar{f} is the function f depth-averaged. The definition used is :

$$\bar{f}(x, t) = \frac{1}{h(x, t)} \int_{-d(x, t)}^{\eta(x, t)} f(x, y, t) dy \quad (1.1)$$

- ϕ_s and ϕ_b are the physical quantity ϕ at the free surface ($y = \eta$) and at the bottom ($y = -d$) respectively.
- ∇ is the following derivative operator $\nabla = (\partial_{x_1}, \partial_{x_2})$

1.2 SOME CLASSICAL NONLINEAR EQUATIONS

In this section, we introduce three classical nonlinear water-waves equations :

- the Korteweg-de Vries (KdV) equation ;
- the Shallow-Water (SW) equations ;
- the Serre Green-Nagdhi (SGN) equations.

For simplification purpose, the equations are given in two dimensions (i.e. one horizontal) and the bottom is assumed to be horizontal (i.e. $d(x, t) = \text{const}$).

1.2.1 Korteweg-de Vries equation

The KdV equation is a mathematical model of shallow water-waves which is exactly solvable (which is interesting for testing purpose, for instance). It has been discovered by Boussinesq in 1877, and rediscovered by Diederik Korteweg and Gustav de Vries in 1895. It may be written as follow ([Eichwald 2013](#)) :

$$\partial_t \eta + c_0 \partial_x \eta + \alpha \eta \partial_x \eta + \beta \partial_x^3 \eta = 0 \quad (1.2)$$

where $\alpha = \frac{3}{2} \sqrt{gd}$, $\beta = \frac{d^2 c_0}{6}$ and $c_0 = \sqrt{gd}$.

This equation admits, among others, two types of solutions : a cnoidal wave solution and a solitary wave solution.

The cnoidal wave solution

The periodic-cnoidal wave solution can be written as follow :

$$\eta = \frac{H}{m} \left(1 - m - \frac{E(m)}{K(m)} \right) + H \operatorname{cn}^2 \left(2K(m) \frac{x - ct}{\lambda} \middle| m \right) \quad (1.3)$$

where cn is the Jacobian elliptic function with parameter m ($0 \leq m \leq 1$), and $K(m)$ and $E(m)$ are the complete elliptic integrals of the first and second kind respectively.

The speed c , the wavelength λ and the period τ are linked by the following relations :

$$c = \sqrt{gh} \left[1 + \frac{H}{md} \left(1 - \frac{1}{2}m - \frac{3E(m)}{2K(m)} \right) \right]$$

$$\lambda = d \sqrt{\frac{16md}{3H}} K(m)$$

$$\tau = \frac{\lambda}{c}$$

The figure 1.2 show the profile of the wave for different values of m .

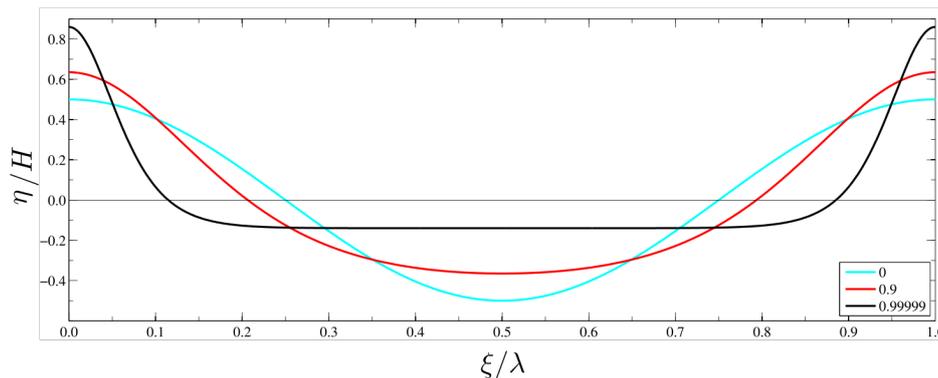


Figure 1.2 – Cnoidal wave profiles for different values of the parameter m ($m = 0$ (cyan), $m = 0.9$ (red), $m = 0.99999$ (black)). The grey curve represents the still water level. (Author [Kraaiennest](#), CC-BY-SA)

The solitary wave solution

When the m parameter of the cnoidal wave solution tends to 1, the wavelength tends to infinity. That is to say, a solitary wave is found. A solitary wave is a wave propagating on a long distance, without any deformation.

The solitary wave solution of the equation 1.2 is given by :

$$\eta = \frac{H}{\cosh^2 \left(\frac{\kappa}{2d} (x - ct) \right)} \quad (1.4)$$

where H is the amplitude of the wave, $\kappa = \sqrt{\frac{3H}{d}}$, the speed is $c = c_0 \left(1 + \frac{H}{2d} \right)$.

1.2.2 Saint-Venant equations (or Shallow-Waters equations)

The SW equations are a set of hyperbolic partial differential equations describing the motion of water waves when the wavelength is much higher than the depth $d(x, t)$. That is to say, when the velocity is mainly horizontal and the vertical component of the velocity can be neglected. The SW equations, for a horizontal bottom are :

$$\begin{cases} \partial_t h + \partial_x (h\bar{u}) = 0 \\ \partial_t (h\bar{u}) + \partial_x (h\bar{u}^2 + \frac{1}{2}gh^2) = 0 \end{cases} \quad (1.5)$$

In 3.3.1, we show how to obtain this set of equations, in three dimensions, for any given bottom.

1.2.3 The Serre Green Naghdi equations

Serre (1953) has developed a wider theory than Saint-Venant. Instead of neglecting the vertical component, he assumed that the vertical velocity was linear from the seabed to the free surface¹. This ansatz permits the vertical acceleration of the fluid to be taken into account. Those equations have been rediscovered by Green, Laws and Naghdi. Finally, Fernando J. Seabra-Santos et al. (1987) generalized those equations to any given bottom (variable in space only, not in time). In 3.3.2 we show how to obtain the SGN equations, in three dimensions, for any given bottom.

The SGN equations, for a horizontal bottom, in two dimensions, are :

$$\begin{cases} \partial_t h + \partial_x (h\bar{u}) = 0 \\ \partial_t (h\bar{u}) + \partial_x \left(h\bar{u}^2 + \frac{g}{2}h^2 + \frac{\Gamma_s h^2}{3} \right) = 0 \end{cases} \quad (1.6)$$

where Γ_s represents the acceleration of the fluid at the free surface. We have $\Gamma_s = h \left((\partial_x \bar{u})^2 - \partial_{xt} \bar{u} - \bar{u} \partial_{xx} \bar{u} \right)$.

As for the KdV equations, this set of equations admits a solitary wave and a cnoidal wave as solutions when the bottom is horizontal.

The cnoidal wave solution

There exists a $(2\pi/k)$ -periodic cnoidal wave solution :

$$\eta = a \frac{\operatorname{dn}^2(0.5\chi(x-ct)|m) - \frac{E(m)}{K(m)}}{1 - \frac{E(m)}{K(m)}} = a - H \operatorname{sn}^2(0.5\chi(x-ct)|m)$$

$$\bar{u} = \frac{c\eta}{d + \eta}$$

where dn and sn are the Jacobian elliptic functions with parameter m ($0 \leq m \leq 1$), and $K(m)$ and $E(m)$ are the complete elliptic integrals of the first and second kind respectively.

The parameters are linked by those relations :

¹i.e., the incompressibility is imposed.

$$\begin{aligned}
 k &= \frac{\pi\chi}{2\mathbf{K}(m)} & H &= \frac{ma\mathbf{K}(m)}{\mathbf{K}(m) - \mathbf{E}(m)} \\
 (\chi d)^2 &= \frac{3gH}{mc^2} & m &= \frac{gH(d+a)(d+a-H)}{g(d+a)^2(d+a-H) - d^2c^2}
 \end{aligned}$$

The solitary wave solution

As $m \rightarrow 1$, $\lambda \rightarrow \infty$ and therefore a solitary wave solution exists, and is given by this set of relations :

$$\begin{aligned}
 \eta &= \frac{a}{\cosh^2(0.5\chi(x-ct))} & \bar{u} &= \frac{c\eta}{d+\eta} \\
 c &= \sqrt{g(d+a)} & (\chi d)^2 &= \frac{3a}{d+a}
 \end{aligned}$$

1.3 SUM UP

Different equations have been introduced. In this report, the KdV equation will be used as our *toy-model*, while the SW and the SGN equations will be generalized to a variable bottom in space and time. (Of course, the particular solutions (solitary and cnoidal) do not hold when the bottom is not horizontal.)

NUMERICAL RESOLUTION OF THE WATER-WAVES EQUATIONS

This chapter is dedicated to the numerical resolution of the water-waves equations. We illustrate the procedure with the resolution of the KdV equation, which is the simpler mathematical model of non-linear water-waves.

The first part of this chapter briefly explains the principle of pseudo-spectral methods. In the second part, we talk about the Runge-Kutta (RK) method, used to approximate the solution. We emphasize the need of an adaptive time-step and explain the use of the dense-output. The third part deals with the Integrating Factor (IF). The IF uses a clever variable substitution to improve the time-step integration of the RK algorithm. We finish this chapter with the Modified Integrating Factor (MIF), introduced by [Eichwald \(2013\)](#), which is an improvement of the IF for strongly non-linear water-waves equations.

Because the equations are strongly non-linear and solved with a non-local method, we need to compute the solution over the whole domain. Because the waves we are studying can spread over huge distance, we need fast solving-methods. Moreover, because the simulation runs over numerous wave periods, we need an accurate¹ method to capture all the physic of the equation and to loose as least as possible in numerical errors. The methods we introduce, IF and MIF, are ad-hoc methods developed for that purpose.

2.1 PSEUDO-SPECTRAL METHODS

2.1.1 Motivation

Our purpose is to solve, numerically, non-linear equations. There exists different methods and they can be classified into two main categories : the *local* ones and the *non-local* ones. In the first category, there are for instance the finite differences or the finite elements. In the second one, there are the pseudo-spectral methods. The finite differences are said to be local because to estimate the derivative in one given point, the values of the function around this point have to be used. Pseudo-spectral methods are said to be non-local, because one has to use all the points of the function to estimate a derivative in one given point.

¹and by *accurate* we mean that the order of the error between the computed solution and the analytical one has to be the same as the machine error ϵ ($\approx 10^{-16}$ or $\approx 10^{-32}$ depending if the computation is done in double or quadruple precision).

Spectral methods involve the decomposition of a given discrete function f onto an orthogonal basis of functions ϕ_n as follow :

$$f(n) = \sum_{i=0}^{N-1} A_i \phi_i(n) \quad (2.1)$$

The derivative of f , $\frac{df(x)}{dx}$ can be obtained computing the derivatives of ϕ_i , and we have :

$$\frac{df(n)}{dn} = \sum_{i=0}^{N-1} A_i \frac{d\phi_i(n)}{dn} \quad (2.2)$$

This show how simple it is to compute the derivative of a given signal when its decomposition is known (because the derivatives of ϕ_i are known !).

In our case, the f function is discrete, and we use the discrete Fourier transform to achieve this decomposition. That is to say, for N discrete points, we have :

$$f(n) = \sum_{k=0}^{N-1} \hat{f}(k) e^{2i\pi kn/N} \quad (2.3)$$

where the $\hat{f}(k)$ are defined as follow:

$$\hat{f}(k) = \sum_{n=0}^{N-1} f(n) e^{-2i\pi kn/N} \quad (2.4)$$

The straight forward decomposition of $f(n)$ is really expensive (about N^2 operations). Fortunately, in 1965, J.W. Cooley and J. Tukey suggested a Fast Fourier Transform (FFT) algorithm that runs in about $N \log N$ operations. This algorithm was the key for real applications using spectral methods.

2.1.2 Why “pseudo” ?

One can notice that in the Fourier space, we have :

$$\frac{d^n \hat{f}(k)}{dx^n} = (ik)^n \hat{f}(k) \quad (2.5)$$

that is to say, calculating derivatives in the Fourier space is no more than calculating a product of two functions. On the other hand, calculating the product of two functions in the physical space is calculating a convolution in Fourier space ; and that is a really expensive operation. The idea is to make the calculations in the Fourier space when only derivatives are involved and to make calculations in the physical space when product of functions are involved. One can quickly switch from one space to the other with the FFT algorithm. That is where the *pseudo* come from ; we take the best from each space.

Remark 2.1 *In this report, when a physical variable is denoted by ϕ , we denote by $\hat{\phi}$ this variable in the Fourier space. The non-linear terms, in the Fourier space, are denoted by*

$\mathcal{F}\{\phi^2\}$, and not by $\widehat{\phi^2}$. For instance, knowing $\widehat{\phi}$, $\mathcal{F}\{\phi^2\}$ is computed as follow :

$$\mathcal{F}\{\phi^2\} = \mathcal{F}\left\{\mathcal{F}^{-1}\{\widehat{\phi}\}^2\right\}$$

where $\mathcal{F}^{-1}\{\}$ denote the inverse of the FFT.

2.1.3 The Korteweg-de Vries equation in the Fourier space

In the Fourier space, the KdV equation may be written as follow :

$$\partial_t \widehat{\eta} + ic_0 k \left(1 - k^2 \frac{d^2}{6}\right) \widehat{\eta} + ik \frac{3}{4} \sqrt{\frac{g}{d}} \mathcal{F}\{\eta^2\} = 0 \quad (2.6)$$

where $\widehat{\eta}$ is the variable η in the Fourier space.

Given this equation, we now introduce different procedures to numerically solve it. Those methods are, of course, usable for other equations.

2.2 RUNGE-KUTTA METHODS

The RK methods are numerical iterative methods used to approximate the solution of Ordinary Differential Equations (ODE). Basically, if the derivative of a physical quantity can be written as a function of this physical quantity, then one can use a RK method to approximate the solution. That is to say, if we have :

$$\begin{cases} y' = f(y, t) \\ y(t_0) = y_0 \end{cases} \quad (2.7)$$

then the approximated solution is given by (Hairer et al. 2004) :

$$y_{n+1} = y_n + \Delta t \sum_{i=1}^s b_i k_i \quad (2.8)$$

where the k_i are defined by :

$$\begin{aligned} k_1 &= f(y_n, t_n) \\ k_2 &= f(y_n + a_{21}k_1\Delta t, t_n + c_2\Delta t) \\ &\vdots \\ k_s &= f\left(y_n + \Delta t \sum_{i=1}^{s-1} a_{si}k_i, t_n + c_s\Delta t\right) \end{aligned}$$

The value for s , c_i , a_{ij} and b_i are given in the literature (Hairer et al. 2004, for instance) and depend on the method's order². Usually, these data are arranged in a mnemonic device, known as a *Butcher tableau* (after John C. Butcher) :

²a method is said to be « of order n » when it is exact for a polynomial of order n .

$$\begin{array}{c|cccc}
0 & & & & \\
c_2 & a_{21} & & & \\
c_3 & a_{31} & a_{32} & & \\
\vdots & \vdots & & \ddots & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} \\
\hline
& b_1 & b_2 & \cdots & b_{s-1} & b_s
\end{array}$$

The most classical example is maybe the RK of order 4 – usually referred as RK4 – and is given by the following *Butcher tableau* :

$$\begin{array}{c|cccc}
0 & & & & \\
1/2 & 1/2 & & & \\
1/2 & 0 & 1/2 & & \\
1 & 0 & 0 & 1 & \\
\hline
& 1/6 & 1/3 & 1/3 & 1/6
\end{array}$$

2.2.1 The adaptive time-step

Constant time-step procedures are shown to be inefficient for nonlinear simulation of surface gravity waves (Clamond 2006). One needs adaptive time-step procedures to capture all the physic of the waves, it enables the procedure to run with the maximal time step while the local relative error is controlled.

The idea is to perform two RK procedures with different orders, p for the first one and $p - 1$ for the second one, and to adapt the time step Δt so that the local relative error between both results is inferior to a given constant. If the error is higher than the given constant, then the *same* step is redone but with a lower Δt , until the error fits the given criteria. The *Butcher tableau* is now represented as follow :

$$\begin{array}{c|cccc}
0 & & & & \\
c_2 & a_{21} & & & \\
c_3 & a_{31} & a_{32} & & \\
\vdots & \vdots & & \ddots & \\
c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} \\
\hline
y_{n+1} & b_1 & b_2 & \cdots & b_{s-1} & b_s \\
\tilde{y}_{n+1} & \tilde{b}_1 & \tilde{b}_2 & \cdots & \tilde{b}_{s-1} & \tilde{b}_s
\end{array}$$

where the $p - 1$ order solution is given by :

$$\tilde{y}_{n+1} = y_n + \Delta t \sum_{i=1}^s \tilde{b}_i k_i \quad (2.9)$$

The Proportional Integral (PI)-step control

Once the local relative error is computed, one has to adapt the time-step Δt . We use the PI-step control (Hairer et al. 2004, p. 164) to achieve this purpose. The corresponding algorithm is given below (Algorithm 1) :

Algorithm 1: PI-step control

Data: $\Delta t_n, tol, \alpha, \beta, y_n$ and err_{n-1}
Result: y_{n+1}, err_n and Δt_{n+1}
 $err_n \leftarrow \infty;$
 $\Delta t_{opt} \leftarrow \Delta t_n;$
while $err_n \geq 1$ **do**
 compute $y_{n+1};$
 compute $\tilde{y}_{n+1};$
 compute $err_n = \|\tilde{y}_{n+1} - y_{n+1}\| / tol;$
 $\Delta t_{opt} \leftarrow \Delta t_n \times err_{n-1}^{-\alpha} \times err_n^\beta$
end
 $\Delta t_{n+1} \leftarrow \Delta t_{opt}$

In the algorithm 1, the quantities α and β are constants defined according to the order of the method. It has been shown (Gustafsson et al. 1988) that the best results are obtained with :

$$\alpha = \frac{0.7}{p} \qquad \beta = \frac{0.4}{p}$$

where p is the order of the method.

One can find a very detailed implementation of this algorithm in Hairer et al. (2004) or in Eichwald (2013).

Bogacki and Shampine

The *Bogacki and Shampine* method (named after B. Bogacki and L. F. Shampine, the authors) is one of the RK schemes we use for the adaptive time-step procedure. The solution y is estimated to the order 3, and the approximated solution \tilde{y} to the order 2. The *Butcher tableau* of this method is the following one :

0					
1/2	1/2				
3/4	0	3/4			
1	2/9	1/3	4/9		
y_{n+1}	2/9	1/3	4/9	0	
\tilde{y}_{n+1}	7/24	1/4	1/3	1/8	

One can notice that we have $k_4^n = k_1^{n+1}$, that is to say the first stage at the $n + 1$ iteration is the same as the last stage at the previous iteration. In other words, for each time step, only two stages are computed³. So, the adaptive time-step procedure comes without extra computational cost ! This property is known as First Same As Last (FSAL).

Other methods of higher order with the FSAL property exist, but our goal is to simulate nonlinear water waves, and it has been shown that

³except for the first time, of course.

when using the MIF, the *Bogacki and Shampine* method is the one giving the best results (Eichwald 2013). Nonetheless, we can name the *Dormand and Prince* method (order 5/4) and the *Verner* one (order 9/8).

2.2.2 Dense Output

As explained in the previous section, constant time-step procedures are inefficient for our purpose. But, using constant time-step procedures enables the user to predict the output times and export the data each x seconds for instance, in order to analyse it or to make an animation or whatever she needs.

Using adaptive time-step procedure makes the user unable to predict the output times. The goal of the dense output is to provide a way to know the result at a given time t between $[t_n; t_{n+1}]$. The idea is to make a Hermit interpolation, using the computed derivatives k_i , to get the expected result. Because only already computed data – the k_i – is involved, the interpolation comes without extra computational cost !

Let's consider we start from t_n and we compute the k_i to obtain y_{n+1} , the value at t_{n+1} . Now, we want the value of $y(t_\theta)$, where $t_\theta = t_n + \theta\Delta t$, for $\theta \in [0; 1]$. This value is obtained using the following formula (Hairer et al. 2004) :

$$y(t_\theta) = y_n + \Delta t \sum_{i=1}^s b_i(\theta) k_i \quad (2.10)$$

where the $b_i(\theta)$ functions depend on the RK method used and are given in the literature. For instance, with the *Bogacki and Shampine* scheme, the $b_i(\theta)$ functions are :

$$\begin{aligned} b_1(\theta) &= \theta - \frac{3\theta^2}{2} + \frac{2\theta^3}{3} \\ b_2(\theta) &= b_3(\theta) = \theta^2 - \frac{2\theta^3}{3} \\ b_4(\theta) &= -\frac{\theta^2}{2} + \frac{2\theta^3}{3} \end{aligned}$$

As explained later, the dense output will also be used to estimate the derivatives of y .

2.2.3 An example with the Korteweg-de Vries equation

As seen before, the KdV equation, in the Fourier space, may be written as (2.6). One can easily see that the time derivative of $\hat{\eta}$ can be written as a function of η , and therefore numerically solved by an adaptive time-step RK procedure as explained hereinbefore :

$$\partial_t \hat{\eta} = -ic_0 k \left(1 - k^2 \frac{d^2}{6} \right) \hat{\eta} - ik \frac{3}{4} \sqrt{\frac{g}{d}} \mathcal{F} \{ \eta^2 \} \quad (2.11)$$

2.3 THE INTEGRATING FACTOR

The IF method consists in a simple – but clever – variable substitution that enables the linear part of the equation to be *exactly* integrated, while the nonlinear one is approximated by a RK method.

Let's take a generic relation :

$$d_t \mathbf{y} + \mathbb{A} \mathbf{y} = \mathcal{N}(\mathbf{y}, t) \quad (2.12)$$

where \mathbf{y} is a vector, \mathbb{A} is the matrix representing the linear part of the equation and $\mathcal{N}(\mathbf{y}, t)$ a function describing the nonlinear part. One can make the following substitution :

$$\mathbf{y} = e^{-\mathbb{A}(t-t_n)} \mathbf{z} \quad (2.13)$$

calculating the derivative of this relation with respect to t , it comes :

$$d_t \mathbf{y} = -\mathbb{A} e^{-\mathbb{A}(t-t_n)} \mathbf{z} + e^{-\mathbb{A}(t-t_n)} d_t \mathbf{z} \quad (2.14)$$

substituting \mathbf{y} back gives :

$$d_t \mathbf{y} + \mathbb{A} \mathbf{y} = e^{-\mathbb{A}(t-t_n)} d_t \mathbf{z} \quad (2.15)$$

leading to :

$$\begin{aligned} d_t \mathbf{z} &= e^{\mathbb{A}(t-t_n)} \mathcal{N}(\mathbf{y}, t) \\ &= e^{\mathbb{A}(t-t_n)} \mathcal{N}(e^{-\mathbb{A}(t-t_n)} \mathbf{z}, t) \end{aligned} \quad (2.16)$$

In (2.16) the time derivative of \mathbf{z} is written as a function of \mathbf{z} and therefore can be solved using a RK method. The value of \mathbf{y} can be found back using the initial substitution.

Remark 2.2 *One can notice that if we had chosen the following substitution $\mathbf{y} = e^{-\mathbb{A}t} \mathbf{z}$, the results would have been exactly the same, algebraically. But not numerically at all ! Numerically, the closest from zero the value is, the better the result is. For instance, one can launch a Python shell and compute different values of $\sin 2k\pi$ for $k \in \llbracket 1; \infty \rrbracket$. When k is small, the given value is correct⁴, but the result is far from being correct when $k \rightarrow \infty$ (for $k = 10^{50}$, the given result is -0.1586 , that is to say...)*

For that reason, the computations must be done close to zero, hence the t_n term.

2.3.1 with the Korteweg-de Vries equation

The KdV equation is directly adaptable to the IF method and we have :

$$\mathbb{A} = ic_0 k \left(1 - k^2 \frac{d^2}{6} \right) \quad \mathcal{N}(\hat{\eta}, t) = -ik \frac{3}{4} \sqrt{\frac{g}{d}} \mathcal{F}\{\eta^2\}$$

⁴ $\sin(2*\pi) = -2.4492935982947064e-16$, which is the machine zero.

2.4 THE MODIFIED INTEGRATING FACTOR

In the last part of this chapter, we talk about the MIF. The MIF is the core of the PhD of [Eichwald \(2013\)](#). As its name suggests, this method is based on the IF we have just talked about earlier. This method aims to reduce the numerical stiffness of the resolution of the equation. It is based on a simple observation : *adding zero never changes, algebraically, the result*. But, as we briefly explained in the remark [2.2](#), numerically, it can yield to a really different result. The results shown by [Eichwald \(2013\)](#) are really impressive. With this method, the time-step is bigger, and thus, the simulation runs faster, for the same error.

As we said, we subtract a given quantity $P(t)$ on both sides of the equation, which returns to add *zero* :

$$d_t \mathbf{y} + \mathbb{A} \mathbf{y} - P(t) = \mathcal{N}(\mathbf{y}, t) - P(t) \quad (2.17)$$

We make a substitution, as for the IF but slightly different :

$$\mathbf{y}(t) = e^{-\mathbb{A}(t-t_n)} \mathbf{z}(t) + \int_{t_n}^t e^{-\mathbb{A}(t-\tau)} P(\tau) d\tau \quad (2.18)$$

calculating the derivative of this relation with respect to t leads to :

$$d_t \mathbf{y}(t) = -\mathbb{A} e^{-\mathbb{A}(t-t_n)} \mathbf{z}(t) + e^{-\mathbb{A}(t-t_n)} d_t \mathbf{z}(t) - \int_{t_n}^t \mathbb{A} e^{-\mathbb{A}(t-\tau)} P(\tau) d\tau + P(t) \quad (2.19)$$

which is equivalent to :

$$d_t \mathbf{z}(t) = e^{\mathbb{A}(t-t_n)} (\mathcal{N}(\mathbf{y}, t) - P(t)) \quad (2.20)$$

One can see that the difference between [\(2.16\)](#) and [\(2.20\)](#) is the presence of the function $P(t)$ in the exponential multiplied term. It is with this key difference that [Eichwald \(2013\)](#) improved the IF.

Now, it is time to choose a convenient function $P(t)$, so that the exponential multiplied term is as closed from zero as possible.

2.4.1 Choosing the best $P(t)$ function

The goal is to choose $P(t)$ such that $\mathcal{N}(\mathbf{y}, t) - P(t)$ is as small as possible. The first idea [Eichwald \(2013\)](#) suggested was choosing $P(t)$ such that at the start of the loop $[t_n; t_{n+1}]$, the difference between $P(t_n)$ and $\mathcal{N}(\mathbf{y}, t_n)$ was zero. Thus we have⁵ :

$$P(t) = d_t \mathbf{y}(t_n) + \mathbb{A} \mathbf{y}(t_n), \text{ for } t \in [t_n; t_{n+1}] \quad (2.21)$$

As one can see, using this definition for $P(t)$ cancels out the right hand side of the equation [\(2.20\)](#) at $t = t_n$, i.e. at the beginning of the loop and therefore decrease the numerical stiffness. But, using a Taylor series of $P(t)$ around $t = t_n$ can perhaps be slightly better :

⁵Here, all the values are known from the previous step. The derivative $d_t \mathbf{y}(t_n)$ is known as k_s , the last stage of the RK step and $\mathbf{y}(t_n)$ as the output value.

$$\begin{aligned}
P_q(t) &= [\mathbf{d}_t \mathbf{y}(t_n) + \mathbb{A} \mathbf{y}(t_n)] + (t - t_n) [\mathbf{d}_t^2 \mathbf{y}(t_n) + \mathbb{A} \mathbf{d}_t \mathbf{y}(t_n)] \\
&\quad + \frac{(t - t_n)^2}{2!} [\mathbf{d}_t^3 \mathbf{y}(t_n) + \mathbb{A} \mathbf{d}_t^2 \mathbf{y}(t_n)] \\
&\quad + \dots + \frac{(t - t_n)^q}{q!} [\mathbf{d}_t^{q+1} \mathbf{y}(t_n) + \mathbb{A} \mathbf{d}_t^q \mathbf{y}(t_n)] \\
&= p_0 + (t - t_n) p_1 + \frac{(t - t_n)^2}{2!} p_2 + \dots + \frac{(t - t_n)^q}{q!} p_q \quad (2.22)
\end{aligned}$$

where $P_q(t)$ is the truncated Taylor series of $P(t)$ around $t = t_n$ at the order q .

Remark 2.3 *MIF₀, MIF₁ and MIF₂ denote the MIF method using the polynomial $P(t)$ truncated at the order 0, 1 or 2. In practice, they are the only ones giving interesting results (Eichwald 2013).*

He explained also that a better approximation could have been chosen calculating the Taylor series around $t = t_n + \frac{\Delta t}{2}$ instead of $t = t_n$, but it would have implied extra computational cost that we prefer to avoid.

2.4.2 Calculating the time derivatives

One can notice that to compute the p_i terms, the time derivatives of $\mathbf{y}(t_n)$ are needed. The RK method directly gives $\mathbf{y}(t_n)$ and $\mathbf{d}_t \mathbf{y}(t_n)$. Others are obtained using the dense output (see 2.2.2 p. 14). Indeed, the derivation of (2.10) leads to :

$$\frac{d^n \mathbf{y}(t)}{dt^n} = \left(\frac{1}{\Delta t} \right)^{n-1} \sum_{i=1}^s \frac{d^n b_i(\theta)}{d\theta^n} k_i \quad \text{for } n \geq 1 \quad (2.23)$$

the evaluation in $t = t_n$ gives the expected value.

2.4.3 Substituting y back

The last needed thing to do is to calculate the integral at the right hand side of the equation (2.18). Because $P(t)$ is a polynomial, the integral can be analytically computed :

$$\begin{aligned}
\int_{t_n}^t e^{-\mathbb{A}(t-\tau)} P(\tau) d\tau &= \left(1 - e^{-\mathbb{A}(t-t_n)} \right) p_0 \mathbb{A}^{-1} \\
&\quad + \left(e^{-\mathbb{A}(t-t_n)} - 1 + \mathbb{A}(t - t_n) \right) p_1 \mathbb{A}^{-2} \\
&\quad \dots
\end{aligned}$$

therefore, \mathbf{y} is equal to :

$$\mathbf{y}(t) = e^{-\mathbb{A}(t-t_n)} \mathbf{z}(t) + \sum_{n=0}^q \left(-e^{-\mathbb{A}(t-t_n)} + \sum_{k=0}^n \frac{\mathbb{A}^k (t_n - t)^k}{k!} \right) (-1)^n \mathbb{A}^{-(n+1)} p_n \quad (2.24)$$

Using this relation, one can easily switch from \mathbf{y} to \mathbf{z} and thus one can use the MIF proposed by Eichwald (2013).

2.5 SOME RESULTS

This section shows some results of the improvements brought by the MIF and IF methods. As in this whole chapter, the KdV equation is used as example. The periodic-cnoidal solution is chosen as initial condition.

With a wave height $H = 1$ and a cnoidal wave parameter $m = 0.9999$ and a tolerance of 10^{-12} and 256 points, we obtain the following averaged time-steps⁶ :

	classic	IF	MIF
dt	$1.5e^{-5}$	$2.08e^{-5}$	$2.21e^{-5}$
improvement		+38.25%	+6.34%

Table 2.1 – Averaged time-steps and percentage improvements for KdV equation with the following parameters : $N = 256$, $H = 1$, $m = 0.9999$, $tol = 10^{-12}$

In the case of horizontal bottom, studying the averaged time-step make sense, because after a short time the adaptive time-step procedure is auto-calibrated on the best value according to the given tolerance⁷. The figure 2.1 shows that property.

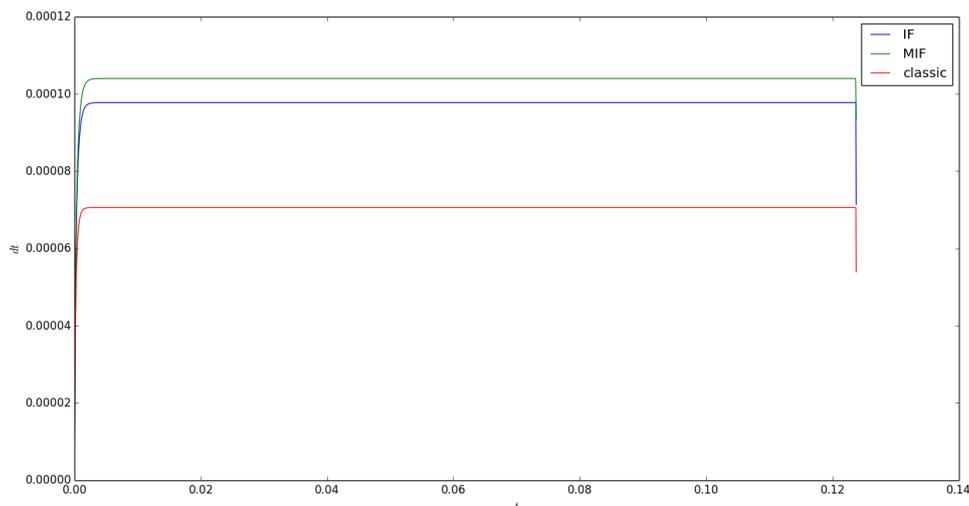


Figure 2.1 – time-steps for the classic, IF and MIF methods

Different cnoidal-waves simulations have been performed with different parameters H and m . The table 2.2 p. 19 shows the improvement percentages we obtained.

When the wave height is small, the IF is really efficient while the MIF is inefficient (a negative percentage is found). As the wave height H and the cnoidal-wave parameter m get higher the efficiency of the IF decreases and the MIF gives better results ! From that observation, we can say the more the non-linear term $\mathcal{N}(\mathbf{y}, t)$ is *important*, the more the results are interesting. In his PhD theses, Brice Eichwald performed the MIF method over different equations, such as the KdV equation, the Non-linear Schrödinger (NLS) equation or the High-Order Spectral (HOS)

⁶Improvement percentages for the IF are computed compared to the classic method, and the percentages for MIF are computed compared to the IF.

⁷If the bottom is moving then the time-steps change to assert that the local error is always lower than the tolerance.

		$H = 0.1$	$H = 0.4$	$H = 0.7$	$H = 1.0$
$m = 0.1$	IF	94.83%	X	X	X
	MIF	-45.15%	X	X	X
$m = 0.2$	IF	146.26%	36.86%	X	X
	MIF	-52.83%	-2.80%	X	X
$m = 0.4$	IF	233.55%	64.21%	35.42%	23.63%
	MIF	-65.34%	-1.75%	3.26%	1.53%
$m = 0.8$	IF	321.00%	84.67%	47.51%	32.40%
	MIF	-73.72%	-16.50%	6.44%	4.11%
$m = 0.99$	IF	365.45%	95.65%	53.64%	36.68%
	MIF	-76.27%	-21.92%	8.41%	5.67%
$m = 0.9999$	IF	378.80%	99.13%	55.78%	38.25%
	MIF	-76.88%	-23.09%	9.52%	6.34%
$m = 0.999999$	IF	386.37%	101.75%	57.45%	39.47%
	MIF	-77.23%	-24.41%	9.82%	6.82%

Table 2.2 – Improvement percentages for different cnoidal-waves simulation with different parameters H and m .

equation. He shows that the more the equation is *non-linear* the more the MIF method is interesting. In particular, for the HOS equation, he gets a impressive improvement varying from 3 440% to 62 513 369.00%⁸, while the improvement of the IF method was only 50%.

Remark 2.4 *The percentages that I have found are quite different from the ones found by Eichwald (2013). I have been unable to understand why. But I think the most important is that the results' behavior is the same.*

⁸The percentage changes when the equation parameters change.

DERIVATION OF NONLINEAR WATER-WAVE EQUATIONS WITH A VARIABLE BOTTOM

In the chapter 1, classical nonlinear water-wave equations have been given. Now, we suggest to show where do those equations come from. As our goal is to simulate tsunamis, we will assume that the seabed is not horizontal but can change in space *and* in time.

We use exactly the same domain as shown in 1.1 p. 3. We denote by $\dot{\phi}$ or $\frac{D}{Dt}(\phi)$, equally, the total derivative of the physical quantity ϕ . That is to say, we have :

$$\frac{D}{Dt}(\phi) = \dot{\phi} = \partial_t \phi + (\mathbf{u} \cdot \nabla) \phi + v \partial_y \phi \quad (3.1)$$

We can notice that for ϕ_s and ϕ_b we have :

$$\dot{\phi}_s = \partial_t \phi_s + (\mathbf{u} \cdot \nabla) \phi_s \quad \dot{\phi}_b = \partial_t \phi_b + (\mathbf{u} \cdot \nabla) \phi_b \quad (3.2)$$

It has been observed that the horizontal velocity in the fluid is usually uniform across the fluid depth (Mitsotakis 2013). That why, we vertically integrate the Euler equations, which describe the flow, to simplify the problem¹. Using the depth-averaging procedure, we obtain a new set of equations. Making assumptions on the vertical velocity will enable us to obtain the SW and SGN equations.

3.1 EULER EQUATIONS

The Euler equations are a set of equations governing inviscid flow. The equations represent continuity and conservation of momentum and energy. When assuming the fluid mass density ρ is constant, the continuity equation is given by (3.3) and the conservation of momentum is given by (3.4a) and (3.4b). They correspond to the Navier-Stokes equation with zero viscosity and heat conduction terms.

$$\nabla \cdot \mathbf{u} + \partial_y v = 0 \quad (3.3)$$

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + (v \partial_y) \mathbf{u} = -\nabla \frac{P}{\rho} \quad (3.4a)$$

¹From 3 parameters (e.g. $\mathbf{u}(x, y, t)$), depth-integrating simplify the problem to 2 parameters ($\bar{\mathbf{u}}(x, t)$).

$$\partial_t v + (\mathbf{u} \cdot \nabla)v + (v\partial_y)v = -\partial_y \frac{P}{\rho} - g \quad (3.4b)$$

where P is the pressure, and g the acceleration due to the gravity. This set of equations is supplemented by two kinematic boundary conditions (at the free surface and at the seabed) :

$$v_s = \partial_t \eta + \mathbf{u}_s \cdot \nabla \eta \quad \text{at } y = \eta(x, t) \quad (3.5)$$

$$v_b = -\partial_t d - \mathbf{u}_b \cdot \nabla d \quad \text{at } y = -d(x, t) \quad (3.6)$$

Those conditions mean that the free surface and the seabed are impermeable. That is to say that material particles on the surface (or the seabed) remain on the surface (or the seabed).

3.2 THE DEPTH-AVERAGED EULER EQUATIONS

The idea is to vertically integrate the Euler equations (3.3), (3.4a) and (3.4b) and to combine the obtained equations to write h , \mathbf{u} and P as a function of v . Then, using different ansatzs on v (based on the physic of the waves), we find the SW and SGN equations.

3.2.1 Integrating the continuity equation

Integrating (3.3) from $y = -d$ to $y = \eta$, we have :

$$\int_{-d}^{\eta} \nabla \cdot \mathbf{u} dy + \int_{-d}^{\eta} \partial_y v dy = 0 \quad (3.7)$$

Using Leibniz rule² yields to :

$$\nabla \cdot \int_{-d}^{\eta} \mathbf{u} dy - \mathbf{u}_s \cdot \nabla \eta - \mathbf{u}_b \cdot \nabla d + v_s - v_b = 0 \quad (3.8)$$

Finally, using conditions (3.5) and (3.6), we can simplify the last equation to :

$$\partial_t h + \nabla \cdot (h\bar{\mathbf{u}}) = 0 \quad (3.9)$$

This equation relates the fluid height h and the depth-averaged horizontal velocity $\bar{\mathbf{u}}$. This equation is common to the SW and SGN equations and one can notice that no approximation has been made, this equation is exact.

3.2.2 Integrating the horizontal momentum equation

In a similar way, the horizontal momentum equation (3.4a) is depth-averaged. But, to do so, it is simpler to rewrite the left-hand side, using the continuity (3.3) and after some algebra³, as follow :

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + v\partial_y \mathbf{u} = \partial_t \mathbf{u} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) + \partial_y (v\mathbf{u}) \quad (3.10)$$

² $d_\theta \left(\int_{a(\theta)}^{b(\theta)} f(x, \theta) dx \right) = \int_{a(\theta)}^{b(\theta)} f_\theta(x, \theta) dx + f(b(\theta), \theta) b_\theta(\theta) - f(a(\theta), \theta) a_\theta(\theta)$

³in particular, using the fact that $\nabla \cdot (\mathbf{u} \otimes \mathbf{v}) = \mathbf{u} \cdot (\nabla \cdot \mathbf{v}) + (\mathbf{v} \cdot \nabla) \mathbf{u}$, for $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^2$ as shown in A.1 p. 33.

where “ \otimes ” denotes the tensor product.

Integrating (3.4a), with the left part written as in (3.10), from $-d$ to η , using the Leibniz rule and the kinematic boundary conditions, after simplification, leads to :

$$\partial_t (h\bar{\mathbf{u}}) + \nabla \cdot (h\overline{\mathbf{u} \otimes \mathbf{u}}) = - \int_{-d}^{\eta} \nabla \frac{P}{\rho} dy \quad (3.11)$$

Now, we extract $\frac{\bar{P}}{\rho}$ from the right-hand side of this equation using Leibniz rule.

$$\int_{-d}^{\eta} \nabla \frac{P}{\rho} dy = \nabla \left(h \frac{\bar{P}}{\rho} \right) - \frac{P_s}{\rho} \nabla \eta - \frac{P_b}{\rho} \nabla d \quad (3.12)$$

Gathering the two last equations, we find :

$$\partial_t (h\bar{\mathbf{u}}) + \nabla \cdot (h\overline{\mathbf{u} \otimes \mathbf{u}}) = -\nabla \left(h \frac{\bar{P}}{\rho} \right) + \frac{P_s}{\rho} \nabla \eta + \frac{P_b}{\rho} \nabla d \quad (3.13)$$

P_s is known *a priori*, \bar{P} and P_b are to be defined using the depth-averaged vertical momentum.

3.2.3 Integrating the vertical momentum equation

The last step in the process of re-writing the Euler equations, is to use the depth-averaged vertical momentum equation to write \bar{P} and P_b as a function of v . For that purpose, we denote by \dot{v} and ζ the following quantities :

$$\begin{aligned} \dot{v} &= \partial_t v + (\mathbf{u} \cdot \nabla) v + (v \partial_y) v = -\partial_y \frac{P}{\rho} - g \\ \zeta &= \int_{-d}^{\eta} (y + d) \dot{v} dy \end{aligned}$$

The quantity ζ can be computed quite straight forward and leads to :

$$\zeta = h \frac{\bar{P}}{\rho} - h \frac{P_s}{\rho} - \frac{g}{2} h^2 \quad (3.14)$$

Therefore, the depth-averaged pressure, needed in (3.13), is given by :

$$h \frac{\bar{P}}{\rho} = \zeta + h \frac{P_s}{\rho} + \frac{g}{2} h^2 \quad (3.15)$$

The last needed quantity is the pressure at the bottom, which can be obtained by integrating the vertical momentum from y to η , and taking the value for $y = -d$. It yields :

$$P_b = \rho h (\bar{v} + g) + P_s \quad (3.16)$$

One can notice that the pressure at the bottom is the sum of the hydrostatic pressure, $P_{hy} = \rho h g + P_s$, and a hydrodynamic one, $P_{dy} = \rho h \bar{v}$.

3.2.4 Final depth-averaged equations

Finally, gathering (3.9), (3.13), (3.15) and (3.16) we obtain the following system :

$$\begin{cases} \partial_t h + \nabla \cdot (h\bar{\mathbf{u}}) = 0 \\ \partial_t (h\bar{\mathbf{u}}) + \nabla \cdot (h\bar{\mathbf{u}} \otimes \bar{\mathbf{u}}) = -\nabla \left(h \frac{\bar{P}}{\rho} \right) + \frac{P_s}{\rho} \nabla \eta + \frac{P_b}{\rho} \nabla d \end{cases} \quad (3.17)$$

with

$$\begin{aligned} \dot{v} &= \partial_t v + (\mathbf{u} \cdot \nabla)v + (v\partial_y)v \\ \bar{\zeta} &= \int_{-d}^{\eta} (y+d)\dot{v}dy \\ h \frac{\bar{P}}{\rho} &= \bar{\zeta} + h \frac{P_s}{\rho} + \frac{g}{2}h^2 \\ P_b &= \rho h (\bar{v} + g) + P_s \end{aligned}$$

3.3 MODELS

The Euler equations rewritten, we can make different ansatzs on the vertical velocity field v . Those different ansatzs will lead to the SW or the SGN equation.

We start with the SW equations, because the calculations are really straight forward from (3.17). Once we are done with the SW equations, we will talk about the SGN equations.

3.3.1 Saint Venant equations (or Shallow-Water equations)

One can assume that the wavelength is much higher than the depth, hence the velocity is mainly horizontal and the vertical component v can be neglected. This yields to $\mathbf{u} \approx \bar{\mathbf{u}} \approx \mathbf{u}(x, t)$ and $v \approx 0$. This simplifies radically our equations, because those assumptions give immediately $\dot{v} \approx 0$ and $\bar{\zeta} \approx 0$. Moreover, because \mathbf{u} is assumed to be constant with respect to y , we have $\bar{\mathbf{u}} \otimes \bar{\mathbf{u}} = \bar{\mathbf{u}} \otimes \bar{\mathbf{u}}$.

Using those observations we get :

$$P_b = \rho h g + P_s \quad h \frac{\bar{P}}{\rho} = h \frac{P_s}{\rho} + g \frac{h^2}{2} \quad (3.18)$$

that is to say, the pressure field is taken as hydrostatic everywhere in the fluid. Therefore, the system (3.17) becomes, after some simple algebra :

$$\begin{cases} \partial_t h + \nabla \cdot (h\bar{\mathbf{u}}) = 0 \\ \partial_t (h\bar{\mathbf{u}}) + \nabla \cdot (h\bar{\mathbf{u}} \otimes \bar{\mathbf{u}} + \frac{1}{2}gh^2\mathbb{I}) = hg\nabla d - h\nabla \frac{P_s}{\rho} \end{cases} \quad (3.19)$$

where \mathbb{I} is the identity matrix.

Those equations are known as the Saint-Venant equations or the Shallow-Water (SW) equations. They describe the motion of an ideal incompressible fluid of a constant density ρ , with a variable bottom.

Conservative form

The conservative form of the SW equations is often preferred (for computational reasons) to the non-conservative form. In this case, it is quite simple to obtain it.

Using $\partial_t h + \nabla \cdot (h\bar{\mathbf{u}}) = 0$ and after some algebra, the last equation of (3.19) can be written as :

$$\partial_t \bar{\mathbf{u}} + \nabla \cdot \left(\frac{1}{2} \bar{\mathbf{u}}^2 + g\eta - \frac{P_s}{\rho} \right) = -(\nabla \wedge \bar{\mathbf{u}}) \wedge \bar{\mathbf{u}} \quad (3.20)$$

One can assume the flow to be irrotational, and it leads to the following system, which is conservative :

$$\begin{cases} \partial_t h + \nabla \cdot (h\bar{\mathbf{u}}) = 0 \\ \partial_t \bar{\mathbf{u}} + \nabla \cdot \left(\frac{1}{2} \bar{\mathbf{u}}^2 + g\eta - \frac{P_s}{\rho} \right) = 0 \end{cases} \quad (3.21)$$

3.3.2 Serre Green-Nagdhi equations

Previously, we assumed that vertical velocity was negligible (3.3.1). We saw that it was equivalent to assume the pressure to be hydrostatic everywhere in the fluid. Now, the vertical velocity is taken as linear with respect to y , this will bring up the hydrodynamic part of the pressure. This ansatz also means that the incompressibility is imposed, i.e. $\nabla \cdot \mathbf{u} = 0$. Our ansatz is that v is defined as follow :

$$v = v_b + \frac{y+d}{\eta+d} (v_s - v_b) \quad (3.22)$$

The vertical velocity is equal to v_b at the seabed and grows linearly to v_s when $y = \eta$. One can observe that we have $v_s - v_b = \partial_t h + \mathbf{u} \cdot \nabla h$ and therefore using (3.9), v can be rewritten as follow :

$$v = v_b - (y+d) \nabla \cdot \mathbf{u} \quad (3.23)$$

Because \mathbf{u} is assumed to be uniform across the fluid depth, we have $\mathbf{u}_b \approx \mathbf{u} \approx \bar{\mathbf{u}}$. Therefore, v_s can be computed because the bottom is exactly known⁴. As a consequence, v is not an unknown anymore. In other words, we can use it to compute \bar{v} and $\bar{\xi}$.

Depth-averaged vertical acceleration

As shown in A.2 page 33, the relation below holds for any physical quantity ϕ of the flow (assumed ideal and incompressible) :

$$\int_{-d}^{\eta} \frac{D}{Dt} (\phi) dy = \partial_t \int_{-d}^{\eta} \phi dy + \nabla \cdot \int_{-d}^{\eta} \mathbf{u} \phi dy \quad (3.24)$$

Using this relation, we can compute the depth-averaged vertical acceleration $\bar{\dot{v}}$, as we have :

$$\dot{v} = \frac{D}{Dt} (v) = \partial_t v + (\mathbf{u} \cdot \nabla) v + v \partial_y v$$

⁴i.e. we know $d(x, t), \forall x, \forall t$.

After some cumbersome algebra, we obtain :

$$\bar{v} = v_b + \frac{h}{2} \left[(\nabla \cdot \mathbf{u})^2 - \nabla \cdot \partial_t \mathbf{u} - (\mathbf{u} \cdot \nabla) (\nabla \cdot \mathbf{u}) \right] \quad (3.25)$$

One can show that, if the bottom was horizontal, the free surface acceleration would be equal to :

$$\frac{D}{Dt} (-h \nabla \cdot \mathbf{u}) = h \left[(\nabla \cdot \mathbf{u})^2 - \nabla \cdot \partial_t \mathbf{u} - (\mathbf{u} \cdot \nabla) (\nabla \cdot \mathbf{u}) \right] \quad (3.26)$$

We denote by $\tilde{\Gamma}$ this quantity. The bottom acceleration is given by v_b , and we designate by Γ_b this quantity. Because when the bottom is variable we have $v_s = v_b - h \nabla \cdot \mathbf{u}$, therefore the acceleration of the free surface, Γ_s , is given by :

$$\Gamma_s = \Gamma_b + \tilde{\Gamma} \quad (3.27)$$

Using those notations, we can write the depth-averaged vertical acceleration as follow :

$$\bar{v} = \Gamma_b + \frac{1}{2} \tilde{\Gamma} \quad (3.28)$$

The hydrodynamic pressure

The last step is to compute the quantity ζ , which is related to the hydrodynamic pressure⁵.

$$\begin{aligned} \zeta &= \int_{-d}^{\eta} (y+d) \frac{D}{Dt} (v) dy \\ &= \int_{-d}^{\eta} \frac{D}{Dt} ((y+d)v) dy - \int_{-d}^{\eta} v(v-v_b) dy \\ &= \partial_t \int_{-d}^{\eta} (y+d)v dy + \nabla \cdot \mathbf{u} \int_{-d}^{\eta} (y+d)v dy + (\nabla \cdot \mathbf{u}) \int_{-d}^{\eta} (y+d)v dy \end{aligned}$$

We have :

$$\int_{-d}^{\eta} (y+d)v dy = \frac{h^2}{2} v_b - \frac{h^3}{3} \nabla \cdot \mathbf{u}$$

Injecting this result in the expression of ζ yields to the following result, after some algebra and simplifications:

$$\zeta = \frac{h^2}{2} \Gamma_b + \frac{h^2}{3} \tilde{\Gamma} \quad (3.29)$$

Final Serre-Green Nagdhi equations

Ultimately, we have the following equation :

$$\partial_t (h\mathbf{u}) + \nabla \cdot (h\mathbf{u} \otimes \mathbf{u}) + \nabla \cdot \left(\frac{gh^2}{2} + \frac{h^2\Gamma_b}{2} + \frac{h^2\tilde{\Gamma}}{3} \right) = h \left(g + \Gamma_b + \frac{\tilde{\Gamma}}{2} \right) \nabla d - h \nabla \frac{P_s}{\rho} \quad (3.30)$$

with :

$$\tilde{\Gamma} = h \left[(\nabla \cdot \mathbf{u})^2 - \nabla \cdot \partial_t \mathbf{u} - (\mathbf{u} \cdot \nabla) (\nabla \cdot \mathbf{u}) \right] \quad (3.31a)$$

⁵The link between ζ and the hydrodynamic pressure is that we have : $P_{dy} = \frac{\rho}{h} \zeta$

$$\Gamma_b = - (\partial_{tt}d + 2\mathbf{u} \cdot \nabla \partial_t d + \mathbf{u}^2 \Delta d + \mathbf{u} \cdot (\nabla \cdot \mathbf{u}) \nabla d + (\partial_t \mathbf{u}) \cdot \nabla d) \quad (3.31b)$$

The momentum equation (3.30) and the continuity equation (3.9) are known as the Serre Green-Nagdhi (SGN) equations.

Remark 3.1 *One can notice that if the bottom is assumed to be variable in space only (not in time), only Γ_b is affected, where the time derivatives of d vanish. And the equations given by [Fernando J. Seabra-Santos et al. \(1987\)](#) are recovered.*

When the bottom is assumed to be horizontal, we have $\Gamma_b = 0$ and the left part of the equation (3.30) vanishes. The equation (3.30) given page 26 is then recovered.

Remark 3.2 *An other remark that can be done, is that when the accelerations of the bottom Γ_b and of the free surface $\tilde{\Gamma}$ – if the bottom would have been horizontal – are taken as zero, then the SW equations derived before are recovered.*

A more classical form

In this subsection, we rewrite the equation (3.30) into a more classical form, i.e. a time derivative plus a gradient (plus something if it is not conservative).

Remark 3.3 *This classical form is given in two dimensions only. One can do it in three dimensions, but the algebra is quite cumbersome... We denote by ϕ_α , the derivative of ϕ according to α . Using this notation, the equation (3.30) becomes :*

$$(hu)_t + \left(hu^2 + \frac{h^2 g}{2} + \frac{h^2 \Gamma_b}{2} + \frac{h^2 \tilde{\Gamma}}{3} \right)_x = h \left(g + \Gamma_b + \frac{\tilde{\Gamma}}{2} \right) d_x \quad (3.32)$$

Starting from the equation (3.32), we gather all the time derivatives of the velocity u . Doing this, time derivatives of h and d appear. The time derivative of the elevation h_t can be eliminated using the continuity equation (3.9). The time derivative of the bottom d_t is supposed to be known.

Finally, after some simplifications⁶, we obtain the following equation :

$$\begin{aligned} q_t + \left(hu^2 - \frac{2h^3 u_x^2}{3} - \frac{h^3 uu_{xx}}{3} - h^2 h_x uu_x - \frac{h^2}{2} (d_{tt} + u^2 d_{xx} + uu_x d_x) \right)_x = \\ -u \left[\left(\frac{3}{2} h^2 d_{tx} - h (hu)_x d_x \right)_x + h_x u d_x^2 + 2h u_x d_x^2 + 4h d_x d_{tx} \right] \\ - h d_x (d_{tt} + u^2 d_{xx}) + \frac{h^2}{2} (u_x^2 - uu_{xx}) d_x - h g \eta_x \end{aligned} \quad (3.33)$$

with

$$q = hu - \left(\frac{h^3 u_x}{3} \right)_x + h u d_x^2 + u \left(\frac{h^2}{2} d_x \right)_x \quad (3.34)$$

Remark 3.4 *We can notice that, in the case of a flat bottom, we recover the classical SGN equations :*

⁶May be it's more simplifiable...

$$\begin{aligned}
q_t + \left(hu^2 + \frac{gh^2}{2} - \frac{2h^3u_x^2}{3} - \frac{h^3uu_{xx}}{3} - h^2h_xuu_x \right)_x &= 0 \\
q - hu + \left(\frac{h^3u_x}{3} \right)_x &= 0
\end{aligned}$$

Solving the algebraic equation with a pseudo-spectral method The first two equations (3.9) and (3.32) are evolution type, while the third one (3.34) relates the conserved variable q to the primitive variables. In this paragraph, we explain how to recover the averaged velocity u from q when the equations are solved using a pseudo-spectral method.

We call d^* the “averaged” value of the depth water. This value is supposed to be the same for all x (but not necessarily for all t). It does not matter if it is not the real averaged value, but it is supposed to be a *known* constant value, in space, around d^* . We write h as this constant value d^* plus a fluctuating value \tilde{h} , so that $h = d^* + \tilde{h}$. So we have :

$$q - d^*u + \frac{d^{*3}}{3}u_{xx} = \underbrace{\tilde{h}u + \frac{d^{*3}}{3}u_{xx} - \left(\frac{h^3u_x}{3} \right)_x + hud_x^2 + u \left(\frac{h^2}{2}d_x \right)_x}_{\mathcal{N}(h,u,d)}$$

Then, we apply a fixed point method, in the Fourier space,

$$\hat{u}_{j+1} = \frac{\hat{q} - \mathcal{F} \{ \mathcal{N}(h, u_j, d) \}}{d^* + \frac{k^2}{3}d^{*3}} \quad (3.35)$$

where $\hat{\phi} = \mathcal{F} \{ \phi \}$. Using this method, u can be recovered, at the desired precision ε .

3.4 AN EXAMPLE OF A LINEARIZATION

In this very last section, we show a way to linearize the SW equations when the bottom is variable so the IF and the MIF methods can be used. (The procedure is more or less the same for the SGN equations... but a little bit longer)

Let’s start from the equation (3.21), re-written here below, in two dimensions only.

$$\begin{cases} \partial_t h + (h\bar{u})_x = 0 \\ \partial_t \bar{u} + \left(\frac{1}{2}\bar{u}^2 + g\eta - \frac{P_s}{\rho} \right)_x = 0 \end{cases} \quad (3.36)$$

Our goal is to rewrite this set of equations into the generic form shown in the previous chapter, $d_t \mathbf{y} + \mathbb{A} \mathbf{y} = \mathcal{N}(\mathbf{y}, t)$, where \mathbb{A} is not zero, obviously...

For that purpose, we write h as the sum of two quantities :

$$h = \tilde{h} + d_0 \quad (3.37)$$

⁷By testing different values, empirically I found that the best value was $d^* = \min(d(t))$

where d_0 is a given value around d . This value is *constant* in space *and* in time *and* horizontal. It is assumed to be known. The next step is to write the equation using this decomposition, which gives :

$$\begin{cases} \partial_t \tilde{h} + d_0 \bar{u}_x = -(\tilde{h}\bar{u})_x \\ \partial_t \bar{u} + g \tilde{h}_x = -\left(\frac{1}{2} \bar{u}^2 + g(d_0 - d) - \frac{P_s}{\rho}\right)_x \end{cases} \quad (3.38)$$

In the Fourier space this set of equations becomes simply :

$$\begin{cases} \partial_t \hat{h} + ikd_0 \hat{u} = -ik\mathcal{F}\{\tilde{h}\bar{u}\} \\ \partial_t \hat{u} + ikg\hat{h} = -ik\mathcal{F}\left\{\frac{1}{2}\bar{u}^2 + g(d_0 - d) - \frac{P_s}{\rho}\right\} \end{cases} \quad (3.39)$$

The last step is about nondimensionalization. Therefore, we introduce \mathbf{y} the vector of dimensionless variables in the Fourier space :

$$\hat{\mathbf{y}} \equiv \begin{pmatrix} ik\hat{h} \\ \frac{i\omega}{g}\hat{u} \end{pmatrix} \quad (3.40)$$

where $\omega^2 = gk^2d_0$. The SW equation are now :

$$\hat{\mathbf{y}}_t + \mathbb{A}\hat{\mathbf{y}} = \mathcal{N}(\hat{\mathbf{y}}, t), \quad \mathbb{A} = \begin{pmatrix} 0 & i\omega \\ i\omega & 0 \end{pmatrix} \quad (3.41)$$

and $\mathcal{N}(\hat{\mathbf{y}}, t)$ contains all the remaining non-linear terms. That is to say :

$$\mathcal{N}(\hat{\mathbf{y}}, t) = \begin{pmatrix} k^2\mathcal{F}\{\tilde{h}\bar{u}\} \\ \frac{\omega k}{g}\mathcal{F}\left\{\frac{1}{2}\bar{u}^2 + g(d_0 - d) - \frac{P_s}{\rho}\right\} \end{pmatrix} \quad (3.42)$$

And that it's. We now have the SW equations written in the classical form that we used to introduce the Integrating Factor and the Modified Integrating Factor methods. Moreover, given the form the matrix \mathbb{A} , one can compute analytically the exponentials, and we have :

$$e^{\mathbb{A}t} = \begin{pmatrix} \cos(\omega t) & i \sin(\omega t) \\ i \sin(\omega t) & \cos(\omega t) \end{pmatrix} \quad (3.43)$$

for that reason, computing $e^{\mathbb{A}t}\mathbf{y}$ has a very small cost.

Remark 3.5 *If the exponential can not be analytically computed, it is possible to compute the action of the matrix exponential without computing the actual matrix exponential. One can find more information in this paper : <http://eprints.ma.man.ac.uk/1591/> called Computing the Action of the Matrix Exponential, with an Application to Exponential Integrators by Awad H. Al-Mohy and Nicholas J. Higham, published in 2011.*

CONCLUSION

EN FRANÇAIS

Ayant suivi le *parcours recherche* proposé à l'UTC, j'ai pu avoir la chance de faire mon stage de fin d'études dans un laboratoire de recherches universitaire. Cela m'a permis de mettre un pied dans le monde de la recherche, d'étudier des méthodes de simulations à la pointe⁸, mais aussi d'avoir une certaine autonomie et liberté.

Si je devais retenir un enseignement de ce stage de recherche, ce serait qu'il faut commencer par s'approprier les méthodes existantes puis chercher à les remettre en cause afin de les améliorer. C'est ainsi que j'ai commencé à travailler avec des méthodes existantes (comme le Facteur Intégrant et le Facteur Intégrant Modifié) et des équations simples. Puis une fois que je me suis correctement approprié les équations, j'ai pu travailler à leur généralisation. Les équations de Saint-Venant à fond variable en temps et en espace sont connues depuis longtemps. En revanche, à ma connaissance, ce n'est pas le cas pour les équations de Serre, qui ne sont connues qu'à fond variable en espace, depuis la thèse de Fernando Seabra Santos en 1985. Généraliser les équations de Serre à un fond variable en temps et en espace me semblait donc important afin de mieux rendre compte de la réalité, notamment lors de tsunamis où le fond sous-marin varie dans le temps. Ce modèle, présenté dans ce rapport, devrait d'ailleurs être publié dans un article scientifique prochainement.

J'espère ainsi qu'avec ce stage de fin d'études, une petite pierre a pu être ajoutée à l'édifice que constitue la recherche en mécanique des fluides.

Remark 3.6 *Le lecteur intéressé trouvera le code Fortran 95 que j'ai écrit lors de ce stage à cette adresse <https://chabotsi.fr/to/sTI8Fg>. Ce code permet de simuler les équations de Saint-Venant et de Serre à fond variable en temps et en espace avec les différentes méthodes que nous avons exposées.*

IN ENGLISH

Because I have been following the *research course* at the UTC, I had the chance to make my final internship in an university research laboratory. This internship enabled me to put one foot in the world of research, to study the state-of-the-art simulation methods, but also to enjoy autonomy and freedom.

If there is a lesson of this experience to take home is that one must take ownership of the existing methods and then try to challenge them in order

⁸la thèse de Brice Eichwald, sur le Facteur Intégrant Modifié, date de 2013 seulement!

to make them better. That's how I started to work with existing methods (such as the Integrating Factor and the Modified Integrating Factor) and simple wave equations. Once I was comfortable with those equations, I have been able to work on their generalization. The SW equations with a variable bottom, in space and time, have been known for a long time. On the other hand, to my knowledge, the Serre equations are known for a variable bottom in space only since 1985, with the PhD thesis of Fernando Seabra Santos. Generalizing the Serre equations to a variable bottom in space and time looked important to me in order to reflect the reality a bit better. This model, introduced in this report, should be published soon in a scientific article.

I hope that with this internship, a small contribution to the research community in fluid mechanics has been done.

Remark 3.7 *Interested readers can find the Fortran 95 code that I wrote in this internship at this address <https://chabotsi.fr/to/sTI8Fg>. This code is used to simulate the SW and SGN equations with a variable bottom in time and space with the different methods we have outlined.*

THE PROOF OF SOME IDENTITIES

A

In this appendix, we prove some identities that have been used in this report and that may be useful to interested readers.

A.1 THE DIVERGENCE OF A TENSOR PRODUCT

Let $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^2 \times \mathbb{R}^2$, we show that :

$$\nabla \cdot (\mathbf{u} \otimes \mathbf{v}) = \mathbf{u} (\nabla \cdot \mathbf{v}) + (\mathbf{v} \cdot \nabla) \mathbf{u} \quad (\text{A.1})$$

Proof.

$$\begin{aligned} \mathbf{u} (\nabla \cdot \mathbf{v}) + (\mathbf{v} \cdot \nabla) \mathbf{u} &= \begin{pmatrix} u_1 \partial_{x_1} v_1 + u_1 \partial_{x_2} v_2 \\ u_2 \partial_{x_1} v_1 + u_2 \partial_{x_2} v_2 \end{pmatrix} + \begin{pmatrix} v_1 \partial_{x_1} u_1 + v_2 \partial_{x_2} u_1 \\ v_1 \partial_{x_1} u_2 + v_2 \partial_{x_2} u_2 \end{pmatrix} \\ &= \begin{pmatrix} u_1 \partial_{x_1} v_1 + v_1 \partial_{x_1} u_1 + u_1 \partial_{x_2} v_2 + v_2 \partial_{x_2} u_1 \\ u_2 \partial_{x_1} v_1 + v_1 \partial_{x_1} u_2 + u_2 \partial_{x_2} v_2 + v_2 \partial_{x_2} u_2 \end{pmatrix} \\ &= \begin{pmatrix} \partial_{x_1} (u_1 v_1) + \partial_{x_2} (u_1 v_2) \\ \partial_{x_1} (u_2 v_1) + \partial_{x_2} (u_2 v_2) \end{pmatrix} \\ &= \nabla \cdot (\mathbf{u} \otimes \mathbf{v}) \end{aligned}$$

□

A.2 A LEIBNIZ RULE-LIKE FOR A TOTAL DERIVATIVE

In the case of an ideal incompressible fluid, we prove that, for a quantity ϕ , we have the following relation :

$$\int_{-d}^{\eta} \frac{D}{Dt} (\phi) dy = \partial_t \int_{-d}^{\eta} \phi dy + \nabla \cdot \int_{-d}^{\eta} \mathbf{u} \phi dy \quad (\text{A.2})$$

Proof.

$$\begin{aligned}
\int_{-d}^{\eta} \frac{D}{Dt}(\phi) \, dy &= \int_{-d}^{\eta} \partial_t \phi + \mathbf{u} \cdot \nabla \phi + v \partial_y \phi \, dy \\
&= \partial_t \int_{-d}^{\eta} \phi \, dy - \phi_s \partial_t \eta - \phi_b \partial_t d + \int_{-d}^{\eta} \nabla \cdot (\mathbf{u} \phi) \, dy - \int_{-d}^{\eta} \phi (\nabla \cdot \mathbf{u}) \\
&\quad + \int_{-d}^{\eta} \partial_y (v \phi) \, dy - \int_{-d}^{\eta} \phi \partial_y v \, dy \\
&= \partial_t \int_{-d}^{\eta} \phi \, dy + \nabla \cdot \int_{-d}^{\eta} \mathbf{u} \phi \, dy - \phi_s v_s - \phi_b v_b + [v \phi]_{-d}^{\eta} \\
&\quad - \int_{-d}^{\eta} \phi \left(\underbrace{\nabla \cdot \mathbf{u} + \partial_y v}_{=0} \right) \\
&= \partial_t \int_{-d}^{\eta} \phi \, dy + \nabla \cdot \int_{-d}^{\eta} \mathbf{u} \phi \, dy
\end{aligned}$$

□

BIBLIOGRAPHY

- Clamond, D. (2006). A note on time integrators in water-wave simulations. (Cited page 12.)
- Eichwald, B. (2013). Intégrateur exponentiel modifié pour la simulation des vagues non linéaires. (Cited pages 4, 9, 13, 14, 16, 17, and 19.)
- Fernando J. Seabra-Santos, D. P. R., Andr and eacute; M. Temperville (1987). Numerical and experimental study of the transformation of a solitary wave over a shelf or isolated obstacle. (Cited pages 6 and 27.)
- Gustafsson, K., Lundh, M. and Söderlind, G. (1988). A PI stepsize control for the numerical solution of ordinary differential equations. (Cited page 13.)
- Hairer, E., Nørsett, S. P. and Wanner, G. (2004). Solving Ordinary Differential Equations I: Nonstiff Problems. Springer. (Cited pages 11, 13, and 14.)
- Mitsotakis, D. (2013). A simple introduction to water waves. (Cited page 21.)
- Serre, F. (1953). Contribution à l'étude des écoulements permanents et variables dans les canaux. (Cited page 6.)

LIST OF FIGURES

1.1	domain definition	3
1.2	Cnoidal wave profiles for different values of the parameter m ($m = 0$ (cyan), $m = 0.9$ (red), $m = 0.99999$ (black)). The grey curve represents the still water level. (Author Kraaiennest, CC-BY-SA)	5
2.1	time-steps for the classic, IF and MIF methods	18

Titre Modélisation et simulation numérique des vagues non linéaires

Résumé Les vagues extrêmes (tsunamis, vagues scélérates, etc.) jouent un rôle important dans l'environnement et les risques naturels. Ces phénomènes sont toujours mal compris et leur compréhension passe, entre autres, par la simulation numérique. Cette simulation est d'autant plus difficile que les vagues sont extrêmes, les domaines grands et les temps longs, ces circonstances correspondant aux situations réelles.

Ce rapport relate les techniques de simulation numérique précises qui ont été développées et mises en place, permettant ainsi la simulation sur des temps longs et des domaines grands. Ces techniques sont basées sur les méthodes du *Facteur Intégrant* et du *Facteur Intégrant modifié*.

Dans le but d'avoir une simulation au plus proche de la réalité, ce rapport expose aussi la généralisation des équations de Serre – Green-Nagdhi à un fond variable en temps et en espace.

Mots-clés TN10, Eaux peu-profondes, équations de Serre, vagues non-linéaires, dense output

Title Modeling and numerical simulation of non-linear water-waves

Abstract Extreme waves (tsunamis, rogue waves) play an important role in the environment and natural hazards. These phenomena are still misunderstood and their understanding requires, *inter alia*, numerical simulation. The simulation is harder the more waves are extreme, the domains big and the times long ; those circumstances corresponding to actual situations.

This report relates the precise numerical simulation techniques that have been developed and used, enabling simulations over big domains and long times to be done. Those techniques involve the *Integrating Factor* and the *Modified Integrating Factor* methods.

In order to have a simulation as closed from the reality as possible, this report shows also the generalization of the Serre – Green-Nagdhi equations to a variable bottom in time and space.

Keywords TN10, Shallow water, Serre Green-Nagdhi, non-linear water-waves, dense output