DEGREE PROJECT IN TECHNOLOGY,
FIRST CYCLE, 15 CREDITS
STOCKHOLM, SWEDEN 2021

Surf Simulation with the Shallow Water Equations

Coupling of a surfer model to a shallow water wave

Anton Thorell

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ENGINEERING SCIENCES

Abstract

This thesis covers the subject of deriving and solving the system of partial differential equations known as the Shallow Water Equations (SWE), and coupling the solutions of this equation system to a simplified model of a surfer - or any floating object, with the right choice of parameters. The SWE are generally used to analyze fluid movement on shallow areas, such as ocean waves nearing the shore, and are derived from the Navier-Stokes equations and restricted to conservation of mass and momentum in a fluid. In this project the solution to these equations was made to yield a propagating wave profile. From the solution, the steepening behaviour and the slow propagation speed of shallow water waves are explained - properties that are necessary for wave surfing. The SWE was solved with the first order accurate finite volume scheme known as the Lax Friedrichs Method (LxF), and a surfable wave is created with a suitable set of initial- and boundary conditions and parameter values. LxF is also derived from the discretization of the conservation form of the SWE. The solver can handle a non-horizontal seabed - bathymetry - but does not take into consideration friction from the seabed. A ”surfer” was created as a point mass acted on by three forces: hydrodynamic drag, gravity and buoyancy. The ”surfer” is made to move realistically by simulating the effect of these forces and updating position and velocity of the surfer accordingly. The surfer is made to move along with the wave.

Key words

Bachelor degree project, Shallow Water Equations, surf simulation, surf waves, numerical analysis, finite volume method, Lax-Friedrichs Method, Godunov scheme, bathymetry.

Sammanfattning

Denna uppsats behandlar härledning av, och lösningen till det system av partiella differentialekvationer som kallas Shallow Water-ekvationerna (SWE), samt att koppla lösningarna till detta ekvationssystem till en förenklad modell av en surfare - eller något annat flytande objekt, medelst ett rimligt val av parametrar. SWE används generellt för att analysera vätskerörelser över grunda områden, såsom havsvågor som närmar sig en strand, och härleds från Navier-Stokes-ekvationerna och begränsas till bevarande av massa och rörelsemängd i en vätska. I detta projekt syftade lösningen på dessa ekvationer till att skapa en fortplantande vågprofil. Utifrån denna lösning förklaras det uppbrantande beteendet och den långsammare utbredningshastigheten hos vågor på grunt vatten - egenskaper som är nödvändiga för vågsurfing. SWE löstes med den finita volymmetod av första ordningens noggrannhet som kallas Lax Friedrichs Metod (LxF), och en surfbar våg genererades med en lämplig uppsättning initial- och gränsvillkor och val av parametervärden. LxF härleds också, utifrån en diskretiserad version av den konservativa formen för SWE. Lösaren kan också hantera en icke-horisontell havsbotten - batymetri - men tar inte i beaktande friktion från havsbotten. En surfare genererades som en punktmassa som påverkas av tre krafter: hydrodynamisk drag, tyngdkraft och flytkraft. Surfaren förmåddes att röra sig på ett realistiskt sätt genom att simulera effekten av dessa krafter och uppdatera positionen och hastigheten för surfaren därefter. Surfaren förmåddes att röra sig med vågen.

Nyckelord

Kandidatexamensarbete, Shallow Water-ekvationerna, surfsimulering, surfvågor, havsvågor, surfning, numerisk analys, batymetri.

Acknowledgements

I would like to thank my supervisor Johan for the assistance with the surfer model and source material, and for answers to all questions asked. Also, my fellow student Frida Thörn deserves a thank you for discussion about the generation of shallow water waves with the Lax-Friedrichs method.

Author

Anton Thorell <antontho@kth.se>
CFATE
School of Engineering Sciences
KTH Royal Institute of Technology

Examiner

Elias Jarlebring
Department of Mathematics,
Division of Numerical Analysis
KTH Royal Institute of Technology

Supervisor

Johan Wärnegård <jwar@kth.se>
Department of Mathematics,
Division of Numerical Analysis
KTH Royal Institute of Technology

Contents

1 Introduction

Waves on shallow water have a different behaviour than waves on deep water. As a wave approaches shallower waters it slows down, becomes higher and, more importantly for this project, the wave front becomes steeper. This kind of wave needs to be analyzed with different methods than regular surface waves. The Shallow Water Equations (SWE) are used in a large variety of applications, for example analyzing dam-breaking problems, tsunamis, flows in lakes, tidal water movement etc [12].

To analyze the movement of a fluid consistent with shallow water theory numerical methods are needed as the differential equations that constitute the SWE cannot be solved analytically. This project has tried to make a simulation of a water wave consistent with shallow water theory, coupled with a physical model of a surfer.

1.1 Background

The definitions of ”shallow” and ”deep” here depend on the length scale of the wave versus the depth of the fluid. A more thorough explanation of this is given in section 2.1. An example of shallow water waves could be a tsunami nearing the shores of an ocean; the tsunami has such a large wavelength that compared to the depth of the ocean the wave could be considered ”shallow”. The shallow water equations are derived from the Navier-Stokes equations which are in turn derived from the conservation of mass and momentum of a fluid moving through a finite volume. See sections 2.1.1 and 2.1.2 for these derivations.

1.1.1 The Shallow Water Model

The SWE-model in one dimension (propagation only in one horizontal direction) - and here with a flat seafloor - consists of two equations: one for the conservation of mass;

$$h_t + \big[hv\big]_x = 0 \tag{1.1}$$

and one for conservation of momentum;

$$\big[hv\big]_t + \big[hv^2 + \frac{1}{2}gh^2\big]_x = 0, \tag{1.2}$$

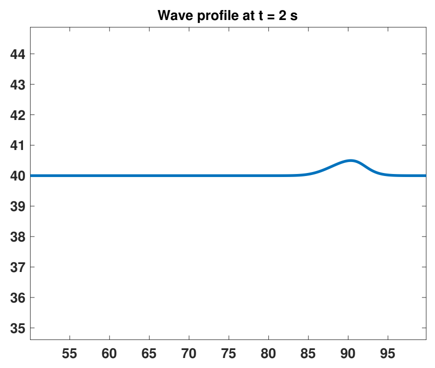
where g is the gravitational acceleration, h is the depth or height above the seafloor and hv is momentum, with v being the horizontal velocity of the water hill. Equations (1.1) and (1.2) are explored in greater detail in section 2.1. The subscripts stand for derivatives with respect to the indicated variable. These are the *one-dimensional* shallow water equations, meaning that water movement is neglected in all directions but in the horizontal x-direction.

1.1.2 Water velocity

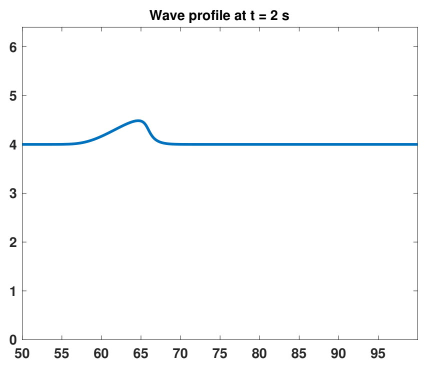
The fundamental differences between shallow water waves and waves on deep water are that deep water waves have a higher propagation speed and keep a constant shape, whereas shallow water waves move slower and get a distorted shape. This is because the *characteristic speed*, c, for a wave, derived from equations (2.5) and (2.10), for a (local) part of the wave depends on $h(x)$:

$$c(x) = v(x) + \sqrt{gh(x)}. \tag{1.3}$$

That is, different parts of the wave have different speeds, which is much more noticeable at shallow water than on deep water; when on deep waters, c is approximately constant for all x because the depth is much larger than the wave hill, resulting in a negligible distortion of the wave profile. On shallow waters, the crest will move faster than the trough in front of the wave - because the water is noticeably deeper at the crest - making the wave front steeper, until the crest eventually ”catches up” with the wave front and the wave breaks [8]. See the comparison in figures 1.1a and 1.1b, where only the depth differs between the two cases. This project did not cover the breaking of waves as that is a different area of physics and mathematics. In any case, it is the steepening behaviour of ocean waves on shallow water that surfers use to propagate themselves, and this will not be, if noticed at all, an obvious feature of waves on deeper water.



(a) Wave on deep water



(b) Wave on shalllow water

1.1.3 Surfer

The surfer will be affected by three essential forces; gravity pulling it down the wave slope; a drag force tangent to the slope and a buoyant force acting normal to the slope [11], see figure 1.1. Under the right conditions the surfer can be put in a state of equilibrium where it is ”pushed” forward by the buoyancy, hindered to drift forward by the drag from the water and at the same time sliding, due to gravity, down the wave front - which is called surfing.

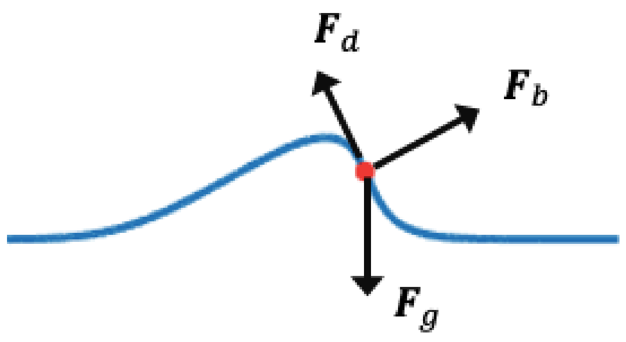


Figure 1.1: Force diagram of surfer on a wave.

1.1.4 Numerical Method

The SWE are a system of nonlinear partial differential equations that need numerical methods to be solved. In this project a first order finite volume method called The Lax-Friedrich scheme was used.

1.2 Purpose

The purpose of this thesis is to display what have been learned - through the degree project - about the shallow water equations, how to solve them, how one can couple a simple but in large physically correct model of a surfer to the SWE.

1.3 Goal

The goal of this degree project was to provide an answer to the following: Is there a finite volume method that will work well in simulating a surfable wave from the shallow water equations, including the case with bathymetry? Could this solution then be coupled with a surfer model similar to the one in the article ”How to ride a wave: The mechanics of surfing” [11] without violating any physical laws or the result in the article - that the slope interval $[-35°, 0°]$ on the concave part of the wave front is the only region on a wave suitable for stable surfing?

1.4 Outline

In the following sections the SWE will be derived, with addition of the case with bathymetry, the numerical method to solve them is derived and presented, the concept of characteristics and the linearized solution are explained, and the theory behind the surfer model is touched on. Then, the work of the author and implementation of all this is explained and results are presented. In the last sections the work done and possible future work are discussed and conclusions from the project are drawn.

2 Method and theory

As the method and theory were so closely connected in the project, they are handled in the same section of this report. In this section the derivation of the SWE and numerical method LxF used are explained, together with explanations of essential concepts as characteristc speed and the CFL-condition, and the theoretical basis of the surfer model.

2.1 The Shallow Water Equations

This derivation of the shallow water equations was based on the derivations made in [6] and [8]. To start with, the environment studied is a one-dimensional pipe of sorts, where some quantities $q(x,t)$ vary only along the length of the pipe. Here a finite section of the pipe between x_1 and x_2 is studied. Inside this section of the pipe, no amount of q is neither created nor destroyed - the quantities are conserved. So the change of q inside the section can only be due to the flow of q, or the flux $f\big(q(x,t)\big)$, into the section. This flux can then be described on integral form as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_1}^{x_2} q(x,t)\,\mathrm{d}x = -f\big(q(x,t)\big)\Big|_{x_1}^{x_2}. \tag{2.1}$$

Note that a flux < 0 is a quantity flowing out of the section, and thus x_2 is the section's downstream boundary. If q and f are smooth functions equation (2.1) can be manipulated into

$$\int_{x_1}^{x_2}\left[\frac{\partial}{\partial t}q(x,t) + \frac{\partial}{\partial x}f\big(q(x,t)\big)\right]\mathrm{d}x = 0. \tag{2.2}$$

As the integrand must equal zero this can be written more concisely as:

$$q_t(x,t) + f_x\big(q(x,t)\big) = 0 \tag{2.3}$$

where the subscripts indicate derivatives [6]. From the conserved quantities in q - which will turn out to be mass and linear momentum in the studied case - the height (or profile) of the shallow water wave and its propagation speed is the quantities sought after.

2.1.1 Conservation of mass

To begin with, the quantity of depth or height of the water $h(x,t)$ is to be derived, that is, $q(x,t)$ in equation (2.3). Now a channel with unit width is studied instead of a one dimensional pipe. A specific density $\tilde{\rho}$ with units mass over length squared can then be used. The water level is allowed to vary along the horizontal axis. The total mass in a section of the channel, between boundaries at x_1 and x_2 at time t is

$$\int_{x_1}^{x_2} \tilde{\rho}(x,t)h(x,t)\,\mathrm{d}x$$

and the change of mass in this section of the channel would be the time-derivative of this; the left hand side of equation (2.1). The flux on the right hand side could be described as the product between the fluid's velocity and the conserved quantity, or the flow rate past a point: qv. With mass as the conserved quantity, $\tilde{\rho}(x,t)h(x,t)v(x,t)$ constitutes the mass flux. The conservation of mass can be described in accordance to equation (2.3):

$$\left[\tilde{\rho}(x,t)h(x,t)\right]_t + \left[\tilde{\rho}(x,t)h(x,t)v(x,t)\right]_x = 0. \tag{2.4}$$

The observed fluid is water though - an incompressible fluid - so the density is constant and can be cancelled out and equation (2.4) simplifies to

$$h_t + (hv)_x = 0. \tag{2.5}$$

Here there are two unknowns, h and hv, and an additional equation is needed.

2.1.2 Conservation of momentum

The velocity v above is not a conserved quantity in itself, but the momentum ρv, is. So the same as done above for mass, can be done for the density quantity of momentum. The total amount of momentum in the section is:

$$\int_{x_1}^{x_2} \tilde{\rho}(x,t)h(x,t)v(x,t)\,\mathrm{d}x$$

and the change of momentum density is thus

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_1}^{x_2}\tilde{\rho}(x,t)h(x,t)v(x,t)\,\mathrm{d}x. \tag{2.6}$$

However, the change of momentum density in the observed section is is caused by two different kinds of momentum flux. First, the *convective* flow

$$qv = (\tilde{\rho}v)v = \tilde{\rho}v^2,$$

of the the quantity, caused by the macroscopic movement in or out of the section - compare this with the case of mass flux above. And second, a microscopic flow of particle momentum originating from the static pressure p at the boundaries of the section. The pressure at a some depth $h - y$ in the fluid is caused by the specific weight of the fluid above. If this is integrated over the whole water column from $y = 0$ to $y = h$ at (x,t), and then also in the x-direction, a force is generated. Remember that force integrated over time is momentum - the sought after quantity. These two terms should balance out (2.6), as shown in equation (2.1):

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{x_1}^{x_2}\tilde{\rho}hv\,\mathrm{d}x = -\int_0^h \tilde{\rho}g(h-y)\,\mathrm{d}y\,\bigg|_{x_1}^{x_2} - \int_0^h \tilde{\rho}v^2\,\mathrm{d}y\,\bigg|_{x_1}^{x_2} \tag{2.7}$$

which leads to

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\rho}hv = -\frac{\partial}{\partial x}\left[\frac{1}{2}\tilde{\rho}gh^2 + \tilde{\rho}v^2h\right]_{x_1}^{x_2}. \tag{2.8}$$

This is re-written in a more concise way as

$$(\tilde{\rho}hv)_t + (\tilde{\rho}hv^2 + \frac{1}{2}\tilde{\rho}gh^2)_x = 0. \tag{2.9}$$

As water is an incompressible fluid, $\tilde{\rho}$ can be canceled out and the sought after momentum equation introduced in section 1.1.1 is obtained:

$$(hv)_t + \left(hv^2 + \frac{1}{2}gh^2\right)_x = 0. \tag{2.10}$$

2.1.3 System of equations

As the last step we combine (2.5) and (2.10) into the system of one-dimensional shallow water equations

$$\begin{bmatrix} h \\ hv \end{bmatrix}_t + \begin{bmatrix} hv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}_x = 0. \tag{2.11}$$

(2.11) can be written as a generic hyperbolic system on conservative form:

$$q_t + f_x(q) = 0 \tag{2.12}$$

where

$$f : \mathbb{R}^2 \to \mathbb{R}^2, \; q = \begin{bmatrix} h \\ hv \end{bmatrix}.$$

Notice that the equations in (2.11) is a simplified shallow water model - it does not account for friction against the seafloor, or a changing shape of the seafloor. The latter will be explored next though.

2.1.4 Bathymetry

The derivation above is done for a case where the bottom of the channel has a constant, flat shape. But the shape of the seafloor is often changing considerably over the observed spatial domain. For example when examining waves closing in on the shore, the shoaling must be considered. A global vertical coordinate $\eta(x)$ of the water surface is introduced, and this is the depth of the water *plus* the bathymetry or topology of the seabed $B(x)$. When deriving the momentum flux (2.10), the pressure is integrated over a water column to find the forces on the fluid, (2.7). So when there is a changing seabed, the topology of the seabed must be considered. The integration of pressure on the right hand side in (2.7) is altered to handle this and becomes

$$\begin{aligned}
-\int_{B(x)}^{\eta} \tilde{\rho} g(\eta - y)\,\mathrm{d}y &= \\
= -\int_{B(x)}^{h+B(x)} \tilde{\rho} g(h + B(x) - y)\,\mathrm{d}y &= \\
= -\int_{0}^{h} \tilde{\rho} g(h + B(x) - y)\,\mathrm{d}y &= \\
= \tilde{\rho} g h\big(\frac{h}{2} + B(x)\big)&
\end{aligned}$$

Canceling out the density and rearranging generates the SWE with a bathymetry source term [1]:

$$\begin{bmatrix} h \\ hv \end{bmatrix}_t + \begin{bmatrix} hv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}_x = -gh \begin{bmatrix} 0 \\ B(x) \end{bmatrix}_x . \tag{2.13}$$

2.2 The linearized case

To get a sense of what the characteristic speeds (1.3) in section 1.1.2 are and where they come from, a simplified case will be examined. The equation system (2.12) can be simplified by linearization. The flux $f(q)$ can be rewritten by using the jacobian matrix

$$A = \begin{bmatrix} \frac{\partial f^1}{\partial q^1} & \frac{\partial f^1}{\partial q^2} \\ \\ \frac{\partial f^2}{\partial q^1} & \frac{\partial f^2}{\partial q^2} \end{bmatrix}$$

and (2.12) can be rewritten on quasi-linear form as a hyperbolic system of equations

$$q_t + A(q)q_x = 0. \tag{2.14}$$

For the system (2.14) to truly be hyperbolic, the eigenvalues of A needs to be distinct and real [4]. So let us confirm this. If the notation $h = q^1$ and $hv = q^2$ is used, f and A can be written as

$$f(q) = \begin{bmatrix} q^2 \\ q^{(2)^2}/q^1 + \frac{1}{2}gq^{(1)^2} \end{bmatrix} \Rightarrow A = \begin{bmatrix} 0 & 1 \\ \\ -\frac{q^{(2)^2}}{q^1} + gq^1 & \frac{2q^2}{q^1} \end{bmatrix}.$$

If a constant state $q_0 = [h_0, 0]$, the "lake at rest state", [9] is considered, and a small deviation $\tilde{q}$ is added to this

$$q = q_0 + \tilde{q} \tag{2.15}$$

is obtained. As $v_0 = 0 \iff q_2 = 0$, A simplifies to

$$A = \begin{bmatrix} 0 & 1 \\ gh_0 & 0 \end{bmatrix}.$$

It is a straightforward process to conclude that the eigenvalues are

$$\lambda_{1,2} = \pm\sqrt{gh_0}$$

where h_0 is the height of the still water. These are clearly real and distinct and thus the system is hyperbolic.

2.2.1 Diagonalization and characteristic variables

To solve the system (2.14) the equations can be decoupled by diagonalization. That is, re-writing A as a diagonal matrix Λ with the eigenvalues on the diagonal. The eigenvectors corresponding to the eigenvalues λ_1 and λ_2 are

$$r_1 = \begin{bmatrix} -1/\sqrt{gh_0} \\ 1 \end{bmatrix} \text{ and } r_2 = \begin{bmatrix} 1/\sqrt{gh_0} \\ 1 \end{bmatrix}.$$

These vectors can be put as columns in a matrix:

$$R = \begin{bmatrix} -1/\sqrt{gh_0} & 1/\sqrt{gh_0} \\ 1 & 1 \end{bmatrix}.$$

A can then be written as $A = R\Lambda R^{-1}$. Now, some new *characteristic variables* is introduced, which are defined as

$$w(x,t) = R^{-1}q(x,t) \tag{2.16}$$

and the system (2.14) can be re-written as

$$w_t + \Lambda w_x = 0. \tag{2.17}$$

So if $A \in \mathbb{R}^{m\times m}$, m decoupled equations is obtained. If the initial state is

$$q(x,0) = \mathring{q}(x)$$

then

$$\mathring{w}(x) = R^{-1}\mathring{q}(x).$$

With $p = 1, 2, ..., m$ the pth equation is obtained:

$$w_t^p + \lambda^p w_x^p = 0. \tag{2.18}$$

with solutions as

$$w^p(x,t) = \mathring{w}^p(x - \lambda^p t)$$

which can be solved for q by reversing the process in equation (2.16). This means that the initial state will be propagating with speed λ^p. In this particular case that means two waves moving in opposite directions with speeds $\pm\sqrt{gh_0}$. This is of course not physically correct if the wave actually propagates on shallow water but could be accurate for other kinds of waves. The shallow water wave compared with the linearized surface wave are visualized in in fig 3.2.

2.3 Characteristics

In the linearized case above, the eigenvalues are constant. That means that the disturbance will move along an unchanged path in the $x - t$-space, called a characteristic curve. This curve could also be interpreted as a ray in the direction pointed out by the corresponding eigenvector multiplied with the characteristic variable. See figure 2.1 for an example with $p = 3$. The Characteristics tell us how fast the information in $q(x,t)$ is traveling.

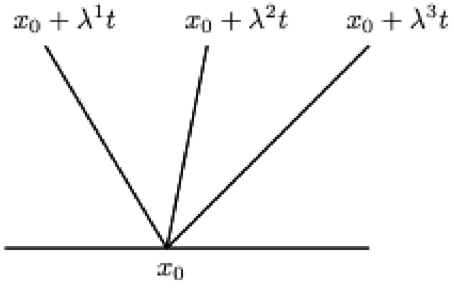


Figure 2.1: Characteristic rays $x_0 + \lambda^p t$

2.4 Non-linear case

If the deviation $\tilde{q}$ is considerable compared to the water depth, or approximately

$$\frac{q_0}{\tilde{q}} \geq 20$$

[12], $q(x,t)$ cannot be calculated in the same way done in section 2.2 as the jacobian matrix in (2.14) is not constant in x. This means that the eigenvalues will change depending on the location, and a general ”wave speed” cannot be concluded. Every point on the water surface will get its own characteristic speed, faster the deeper the water below is. The solution needs to be approximated with a numerical method.

2.5 Numerical Method

This section will explain the derivation of a time marching numerical method suitable to calculating the SWE: The Lax-Friedrichs finite volume method. See [7] for additional explanations.

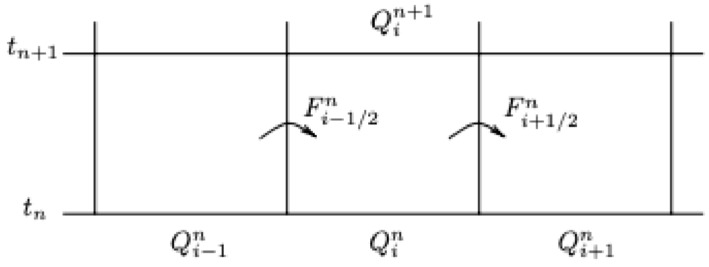


Figure 2.2: Discretized structure of the method

2.5.1 Discretization

The environment to work with is discretized in I number of cells along the horizontal axis. The width of these cells, Δx, is then L/I with L being the length of the channel to be studied. The time-domain extension of the cells is determined by the time step, which is just a multiple of Δx: $\Delta t = C\Delta x$. How to choose C is described in 2.5.7. The time axis is discretized into N values by choosing an end time T and dividing it by the time step: $N = T/\Delta t$.

2.5.2 Finite Differences

The equation (2.12) can be written as a finite difference approximation

$$\frac{Q_i^{n+1} - Q_i^n}{\Delta t} + \frac{F_{i+1/2}^n - F_{i-1/2}^n}{\Delta x} = 0 \tag{2.19}$$

where Q_i^n are the values of the quantities inside a cell (as visualized in figure 2.2), analogous to the quantities $q(x,t)$ in the analytical case before. $F_{i-1/2}^n$ is a numerical approximation of the average flux at $x = x_{i-1/2}$, with i and n being the indices for the discretized points on the horizontal axis and the time axis, respectively. See fig. 2.2. Equation (2.19) can be written on the form

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+1/2}^n - F_{i-1/2}^n\right) \tag{2.20}$$

to get a time marching scheme. Note that the number $\frac{\Delta t}{\Delta x} = \alpha$ has units time/length - the inverse of speed - which goes back to the statement in 2.1 that the flux is just the quantity times the speed of the fluid past a point.

2.5.3 Numerical flux

As described in 2.1, the flux of quantities can be calculated from the values of the quantities themselves $q(x,t)$. In this discretized environment a numerical flux function $\Phi(Q_{i-1}^n, Q_i^n)$ can approximate the numerical flux $F_{i-1/2}^n$ by averaging the values of the function $f(Q)$ from equation (2.12), and inserting cell values Q_{i-1}^n and Q_i^n on either side of the the cell interface at $x_{i-1/2}$:

$$F_{i-1/2}^n = \Phi(Q_{i-1}^n, Q_i^n) = \frac{1}{2}\big[f(Q_{i-1}^n) + f(Q_i^n)\big]. \tag{2.21}$$

(2.21) can then be put into (2.20) to become

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{2\Delta x}\Big(f(Q_{i+1}^n) - f(Q_{i-1}^n)\Big). \tag{2.22}$$

2.5.4 The Lax-Friedrichs Method

Equation (2.22) is almost the same structure as the Lax-Friedrichs method (LxF), but a couple of changes are needed to make (2.22) stable. First, a diffusion term is added to the flux in equation (2.12):

$$q_t + f_x(q) = \beta q_{xx} \text{ with } \beta = \frac{(\Delta x)^2}{2\Delta t}.$$

Noticeable here is that the diffusion term vanishes as the grid is being refined. The expression for the numerical diffusion flux is

$$\Phi_{diff}(Q_{i-1}^n, Q_i^n) = -\beta_{i-1/2}\Big(\frac{Q_i^n - Q_{i-1}^n}{\Delta x}\Big). \tag{2.23}$$

The other change to (2.22), is that the values Q_i^n are replaced by an average $\frac{1}{2}(Q_{i+1}^n + Q_{i-1}^n)$ from neighbouring cells. This, together with the diffusion flux, can be put into (2.22) to form the Lax-Friedrichs Scheme:

$$Q_i^{n+1} = \frac{1}{2}(Q_{i+1}^n + Q_{i-1}^n) - \frac{\Delta t}{\Delta x}\Big(\Phi(Q_i^n, Q_{i+1}^n) - \Phi(Q_{i-1}^n, Q_i^n)\Big) \tag{2.24}$$

that uses the flux function

$$\Phi(Q_{i-1}^n, Q_i^n) = \frac{1}{2}\Big(f(Q_{i-1}^n) + f(Q_i^n)\Big) - \frac{\Delta x}{2\Delta t}(Q_i^n - Q_{i-1}^n). \tag{2.25}$$

2.5.5 Ghost Cells

The scheme in (2.24) uses averages of the neighbouring cells at time t^n to calculate the values at time t^{n+1}. So something must be done when calculating the values at the edges at $x = a$ and $x = b$ as there are no values to the left at a and no cell values to the right at b. Some boundary conditions must be set. This can be done by adding so called ghost cells ”outside” of the physical domain, at $i = 0$ and

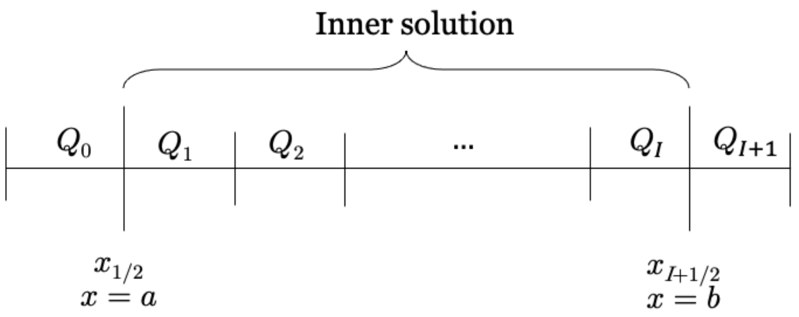


Figure 2.3: Setup with inner solution and ghost cells at either edge

$i = I + 1$. See figure 2.3. If

$$Q_i = \begin{bmatrix} q_i^1 \\ q_i^2 \end{bmatrix} = \begin{bmatrix} h(x_i, t) \\ h(x_i, t)v(x_i, t) \end{bmatrix}$$

are the cell values and the edges should be reflecting walls, the information in ghost cells Q_0 and Q_{I+1} are

$$\text{For } Q_0 : \quad q_0^1 = q_1^1, \text{ and } q_0^2 = -q_1^2$$

$$\text{For } Q_{I+1} : \quad q_{I+1}^1 = q_I^1, \text{ and } q_{I+1}^2 = -q_I^2$$

[5][2].

2.5.6 Bathymetry

To account for the effect that bathymetry has on the numerical solution, a source term S is added to (2.24) as in (2.13), but the continuous data is replaced by the discretized values and finite differences, and

$$S = -gq_i^1 \frac{\Delta t}{\Delta x}(B_{i+1} - B_{i-1}) \tag{2.26}$$

is added to the momentum part on the right side of (2.24).

2.5.7 The CFL-condition

It is important that the information in the numerical method does not travel faster than the actual physical information, or in nicer words: the method's domain of dependence should not go beyond the true domain of dependence of the PDE. This is a necessary condition for stability and is called the CFL-condition after Courant, Friedrich and Lax [7]. This means that

$$\frac{\Delta x}{\Delta t} < \max|\lambda^p|, \tag{2.27}$$

or that

$$\Delta x = C\Delta t, \;\; C < \max\left|q^2/q^1 + \sqrt{gq^1}\right|.$$

This means that the information needed to compute a value in the next time step, cannot lie outside the area where the actual physical information comes from. Here is an example with $p = 3$.

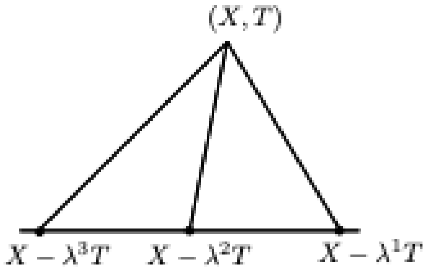


Figure 2.4: Domain of dependence for the point (X, T)

A value computed at (X, T) can only depend on values from within the interval $[X - \lambda^3 T, X - \lambda^1 T]$ in order for the method to be convergent.

2.5.8 Convergence analysis

The Lax-Friedrichs method is a first order accurate method. That means that if the step-size δ of the numerical method is changed by a factor α the error of the numerical approximation would change by the same factor. Let us call these approximations $\tilde{u}_1, \tilde{u}_2, ..., \tilde{u}_N$. If the grid size is decreased by a factor $0 < \alpha < 1$ for every approximation in the sequence then $\tilde{u}_1 = \tilde{u}_{\alpha^0\delta},\; \tilde{u}_2 = \tilde{u}_{\alpha^1\delta}, ..., \tilde{u}_N = \tilde{u}_{\alpha^{N-1}\delta}$. Generally, if u is the exact solution and $\tilde{u}$ is the approximation, the method would converge if

$$|\tilde{u_\delta} - u| \leq C\delta^p \tag{2.28}$$

for sufficently small δ. Here C is a constant and p is the order of convergence. The convergence rate is said to be δ^p. Assuming δ is small enough the order of convergence can be found by looking at ratios of errors computed from different grid sizes:

$$\frac{\tilde{u}_1 - u}{\tilde{u}_2 - u} = \frac{\tilde{u}_\delta - u}{\tilde{u}_{\alpha\delta} - u} = \frac{C\delta^p}{C(\alpha\delta)^p} = \alpha^{-p}.$$

The problem here is that there is no exact solution to compare with, and to set an extremely fine grid size to simulate a solution u closely cannot be done as that would be too costly. Though, approximations with not so extreme grid sizes can be computed and compared with a *more exact* approximation:

$$\frac{\tilde{u}_1 - \tilde{u}_2}{\tilde{u}_2 - \tilde{u}_3} = \frac{\tilde{u}_\delta - \tilde{u}_{\alpha\delta}}{\tilde{u}_{\alpha\delta} - \tilde{u}_{\alpha^2\delta}} = \frac{C\delta^p - C(\alpha\delta)^p}{C(\alpha\delta)^p - C(\alpha^2\delta)^p} = \frac{1-\alpha^p}{\alpha^p - \alpha^{2p}} = \frac{1-\alpha^p}{\alpha^p(1-\alpha^p)} = \alpha^{-p}.$$

Here the logarithm with base $1/\alpha$ is used to find p. By increasing the number of times the grid size is reduced, or increasing N, we can get an increasingly better calculation of p, as done in section 3.5 [10].

2.6 Surfer

As said in section 1.1.3 the surfer is affected by three essential forces: The bouyancy $\mathbf{F}_b$ normal to the surface, gravity $\mathbf{F}_g$ and a drag force along the surface $\mathbf{F}_d$. The surfer is here represented as a point mass and cannot jump off the surface or sink below it [11].

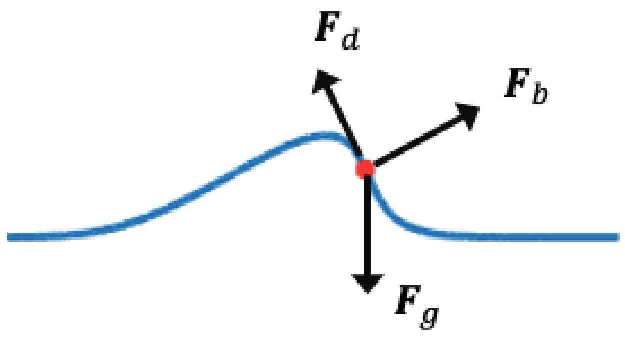


Figure 2.5: Forces acting on ”surfer” on a wave

2.6.1 Coordinate system

The coordinate system used follows the surfer along its path on the surface. To be able to calculate the acceleration and position in 3.4, arising from the forces mentioned, it is essential to first determine the slope of the surface where the surfer is, and from the slope: the tangential and normal unit vectors $\hat{\mathbf{t}}$ and $\hat{\mathbf{n}}$. See section 3.3 for an explanation of how this is done.

3 Result and Contribution of the author

In this section the results from the author's work are described, on basis of the theory and methodology described in previous sections. Because the goal of the project is so much about implementing theory, the new contributions from the author is very closely related to the results, but also to the theory and methodology part of the thesis. For all programming MATLAB was used.

3.1 Propagation of the linearized solution

The result from 2.2.1 was examined by letting the solutions arising from equation (2.18) propagate. An initial wave profile was simulated with a Gauss curve at the middle of the channel:

$$h(x,0) = h_0 + \varepsilon e^{-(x-L/2)^2/W^2} \tag{3.1}$$

which propagates (approximately) with speed $\sqrt{gh_0}$, as explained in section 2.2.1. Here L is the length of the "channel", ε is the height of the disturbance, and W is the width of the disturbance. This means that the wave profile at time t will be

$$h(x,t) = h_0 + \varepsilon e^{-(x-\lambda_1 t-L/2)^2/W^2} = h_0 + \varepsilon e^{-(x+\sqrt{gh_0}t-L/2)^2/W^2}$$

for the right moving wave. Both propagating waves are visualized in figure 3.1 and a comparison between wave profiles with and without shallow water effects are visualized in figure 3.2.

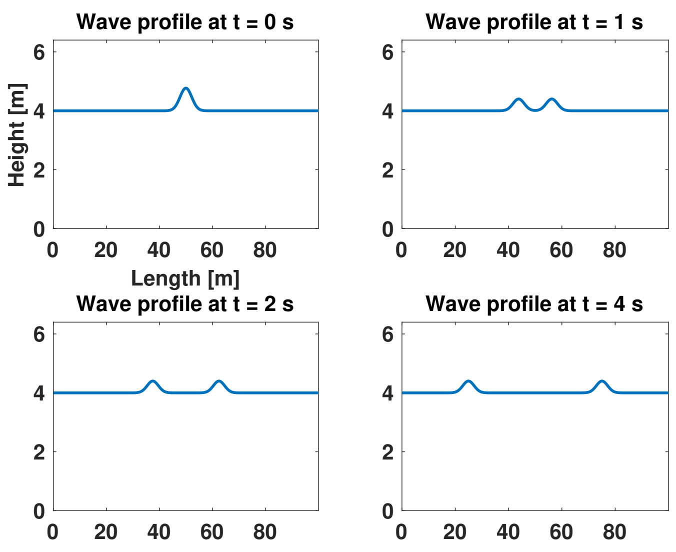


Figure 3.1: Linearized wave, without shallow water effects

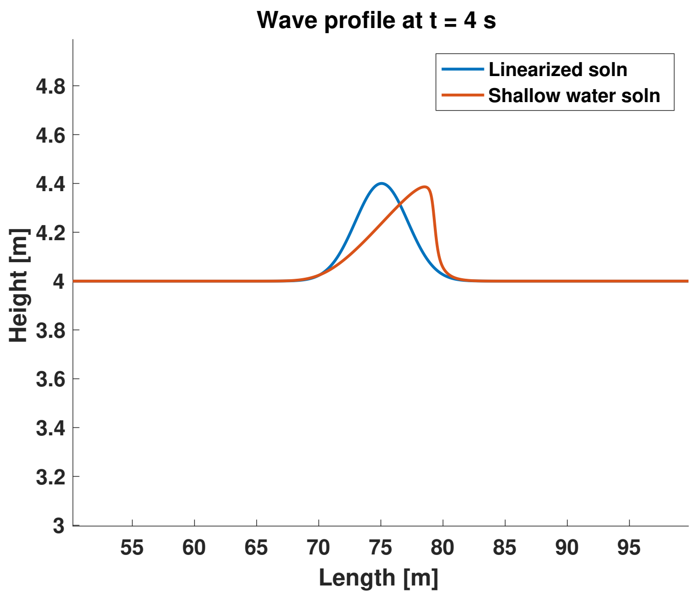


Figure 3.2: Comparison of solutions

3.2 Generating the shallow water wave

A disturbance was generated in the same way as in 3.1 by starting with a Gauss-curve profile and defining the parameters in (3.1). The boundary conditions, i.e. the ghost cells, were set as in 2.5.5. The solution was then propagated by calculating the cell values in next time step by using the Lax-Friedrichs method (2.24) for every time step until the set final time $T = t_N$ is reached. If the case with bathymetry is examined, the source term (2.26) is added to the momentum part of (2.24).

3.3 Surfer

How the physical properties of the surfer was generated is explained in this section.

3.3.1 Paddling

In order for a surfer to ”catch” an incoming wave it has to paddle and build up some speed before standing up on the board and start to surf. This was simulated by assigning a suitable initial tangential velocity $V^{n=1}$ to the surfer.

3.3.2 Slope and Coordinate system

The model only allows for movement tangential to the water surface. If U^1_ι is the cell value of water surface height at the horizontal position of the surfer x_ι - obtained with the MATLAB-function interp1 - the slope Γ can be approximated by

$$\Gamma = \frac{U^1_{\iota+1} - U^1_{\iota-1}}{2\Delta x}. \tag{3.2}$$

From this the tangential unit vector was obtained:

$$\hat{\mathbf{t}} = \frac{\mathbf{T}}{||\mathbf{T}||}, \qquad \mathbf{T} = \begin{bmatrix} x \\ \Gamma x \end{bmatrix}.$$

The normal unit vector was computed with a $\theta = 90°$ rotational transformation of $\hat{\mathbf{t}}$:

$$\hat{\mathbf{n}} = \mathbf{B}\hat{\mathbf{t}} = \begin{bmatrix} -\hat{t}^2 \\ \hat{t}^1 \end{bmatrix}, \qquad \mathbf{B} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}.$$

3.3.3 Drag

The drag used is only the regular hydrodynamic force from fluid mechanics

$$F_d = -\frac{1}{2}\rho C_d A_{cr} V^2 \tag{3.3}$$

where ρ is the water density, C_d is the drag coefficient, A_{cr} is the cross-section area, and V is the tangential velocity.

3.3.4 Gravity

The tangential component of the gravity is

$$F_{g,t} = \begin{bmatrix} 0 \\ -mg \end{bmatrix} \cdot \hat{\mathbf{t}} \tag{3.4}$$

where $\hat{\mathbf{t}}$ is the tangential unit vector derived in 3.3.2.

3.3.5 Buoyancy

The last force is the buoyancy, acting normal to the surface. The board was assumed to always be floating, and this floating effect was simulated without any inertia, which is not completely physically correct of course, but good enough for the purpose. The thought is that if the surfer is below the water surface, either from sinking or that the wave is ”overtaking” the surfer, the position at the surface where it should be floating can be found by accurately guessing the distance the surfer would have to move in the normal direction until this position at the surface is found. This approximation was done by fix-point iteration, described in 3.4.2.

3.4 Coupled surfer and shallow water wave

How the surfer model is coupled with the solution from solving the shallow water equations with the Lax-Friedrichs method is explained in this section.

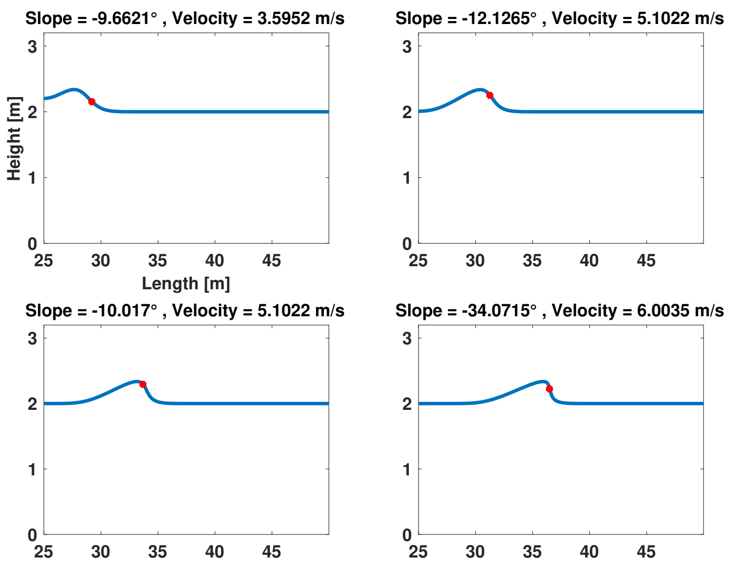


Figure 3.3: Surfing

3.4.1 Initial position

The surfer is deployed by assigning it a position on the horizontal axis by choosing a cell index ι using the round-function in MATLAB: $\iota = \text{round}(Id)$ where I is the number cells on the horizontal axis and d is how far out on the horizontal domain the surfer is placed as a fraction of the length L. That is, if $d = 0.5$ the surfer is placed in the middle of the horizontal domain. So the global horizontal position of the surfer ξ is initially $\xi = x_\iota$. The global vertical position of the surfer η should be on the surface, i.e. coincide with the solution for height at x_ι: $\eta = U_\iota^1$, if $\mathbf{U}_\iota := \left[U_\iota^1, U_\iota^2\right]$ is the solution when (2.24) is solved. The position $\mathbf{X}$ of the surfer is then $\mathbf{X} := \left[\xi, \eta\right]$.

3.4.2 Updating position

With an updated slope and thus a new tangent unit vector $\hat{\mathbf{t}}$ the position $\mathbf{X}$ of the surfer can be updated. This was done with Euler's method:

$$\mathbf{X}^n = \mathbf{X}^{n-1} + V^{n-1} \cdot \hat{\mathbf{t}}\,\Delta t. \tag{3.5}$$

Equation (3.5) is only a guess though, as the buoyancy is not taken into consideration yet; it allows the surfer to be above or below the surface, which is not consistent with the assumption that the surfer should always follow the surface. The height of the surface where the surfer is positioned is $h(\xi)$. To find this value in reality, the MATLAB-function interp1 was used. So, if $h(\xi) - \eta > 0$ the surfer is below the water surface. Here, fix-point iteration with $j = 1, 2, ..., J$ iterations was used to find how big of a step k in the normal direction was needed for the surfer to find its way back to the surface to simulate floating. The equation to be solved for k in the fix-point iteration is

$$h\big(\xi(k)\big) - \eta(k) = 0. \tag{3.6}$$

That is, the global vertical coordinate of the surfer should coincide with water depth or height at the same horizontal position. An initial guess for k was set to $k_0 = \Delta x$ and the number of iterations to $J = 10$. In each iteration j the position $\mathbf{X}^n$ from (3.5) was updated to "try" a new position along the normal:

$$\mathbf{X}^n = \mathbf{X}^{n-1} + V^{n-1} \cdot \hat{\mathbf{t}}\,\Delta t + k\,\hat{\mathbf{n}} \tag{3.7}$$

which leads to new $\xi(k)$ and $\eta(k)$ to be put into (3.6):

$$h\big(\xi(k)\big) - (\eta^{n-1} + V^{n-1} \cdot \hat{\mathbf{t}}^{(2)}\,\Delta t + k\,\hat{\mathbf{n}}^{(2)}) = 0 \tag{3.8}$$

which could be solved for k to get a new, better approximation of the step size:

$$k_{new} = \frac{h\big(\xi(k_{old})\big) - \eta^{n-1} - V^{n-1} \cdot \hat{\mathbf{t}}^{(2)}\,\Delta t}{\hat{\mathbf{n}}^{(2)}}. \tag{3.9}$$

For the case $h(\xi) - \eta < 0$, where the surfer is above the surface, the surfer simply "falls down" and $\eta = h(\xi)$.

3.4.3 Updating velocity

With the forces and a previous velocity provided, a new velocity was generated, also with Euler's method:

$$V^{n+1} = V^n + a^n \Delta t \tag{3.10}$$

where

$$a = \frac{1}{m}\big(F_{g,t} + F_d\big)$$

is the tangential acceleration and V^n is the previous tangential velocity.

3.5 Convergence analysis

A convergence test was made as per the theory in section 2.5.8, in order to confirm that the implementation of the Lax-Friedrichs method actually did converge like a first order method and that it was correctly implemented. Using $\alpha = 1/2$, that is, for every iteration of calculating an approximation the step size is halved, and with $N = 9$, using 9 approximations around a chosen x-value (here at 1.65 m), $N - 2 = 7$ resulting ratios

$$r = \frac{\tilde{u}_\delta - \tilde{u}_{\delta/2}}{\tilde{u}_{\delta/2} - \tilde{u}_{\delta/4}}$$

were be obtained. From this the order of convergence could be calculated using the 2-logarithm because $\alpha = 1/2$ as describes in section 2.5.8:

$$p = \log_2(r).$$

The seven obtained results for the nine step sizes is shown in figure 3.4 and it is clear that the method is converging towards $p = 1$, even though some results are a bit off, probably due to small errors when finding the index for $x = 1.65$ m.

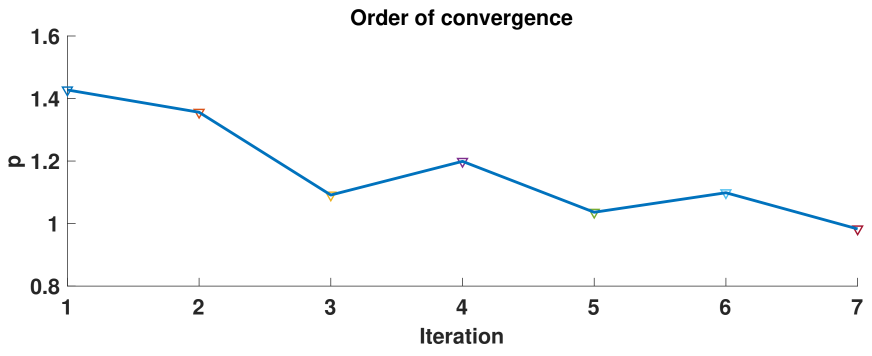


Figure 3.4: Order of convergence

4 Conclusion

In retrospect, the production of surfable waves with the shallow water equations and a finite volume method was not the hard part of the project, but the creation of the surfer model. In the end, the model described in this report did do well in simulations, and answered the questions asked in the introduction; a model of a surfer was successfully created and was made to move in a physically correct way on a shallow water surface - even though the model has several areas where it could be refined and should not be used for any high-precision applications as it is.

5 Discussion

In this section an evaluation of the project and recommended future work is discussed.

5.1 Evaluation

The results from this project can be used for entertaining simulations and rough analysis of the qualities of shallow water waves and surfing, but as it is now, it should not be used for any applications that need precision, at least not for the surfer.

5.1.1 Numerical Method

The numerical method used, The Lax Friedrichs Method, did work well in generating surfable shallow water waves. It is only a first order accurate model though. One could use a higher order model as the Lax-Wendroff scheme for the application, but that would introduce other problems, such as uncontrollable fluctuations according to the Godunov Theorem: *Any linear scheme for solving conservation laws with the property that it does not create new extrema is at most of order 1* [3].

5.1.2 Surfer

The resulting model created did do well for a case with a rather ”inactive” surfer as the physical properties connected with the created body were few. The three forces from the article [11] is the most basic setup for a floating body. In this project the ”paddling” was added (by setting an initial velocity) and also a kind of braking behavior was simulated by increasing the cross-section area in the drag force (3.3) if the surfer was speeding away from the wave, to mimic an actual surfer somewhat. Maybe a better use of this model would be to simulate the behaviour for a buoy or some similar passive body.

A model with a fixed coordinate system and buoyancy with included inertia was attempted, but the author was not able to make this model stable, even with a fine grid. It would be possible to replace the current way of implementing the holonomic constraint of always staying on the surface with an actual buoyancy according to Archimedes' principle, where the floating force depends on how much of the body is submerged.

5.2 Future Work

Several things could be done with the model in the future, if one would want to refine it. Here, the areas that the author believes could be improved the most are discussed.

5.2.1 Revision of chosen parameters

To make the model more realistic one could add more properties to the surfer. Such as adding a more advanced simulation of the ”paddling” as well as adding torque to the surfer. Also, all parameters that are rather arbitrarily chosen in the work here, such as the drag coefficient, drag-generating cross section area and the mass, should be revised and analyzed more closely than has been done here, to make the model trustworthy in any real calculations or applications.

5.2.2 Holonomic constraint

Another way of simulating a floating force that includes inertia could be implemented by using Archimedes' principle to make the buoyancy more realistic.

5.3 Final Words

Through this project a lot have been learned about mainly finite volume methods and the shallow water equations, subjects completely new to the author, but also a lot about MATLAB programming general techniques in numerical analysis. Special thanks are directed towards the supervisor Johan Wärnegård for the assistance he has provided with source material and in the creation of the surfer model.

References

[1] Garcia-Navarro, Pilar et al. “The shallow water equations and their application to realistic cases”. In: *Environmental Fluid Mechanics* 19.5 (2019), pp. 1235–1252.

[2] Henning, Partick. *Numerical solutions of differential equations - Boundary conditions for hyperbolic equations*. Stockholm: KTH Numerical Analysis, 2018.

[3] Henning, Partick. *Numerical solutions of differential equations - Finite Volume schemes of higher order*. Stockholm: KTH Numerical Analysis, 2018.

[4] Lax, Peter D. *Hyperbolic systems of conservation laws and the mathematical theory of shock waves*. SIAM, 1973.

[5] LeVeque, Randall J. “Boundary Conditions and Ghost Cells”. In: *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002, pp. 129–138. doi: 10.1017/CBO9780511791253.008.

[6] LeVeque, Randall J. “Conservation Laws and Differential Equations”. In: *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002, pp. 15–46. doi: 10.1017/CBO9780511791253.003.

[7] LeVeque, Randall J. “Finite Volume Methods”. In: *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002, pp. 64–86. doi: 10.1017/CBO9780511791253.005.

[8] LeVeque, Randall J. “Nonlinear Systems of Conservation Laws”. In: *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002, pp. 253–290. doi: 10.1017/CBO9780511791253.014.

[9] Michel-Dansac, Victor et al. “A well-balanced scheme for the shallow-water equations with topography or Manning friction”. In: *Journal of Computational Physics* 335 (2017), pp. 115–154. doi: 10.1016/j.jcp.2017.01.009. url: https://hal.archives-ouvertes.fr/hal-01247813.

[10] Runborg, Olof. *Verifying Numerical Convergence Rates*. Stockholm: KTH Computer Science, 2012.

[11] Sugimoto, Takeshi. “How to Ride a Wave: Mechanics of Surfing”. In: *SIAM Review* 40.2 (1998), pp. 341–343. issn: 00361445. url: http://www.jstor.org/stable/2653343.

[12] Vreugdenhil, C. B. “Shallow-water flows”. In: *Numerical Methods for Shallow-Water Flow*. Dordrecht: Springer Netherlands, 1994, pp. 1–14. isbn: 978-94-015-8354-1. doi: 10.1007/978-94-015-8354-1_1. url: https://doi.org/10.1007/978-94-015-8354-1_1.