LAPPEENRANTA-LAHTI UNIVERSITY OF TECHNOLOGY LUT
School of Engineering Science
Computational Engineering and Technical Physics
Technomathematics

Nosiba Elfadil

NUMERICAL APPROXIMATIONS OF NON-LINEAR SHALLOW WATER EQUATIONS

Master's Thesis

Examiners: Professor Heikki Haario
Dr. Ashvinkumar Chaudhari

Supervisors: Professor Heikki Haario
Dr. Ashvinkumar Chaudhari

ABSTRACT

LAPPEENRANTA-LAHTI UNIVERSITY OF TECHNOLOGY LUT
School of Engineering Science
Computational Engineering and Technical Physics
Technomathematics

Nosiba Elfadil

Numerical Approximations of Non-linear Shallow Water Equations

Master's Thesis

2020

63 pages, 14 figures, 3 table.

Examiners: Professor Heikki Haario
Dr. Ashvinkumar Chaudhari

Keywords: partial differential equation, numerical approximation, shallow water, finite difference methods, Crank-Nicholson approximation, Method of Lines.

In this work, the Finite Difference Methods (FDMs), and Method of Lines (MOL) have been used to solve the Shallow Water Equations (SWEs) in one dimension. The SWEs has a free surface with no rotation, it's kinematic viscosity is zero, it's pressure distribution is approximately hydrostatic, and it has a flat bottom topography with zero height. The SWEs were derived by using the equations of the conservation of mass and momentum to have a non-linear partial differential equations (PDEs). The PDE problem was tackled by applying a number of numerical models. To solve it numerically, the Backward Euler (BE), Forward Euler (FE), Crank-Nicolson (CN), and Method of Lines (MOL) approximations were used for the time and space discretizations. Once the set of non-linear PDEs has been converted to a system of algebraic equations using different schemes, the algebraic equations were solved with the help of MATLAB, and the numerical results were compared with the analytic solution. The analytical solution was obtained using the eigenvalues and eigenvectors for the quasi-linear form of the non-linear PDEs.

PREFACE

I would firstly like to thank God for helping me complete this thesis successfully.

I express my deepest gratitude to my supervisors Professor Heikki Haario and Dr. Ashvinkumar Chaudhari for their useful comments, remarks, and engagement through the learning process of this master thesis. I am also grateful to my parents, my sisters, and my fiance for moral support and encouragement to complete this research. There are no suitable words of thanks to say to all those who helped me, all I could sincerely say **thank you**.

Lappeenranta, May 27, 2020

Nosiba Elfadil

CONTENTS

LIST OF ABBREVIATIONS

2D	Two-dimensional
BE	Backward Euler scheme
CN	Crank-Nicolson
FE	Forward Euler scheme
FDM	Finite Difference Method
MOL	Method of Line
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
SWE	Shallow Water Equation

1 INTRODUCTION

1.1 Background

Saint-Venant (1871) proposed a shallow water system to model the flow in a channel. These equations are a two-dimensional (2D), time-dependent, non-linear partial differential hyperbolic equation system [1]. In recent decades, shallow water equations have been universally used to model physical water flow phenomena, as well as hydraulic engineering, such as flood waves, dam-breaks, tidal flows in the estuary, bore wave propagation in rivers, and so on.

There is little hope in real situations (realistic geometry, sharp spatiality) of explicitly solving shallow water equations, i.e. of finding an analytical formula for solutions. But there are steady-state solutions for the shallow water system. So it is advantageous to develop numerical approximations for this steady-state, which conserve these exact solutions, then, therefore, use these numerical methods for the calculation of approximate solutions of shallow water systems in real situations. [2].

Numerical analysis is an approach to the study of algorithms using numerical approximation to analyze mathematical problems. Naturally, applications in all fields of engineering and physical science use numerical analysis, but in the 21^{st} century, numerical methods also have a place in life and social sciences, medicine, business, and even in the arts. Numerical analysis is a branch of mathematics and computer science that generates, analyzes, and implements algorithms. Power growth and the computer revolution have increased the use of realistic mathematical models for both science and engineering. The complex numerical analysis is needed to find approximate solutions for the more complicated real-life models [3].

Numerical methods usually depend on hand interpolation in large printed tables before the advent of modern computers. Computers have been computing the required functions since the mid-20th century. However, these same formulas of interpolation continue to be used as part of the algorithms of the software to solve differential equations [4]. Modern numerical analysis does not seek accurate answers, as it is often impossible to get accurate answers in practice. Instead, many numerical analyzes are concerned with the achievement of approximate solutions while maintaining reasonable error limits [5].

1.2 Objectives and delimitations

Since the aim of the work is to find the numerical approximation of the non-linear Shallow-Water equations, the objectives for this work are

1. To derive the system of non-linear shallow water equation under specific assumptions.
2. To find the analytical solution for the shallow water equations.
3. To employ Finite Difference Method (FDM) and the Method of Lines (MOL) techniques based on a number of numerical schemes and to build a numerical model for solving the Shallow-Water model.
4. To compare the numerical results with the analytical one, and to test the accuracy of each numerical scheme employed.

1.3 Related work

For a long time, scientists have been trying to find a numerical solution for models of shallow water flow of different dimensions under different assumptions, e.g. flat or non-flat bottom topography, or the presence or absence of vortices (rotation). This section presents some literature reviews from the researchers who have tried to solve the system of shallow water equations under different assumptions and using different numerical methods.

In the work of A.I.Delis, and Th.Katsaounis [6], a FDM approach has been studied and generalized as a numerical method for the solution of two-dimensional (2D) shallow water system. [6] The previous non-oscillatory relaxation scheme has been revised and extended to include two-dimensional problems with source terms, resulting in the class of first and second-order TVD (Total Variance Diminishing) methods at the discretion of space and time. Approaches are based on classical relaxation models coupled with the Runge–Kutta time step mechanisms. Numerical tests are carried out with and without the source term for multiple test problems. The benchmark tests had shown that, in good alignment with well reported observed events, the schemes offered appropriate solutions.

A finite volume method for finding a numerical approximation for shallow water equations for either flat or non-flat bottom topography in one dimension (1D) has been presented in [7]. The technique is easy and accurate and prevents the solution of Riemann issues during the time integration phase. The predictor phase used the characteristic technique to reconstruct the numerical fluxes, while the corrector phase recovered the conservation equations. The proposed technique of finite-volume features has been tested under various flow regimens for shallow-water equation schemes. The findings collected in smooth regions are very accurate and provided excellent shock resolution without any non-physical oscillations near the shock fields.

The effective numerical model for the two-dimensional (2D) shallow water equation was studied in [8]. The model also used a finite-volume method, with a first-order central (FORCE) scheme, a surface-gradient method (SGM) for space, and a third-order Runge-Kutta method for time discretization. Overall, the findings showed excellent agreement with the observation that the system was capable of capturing shock and representing the soft symmetrical areas of the flow domain. The model also demonstrates acceptable results in the modeling of bi-directional flow conditions.

In the work of M.T.Capilla, and A.Balaguer-Beser [2], they developed a high-order well-balanced central schemes to solve the two dimensions shallow water model with flat bottom topography. A Runge–Kutta scheme used for time discretization. A Gaussian quadrature rule using third-degree polynomial to calculate point value from cell averages is employed to evaluate time integrals. The author builds a scheme to solve the problem of a trans-critical flow over a flat bottom. In order to prove the accuracy of the numerical results, a minor disturbance of the rest of the lake is resolved as a study. It has been shown that the numerical scheme is capable of reproducing the evolution of the variables, and that there is a strong approximation between the results and the observations recorded in the literature.

In the work of A.Emad et al. [9], they used a relatively new semi-analytical methodology, using a reduced-differential transformation method to obtain highly accurate solutions of shallow water equations in one dimension (1D), and the solutions were obtained in the form of an easily computable converging power sequence. The reduced-differential method of transformation is simple to implement, reduces the size of the computation, and generates an estimated solution without discretization. The findings show that the reduced method of converting differentials is more accurate and effective than other existing methods.

A finite-element discretization of a non-linear shallow water model for rotating spheres has been established in [10], in which consistent upwind stabilization is integrated into the system. The authors have introduced higher-order upwind advection schemes, maintaining compliance with the law on PV conservation and PV advection. By applying the model, both large-scale controlled and chaotic, turbulent flows were able to demonstrate to standard test cases that this model predicted a second-order convergence rate and could produce the required features.

1.4 Structure of the thesis

To achieve the goal of the thesis, the rest of it is organized as follows. Chapter 2 is dedicated to the derivation of a one-dimensional, non-linear, shallow equation under certain specific assumptions, and from the conservation of mass and momentum, after which an analytical solution for the shallow water system is derived and presented.

In chapter 3, an introduction to the finite difference methods will provide, and then, using different numerical methods, the schemes are designed in one dimension. The Backward Euler (BE), Forward Euler (FE), and Crank Nicolson (CN) schemes are used. In addition, the Method of Lines (MOL) technique is also used. Then from these schemes, the respective system of algebraic equations is obtained which is easy to solve.

In chapter 4, there will be a comparison for the numerical results, i.e. the results for the BE, FE, CN, and MOL schemes, with the analytical solution to see the performance of each of them. The results of the numerical methods will also be compared with the analytical ones in terms of accuracy and CPU time. Finally, in chapter 5 the conclusions are given.

2 Shallow Water Model

The movement of water over the surface is a fundamental physical phenomenon of practical interest to scientists and engineers. A number of factors, including ocean tides, dam breaks, floods, and tsunamis, reflect a strong concern from a variety of fields. Given the day-to-day familiarity of the flow of water, the governing equations for the flow consist of an intractable set of equations that is too complex for most applications to be applied in practice. As a result, rough flow descriptions are also used for different applications [11].

The approximation of common use in many applications results in a system of shallow water, a system of non-linear partial differential equations. Although simplified from the full regulatory equations, the shallow water model can rarely be solved analytically in general and still presents numerical calculations with many difficulties. In addition, many specific implementations present additional complexity, such as variable topography and emerging dry domain regions.

In this chapter, the derivation of shallow water equations using a number of assumptions will be presented [12], and also an analytical solution for this system of shallow water equations will provide.

2.1 Derivation of non-linear shallow water equations

Consider an incompressible fluid (water) of depth $h(x,t)$ over a bottom topography of depth $h_b(x,t)$ moving at a velocity $u(x,t)$ which is a function of the space and time, as indicated in Figure 1

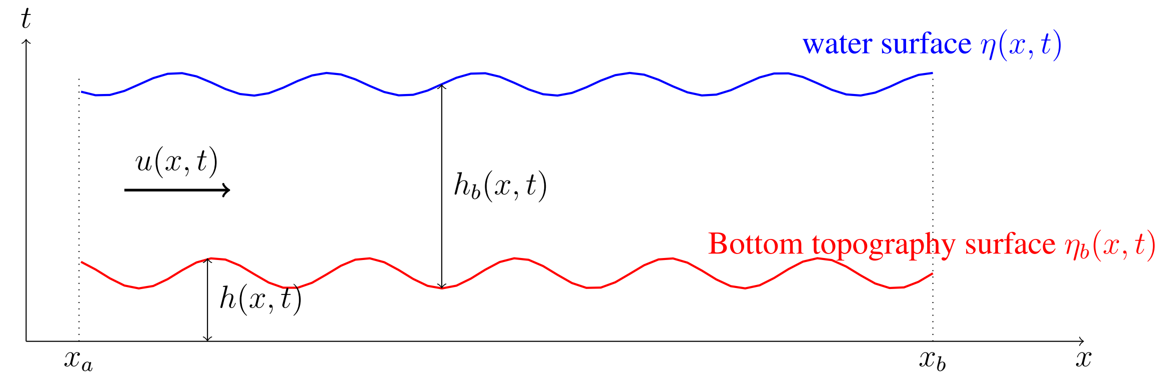

Figure 1. Definition sketch for derivation of the shallow water equations in one dimension.

In this thesis, The shallow water equations will be considered only on one dimension (1D). These equations are derived from the principles of conservation of mass equations (continuity equation in fluid dynamics) and conservation of momentum equations (Navier–Stokes equations) under certain specific assumptions.

2.1.1 The model assumptions

In order to derive the system of the shallow water equation, we need to know what kind of assumptions we are working on, so in this work, the assumptions are as follows

1. Horizontal length scale $>>$ vertical length scale (the main assumption in the shallow water equations).
2. One dimensional flow, i.e the velocity and the height are the functions of time t and position x only (no velocity in y and z directions).
3. No rotation (Coriolis force is zero, and the voracity vanishes).
4. Kinematic viscosity equals to zero (ν = 0).
5. Pressure distribution is approximately hydro-static ($p = \rho g h$).
6. The bottom topography surface is flat and the height is equal to zero ($h_b = 0$).
7. Free surface condition.

Next, the derivation of the shallow water model using the laws of mass and momentum conservation is shown. The derivation finally leads to a system of nonlinear partial differential equations.

2.1.2 Conservation of mass

The continuity equation in fluid dynamics states that the rate at which the mass enters the system is equal to the rate at which the mass leaves the system plus the mass accumulation inside the system. The equation that describes this conservation of the mass is then given by the

$$\frac{\partial \rho}{\partial t} + \nabla.(\rho \vec{U}) = 0 \tag{1}$$

where ρ is the density of the fluid, $\vec{U}$ is the flow velocity factor in 3D, and t is time. [13] [14].

Sine in this work, the fluid is water, and it's an incompressible fluid, i.e. the density is constant, then

$$\frac{\partial \rho}{\partial t} = 0.$$

By substitute this in equation (1), the first term will be canceled, so

$$\nabla.(\rho \vec{U}) = 0 \quad \text{or} \quad \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0. \tag{2}$$

Now by integrate equation (2) vertically from bottom topography wave $\eta_b(x,t)$ to the water wave $\eta(x,t)$ with respect to z

$$\int_{\eta_b}^{\eta} (\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z}).dz = 0$$

$$\int_{\eta_b}^{\eta} \frac{\partial u}{\partial x}.dz + \int_{\eta_b}^{\eta} \frac{\partial v}{\partial y}.dz + \int_{\eta_b}^{\eta} \frac{\partial w}{\partial z}.dz = 0.$$

In the last term, the derivation will cancel the integration, since both are related to the same variable z, so the remaining are

$$\int_{\eta_b}^{\eta} \frac{\partial u}{\partial x}.dz + \int_{\eta_b}^{\eta} \frac{\partial v}{\partial y}.dz + w|_{\eta_b}^{\eta} = 0. \tag{3}$$

In order to evaluate the two remaining integrals in equation (3), the following theorem will be needed.

Theorem 1 (Leibniz Theorem) *states that for an integral of the form [15]*

$$\int_{a(y,t)}^{b(y,t)} f(x,y,t).dx, \qquad -\infty < a(y,t), b(y,t) < \infty$$

can be express as

$$\frac{\partial}{\partial t}\left(\int_{a(y,t)}^{b(y,t)} f(x,y,t).dx\right) = \int_{a(y,t)}^{b(y,t)} \frac{\partial f}{\partial t}.dx - f(a,y,t)\frac{\partial a}{\partial t} + f(b,y,t)\frac{\partial b}{\partial t}. \tag{4}$$

Now from Leibniz theorem

$$\begin{aligned}
\int_{\eta_b}^{\eta} \frac{\partial u}{\partial x}.dz =& \frac{\partial}{\partial x}\int_{\eta_b}^{\eta} u.dz - u|_{\eta}\frac{\partial \eta}{\partial x} + u|_{\eta_b}\frac{\partial \eta_b}{\partial x}\\
\int_{\eta_b}^{\eta} \frac{\partial v}{\partial x}.dz =& \frac{\partial}{\partial x}\int_{\eta_b}^{\eta} v.dz - v|_{\eta}\frac{\partial \eta}{\partial x} + v|_{\eta_b}\frac{\partial \eta_b}{\partial x}.
\end{aligned}$$

Then from the above equations, equation (3) can be written as

$$\frac{\partial}{\partial x}\int_{\eta_b}^{\eta} u.dz - u|_{\eta}\frac{\partial \eta}{\partial x} + u|_{\eta_b}\frac{\partial \eta_b}{\partial x} + \frac{\partial}{\partial x}\int_{\eta_b}^{\eta} v.dz - v|_{\eta}\frac{\partial \eta}{\partial x} + v|_{\eta_b}\frac{\partial \eta_b}{\partial x} + w|_{\eta} - w|_{\eta_b} = 0.$$

From sixth condition, which is saying that the bottom topography is flat and the height is zero ($\eta_b = 0$), then $\frac{\partial \eta_b}{\partial x} = 0$

$$\frac{\partial}{\partial x}\int_{\eta_b}^{\eta} u.dz - u|_{\eta}\frac{\partial \eta}{\partial x} + \frac{\partial}{\partial x}\int_{\eta_b}^{\eta} v.dz - v|_{\eta}\frac{\partial \eta}{\partial x} + w|_{\eta} = 0.$$

And from the second condition, one dimensional flow, the velocity v in the direction of y is zero

$$\frac{\partial}{\partial x}\int_{\eta_b}^{\eta} u.dz - u|_{\eta}\frac{\partial \eta}{\partial x} + w|_{\eta} = 0. \tag{5}$$

Now from the last assumption of the free surface condition, the velocity w in z direction is

$$\begin{aligned} w =& \frac{\partial \eta}{\partial t} + \vec{U}.\nabla \eta \\ =& \frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y}. \end{aligned}$$

By substitute w in equation (5)

$$\begin{aligned} \frac{\partial}{\partial x}\int_{\eta_b}^{\eta} u.dz - u|_{\eta}\frac{\partial \eta}{\partial x} + \frac{\partial \eta}{\partial t} + u|_{\eta}\frac{\partial \eta}{\partial x} + v|\eta\frac{\partial \eta}{\partial y} = 0 \\ \frac{\partial}{\partial x}\int_{\eta_b}^{\eta} u.dz + \frac{\partial \eta}{\partial t} + v|\eta\frac{\partial \eta}{\partial y} = 0. \end{aligned}$$

And again from one dimensional assumption, the velocity in y direction is equals to zero ($v = 0$), so

$$\frac{\partial}{\partial x}\int_{\eta_b}^{\eta} u.dz + \frac{\partial \eta}{\partial t} = 0. \tag{6}$$

Since the velocity u is a function of x and t, and the integration is not depend on x and t, then equation (6) will be

$$\frac{\partial}{\partial x}(u(\eta - \eta_b)) + \frac{\partial \eta}{\partial t} = 0.$$

But we know $\eta - \eta_b = \eta = h$, then

$$\frac{\partial}{\partial x}(uh) + \frac{\partial h}{\partial t} = 0.$$

Or we can rewrite it as

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) = 0. \tag{7}$$

which is the first equation in the system of the shallow water equation.

2.1.3 Conservation of momentum

Conservation of momentum or Navier Stokes equation, it is an equation which describes the motion of the fluid and it is given by

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + f \tag{8}$$

where u, v, w are velocity in x, y, z directions respectively, p is pressure, ρ is density of the water, ν is kinematic viscosity, f is the sum of all forces acting on a fluid.

Now by applying the first assumption, which is we working in one dimension, then the velocity in y, z directions are zeros ($v, w = 0$), equation (8) can be rewritten as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -\frac{1}{\rho}\frac{\partial p}{\partial x} + \nu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2}\right) + f.$$

And, based on the assumption of the viscosity of the fluid is equal to zero ($\nu = 0$),then

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -\frac{\partial p}{\partial x}\frac{1}{\rho} + f. \tag{9}$$

The pressure distribution is approximately hydro-static ($p = \rho g h$), so

$$\frac{\partial p}{\partial x} = \rho g\frac{\partial h}{\partial x} \Longrightarrow \frac{1}{\rho}\frac{\partial p}{\partial x} = g\frac{\partial h}{\partial x} \tag{10}$$

where g is the acceleration of the gravity.

Now by combining equation (9) and (10)

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial h}{\partial x} + f. \tag{11}$$

As we define f as the sum of all the forces acting on the fluid, which is the weight, the friction, and the Coriolis forces. However, in the x-direction, the weight that is due to the gravity has a zero component, and since there is no rotation (second assumption), the Coriolis force is also equal to zero, f is therefore equal to the friction force and it is given by

$$f = -g\frac{\partial h_b}{\partial x}. \tag{12}$$

Then substitute equation (12) into equation (11), will get

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial h}{\partial x} - g\frac{\partial h_b}{\partial x}.$$

Now from the sixth assumption, the bottom surface is flat, then $\frac{\partial h_b}{\partial x} = 0$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = -g\frac{\partial h}{\partial x}. \tag{13}$$

Since the gravity g is constant, then it can move inside the derivation, so equation (13) can be rewritten as

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial}{\partial x}(gh) = 0. \tag{14}$$

Multiply equation (14) by h

$$h\frac{\partial u}{\partial t} + (hu)\frac{\partial u}{\partial x} + h\frac{\partial}{\partial x}(gh) = 0. \tag{15}$$

By adding and subtracting $u\frac{\partial}{\partial x}(hu)$ in equation (15)

$$h\frac{\partial u}{\partial t} - u\frac{\partial}{\partial x}(hu) + u\frac{\partial}{\partial x}(hu) + (hu)\frac{\partial u}{\partial x} + h\frac{\partial}{\partial x}(gh) = 0 \tag{16}$$

Now we know that

$$\begin{aligned} u\frac{\partial}{\partial x}(hu) + (hu)\frac{\partial u}{\partial x} &= \frac{\partial}{\partial x}(hu^2) \\ \text{and} \qquad h\frac{\partial}{\partial x}(gh) &= \frac{\partial}{\partial x}(\frac{1}{2}gh^2) \end{aligned}$$

and from equation (7), we have

$$\frac{\partial}{\partial x}(hu) = -\frac{\partial h}{\partial t} \Longrightarrow u\frac{\partial}{\partial x}(hu) = -u\frac{\partial h}{\partial t}$$

Now by substitute all this in equation (16), will get

$$h\frac{\partial u}{\partial t} + u\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu^2) + \frac{\partial}{\partial x}(\frac{1}{2}gh^2) = 0 \tag{17}$$

Since

$$h\frac{\partial u}{\partial t} + u\frac{\partial h}{\partial t} = \frac{\partial}{\partial t}(hu)$$

Then equation (17) will be

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}(hu^2 + \frac{1}{2}gh^2) = 0 \tag{18}$$

And which is the second equation in the system of the shallow water equation.

Then the system of the shallow water equations in one dimension, and under specific assumptions (shown above) are

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) = 0 \tag{19}$$

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}(hk + \frac{1}{2}gh^2) = 0 \tag{20}$$

Where $k = u^2$ is the kinetic energy per unit mass [16].

Or the above system can be written in the matrix form as

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ hu \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{21}$$

with initial and boundary conditions are [17]

$$\begin{aligned} \text{Initial} \quad & u(x,0) = 0 \;\text{ for all } x \in [x_a, x_b] \\ & h(x,0) = 1 + \frac{2}{5}e^{-5x^2} \;\text{ for all } x \in [x_a, x_b] \\ \text{Boundary} \quad & u(x_a,t) = u(x_b,t) = 0 \\ & h(x_a,t) = h(x_b,t) = 1 \end{aligned}$$

The unit of the velocity u is $meter/second$, and for the height h is $meter$ and for the time t is $second$, and space x is $meter$.

Now after founding the non-linear shallow water equations, which are a system of partial differential equations (PDEs), the analytical solution will be shown in the next section.

2.2 Analytical solution for nonlinear shallow water model

An analytical solution for the non-linear shallow water equation in one dimension and on the specific assumptions (provided above) will be shown in this section. First of all, let us recall the shallow water system

$$\frac{\partial}{\partial t}\begin{bmatrix} h \\ hu \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \tag{22}$$

Let us define

$$q(x,t) = \begin{bmatrix} h \\ hu \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \tag{23}$$

and

$$f(q) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} q_2 \\ \frac{q_2^2}{q_1} + \frac{1}{2}gq_1^2 \end{bmatrix}. \tag{24}$$

Now for a smooth solution, the quasi-linear form for equations (22) can written as

$$q_t + f'(q)q_x = 0 \tag{25}$$

where $f'(q)$ is the Jacobin matrix of $f(q)$, and it's equal to

$$f'(q) = \begin{bmatrix} 0 & 1 \\ -\left(\frac{q_2}{q_1}\right)^2 + gq_1 & 2\frac{q_2}{q_1} \end{bmatrix}. \tag{26}$$

The solution of equations (22) can be given by finding the eigenvalues and eigenvectors for the Jacobin matrix. So

$$|f'(q) - \lambda| = 0 \quad \rightarrow \quad \begin{vmatrix} -\lambda & 1 \\ -\left(\frac{q_2}{q_1}\right)^2 + gq_1 & 2\frac{q_2}{q_1} - \lambda \end{vmatrix} = 0.$$

And the characteristic equation of λ is

$$-\lambda\left(2\frac{q_2}{q_1} - \lambda\right) + \left(\frac{q_2}{q_1}\right)^2 - gq_1 = 0$$
$$\lambda^2 - \left(2\frac{q_2}{q_1}\right)\lambda + \left(\frac{q_2}{q_1}\right)^2 - gq_1 = 0$$

which is a quadratic equation, so the solution is

$$\lambda_{1,2} = \frac{\frac{2q_2}{q_1} \pm \sqrt{\left(\frac{2q_2}{q_1}\right)^2 - 4\left(\left(\frac{q_2}{q_1}\right)^2 - gq_1\right)}}{2} = \frac{q_2}{q_1} \pm \sqrt{gq_1}.$$

The corresponding eigenvectors it can be given by the equation

$$(f'(q) - \lambda)r = 0. \tag{27}$$

For $\lambda_1 = \frac{q_2}{q_1} + \sqrt{gq_1}$, the corresponding eigenvector is

$$\begin{bmatrix} -\frac{q_2}{q_1} - \sqrt{gq_1} & 1 \\ -\left(\frac{q_2}{q_1}\right)^2 + gq_1 & 2\frac{q_2}{q_1} - \frac{q_2}{q_1} - \sqrt{gq_1} \end{bmatrix} \begin{bmatrix} r_{11} \\ r_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

Then the system of equations is going to be like

$$\begin{aligned} \left(-\frac{q_2}{q_1} - \sqrt{gq_1}\right) r_{11} + r_{12} = 0 \\ \left(-\left(\frac{q_2}{q_1}\right)^2 + gq_1\right) r_{11} + \left(\frac{q_2}{q_1} - \sqrt{gq_1}\right) r_{12} = 0. \end{aligned}$$

Now from the first equation, we have

$$r_{12} = \left(\frac{q_2}{q_1} + \sqrt{gq_1}\right) r_{11}.$$

If r_{11} is assumed to be equal to one, the above equation will be

$$r_{12} = \left(\frac{q_2}{q_1} + \sqrt{gq_1}\right).$$

Now the corresponding eigenvector for λ_1 is

$$r_1 = \begin{bmatrix} 1 \\ \frac{q_2}{q_1} + \sqrt{gq_1} \end{bmatrix}.$$

For $\lambda_2 = \frac{q_2}{q_1} - \sqrt{gq_1}$, the corresponding eigenvector is

$$\begin{bmatrix} -\frac{q_2}{q_1} + \sqrt{gq_1} & 1 \\ -\left(\frac{q_2}{q_1}\right)^2 + gq_1 & 2\frac{q_2}{q_1} - \frac{q_2}{q_1} + \sqrt{gq_1} \end{bmatrix} \begin{bmatrix} r_{21} \\ r_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}.$$

So from the matrix above, the equations are

$$\left(-\frac{q_2}{q_1}+\sqrt{gq_1}\right)r_{21}+r_{22}=0$$
$$\left(-\left(\frac{q_2}{q_1}\right)^2+gq_1\right)r_{21}+\left(\frac{q_2}{q_1}+\sqrt{gq_1}\right)r_{22}=0.$$

The first equation could be written as

$$r_{22}=\left(\frac{q_2}{q_1}-\sqrt{gq_1}\right)r_{21}.$$

Also if we assume that $r_{21}=1$, then

$$r_{22}=\left(\frac{q_2}{q_1}+\sqrt{gq_1}\right).$$

So the corresponding eigenvector for λ_2 is

$$r_2=\begin{bmatrix}1\\ \frac{q_2}{q_1}-\sqrt{gq_1}\end{bmatrix}.$$

Then the eigenvalues and the corresponding eigenvectors can be written in terms of h and u as

$$\lambda_1=u+\sqrt{gh}\quad\rightarrow\quad r_1=\begin{bmatrix}1\\ u+\sqrt{gh}\end{bmatrix}$$
$$\lambda_2=u-\sqrt{gh}\quad\rightarrow\quad r_2=\begin{bmatrix}1\\ u-\sqrt{gh}\end{bmatrix}.$$

Now the solution for the equation (25) is in the form of [17]

$$q(x,t)=\bar{w_1}(x+\sqrt{gh_0}t)r_1+\bar{w_2}(x-\sqrt{gh_0}t)r_2 \tag{28}$$

where $\bar{w_1}$ and $\bar{w_2}$ are scalar functions. So to find this unknown functions, the initial values q_0 for the height and the mass velocity will be used

$$q_0=\bar{w_1}(x)r_1+\bar{w_2}(x)r_2.$$

Let put $R=[r_1|r_2]$, which is 2×2 matrix for the both eigenvectors, then the solution at the initial time is

$$q_0=R\bar{w}(x)$$

where $\bar{w}$ is a vector contains $\bar{w_1}$ and $\bar{w_2}$. The values of this vector $\bar{w}$ can now be given by

$$\begin{aligned}\bar{w}(x) = R^{-1}q_0 &= \begin{bmatrix} 1 & 1 \\ u_0 + \sqrt{gh_0} & u_0 - \sqrt{gh_0} \end{bmatrix}^{-1} \begin{bmatrix} h_0 \\ h_0 u_0 \end{bmatrix} \\ &= -\frac{1}{2\sqrt{gh_0}} \begin{bmatrix} u_0 - \sqrt{gh_0} & -1 \\ -(u_0 + \sqrt{gh_0}) & 1 \end{bmatrix} \begin{bmatrix} h_0 \\ h_0 u_0 \end{bmatrix}.\end{aligned}$$

Now the first scalar function $\bar{w_1}$ is

$$\begin{aligned}\bar{w_1}(x) &= -\frac{1}{2\sqrt{gh_0}} \left((u_0 - \sqrt{gh_0})h_0 - h_0 u_0 \right) \\ &= -\frac{1}{2\sqrt{gh_0}} \left(-\sqrt{gh_0} h_0 \right) = \frac{h_0}{2}.\end{aligned}$$

And the second scalar function $\bar{w_2}$ is

$$\begin{aligned}\bar{w_2}(x) &= -\frac{1}{2\sqrt{gh_0}} \left(-(u_0 + \sqrt{gh_0})h_0 + h_0 u_0 \right) \\ &= -\frac{1}{2\sqrt{gh_0}} \left(-\sqrt{gh_0} h_0 \right) = \frac{h_0}{2}.\end{aligned}$$

So by substituting the values for both scalar functions on equation (28), will have

$$q(x,t) = \frac{1}{2} h_0(x + \sqrt{gh_0}t) r_1 + \frac{1}{2} h_0(x - \sqrt{gh_0}t) r_2.$$

Now the solution for the nonlinear shallow water model in terms of the height of the water and the mass velocity is

$$\begin{aligned}h(x,t) &= \frac{1}{2} h_0(x + \sqrt{gh_0}t) + \frac{1}{2} h_0(x - \sqrt{gh_0}t) \\ h(x,t)u(x,t) &= \frac{1}{2} h_0(x + \sqrt{gh_0}t)(u_0 + \sqrt{gh_0}) + \frac{1}{2} h_0(x - \sqrt{gh_0}t)(u_0 - \sqrt{gh_0})\end{aligned} \tag{29}$$

where h_0 and u_0 is the initial values for the height and the mass velocity at time zero.

Once the analytical solution for a shallow-water model has been solved, numerical approximations will be found to solve this system using different numerical methods in the next chapter.

3 Numerical Methods for Shallow Water Model

For some models, a numerical approximation is used because the analytical one is not easy to find or does not even exist for some time. In this work, since the model is a simple version of nonlinear shallow water equations (no rotation, viscosity is vanished, ...etc), the numerical approximation is adequate for the purpose of evaluating numerical methods in such a way to make the work easier for the two dimensions (2D) model. The Finite difference methods [12], the Crank-Nicolson method, and the Method of Lines will be used in this work.

3.1 Finite Difference Methods

Finite Difference Methods (FDMs) are a type of numerical methods used to solve differential equations by approximating them with equations of differentials. FDMs transform a linear (non-linear) Ordinary Differential Equations (ODEs)/Partial Differential Equations (PDEs) into a linear (non-linear) equation system that can then be solved using algebra matrix techniques [18].

Before implementing FDM on the system of shallow water, a small introduction for the FDM derivation will be introduced.

3.1.1 Derivation of finite difference methods

FDMs have three approximations, forward, backward, and central approximations, so let's assume that to derive it, f is a function whose derivatives are defined, if $x_0$belongs to the domain of f, and Δx is the uniform step size (mesh size), then Taylor's theorem can be used to create a Taylor series expansion as [19]

$$f(x_0 + \Delta x) = f(x_0) + \Delta x f'(x_0) + \frac{(\Delta x)^2}{2!} f''(x_0) + \frac{(\Delta x)^3}{3!} f'''(x_0) + ... \qquad (30)$$

and

$$f(x_0 - \Delta x) = f(x_0) - \Delta x f'(x_0) + \frac{(\Delta x)^2}{2!} f''(x_0) - \frac{(\Delta x)^3}{3!} f'''(x_0) + ... \qquad (31)$$

Then the general forms for the three approximations, forward, backward, and central are

Forward Euler approximation: To find the value of the first derivative of function f at the point x_n, the value of the function at point x_{n+1} will be needed. Figure 2 show the strategy of forward approximation.

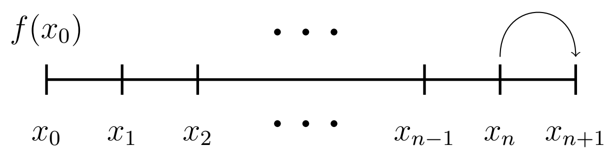

Figure 2. Graph shown the strategy of forward approximation

From equation (30), by taking $x_0 = x_n$ and since here the first derivative is needed, then the terms with power in Δx greater than or equal 2 will be neglected, so

$$f(x_n + \Delta x) \approx f(x_n) + \Delta x f'(x_n).$$

Then forward approximation of the first derivative of f is

$$f'(x_n) \approx \frac{f(x_n + \Delta x) - f(x_n)}{\Delta x}. \tag{32}$$

Backward Euler approximation: To find the value of the first derivative of function f at the point x_n, the value of the function at point x_{n-1} will be needed. The strategy of this method as shown in Figure 3.

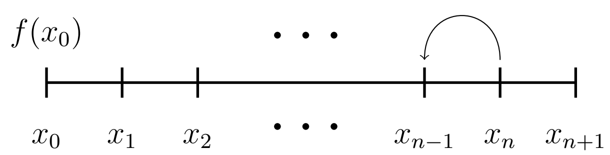

Figure 3. Graph shown the strategy of backward approximation

From equation (31), and also by ignore the terms with power in Δx grater than or equal 2, will get

$$f(x_n - \Delta x) \approx f(x_n) - \Delta x f'(x_n).$$

Then backward approximation for the first derivative of f is

$$f'(x_n) \approx \frac{f(x_n) - f(x_n - \Delta x)}{\Delta x}. \tag{33}$$

Central approximation: To find the value of the first derivative of function f at point x_n, the values of the function at points x_{n-1} and x_{n+1} will be needed as shown in Figure 4.

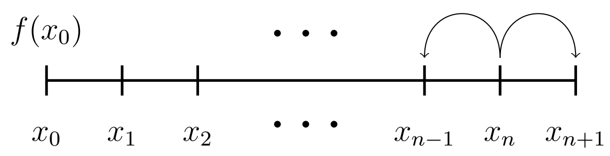

Figure 4. Graph shown the strategy of central approximation

From Figure 4, the central approximation is the sum of the backward and forward approximations divided by 2, then from equation (32) and (33)

$$\begin{aligned} f'(x_n) &\approx \frac{1}{2}\left(\frac{f(x_n+\Delta x)-f(x_n)}{\Delta x}+\frac{f(x_n)-f(x_n-\Delta x)}{\Delta x}\right) \\ &\approx \frac{1}{2}\left(\frac{f(x_n+\Delta x)-f(x_n)+f(x_n)-f(x_n-\Delta x)}{\Delta x}\right) \\ &\approx \frac{f(x_n+\Delta x)-f(x_n-\Delta x)}{2\Delta x}. \end{aligned}$$

Then the central approximation for the first derivative of f is

$$f'(x_n) \approx \frac{f(x_n+\Delta x)-f(x_n-\Delta x)}{2\Delta x}. \tag{34}$$

Now different schemes to find a numerical approximation for the non-linear shallow water equations will be introduced.

3.1.2 Backward Euler scheme

Since the problem is to find a numerical approximation for non-rotational and non-linear shallow water equations in one dimension, and there are two equations that depend on each other, one of the numerical methods is to use backward time-stepping and central spatial discretion as shown in Figure5.

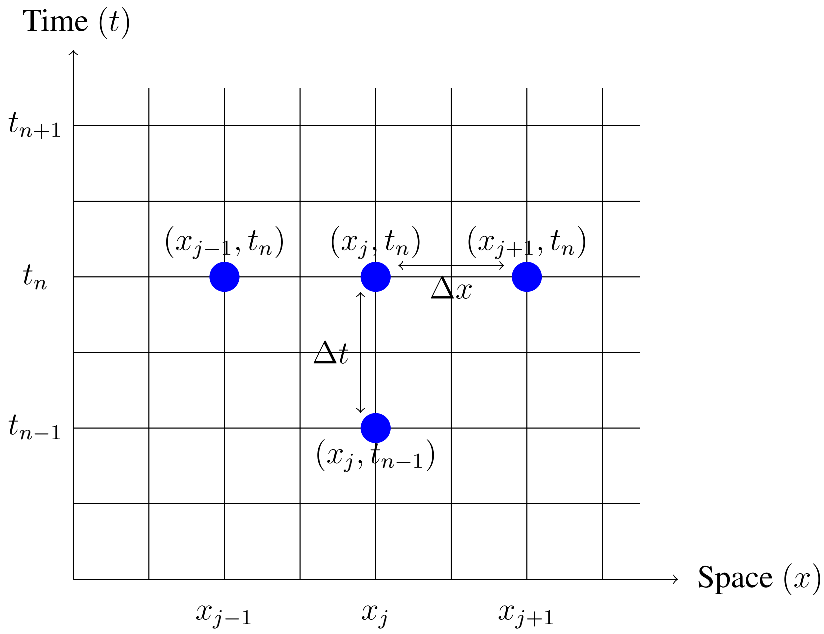

Figure 5. The mesh for backward Euler scheme

Recall the system of non-linear shallow water equation

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) = 0 \tag{35}$$

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}(hu^2 + \frac{1}{2}gh^2) = 0. \tag{36}$$

In equation (35), the second term is a multiplication of two functions, then it can rewrite as

$$\frac{\partial h}{\partial t} + h\frac{\partial u}{\partial x} + u\frac{\partial h}{\partial x} = 0. \tag{37}$$

And for equation (36), the first and the second terms are also multiplication for two functions, then

$$h\frac{\partial u}{\partial t} + u\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu^2) + \frac{\partial}{\partial x}(\frac{1}{2}gh^2) = 0$$
$$h\frac{\partial u}{\partial t} + u\frac{\partial h}{\partial t} + h\frac{\partial u^2}{\partial x} + u^2\frac{\partial h}{\partial x} + \frac{1}{2}g\frac{\partial h^2}{\partial x} = 0.$$

But we know that $\frac{\partial u^2}{\partial x} = 2u\frac{\partial u}{\partial x}$, and $\frac{\partial h^2}{\partial x} = 2h\frac{\partial h}{\partial x}$, then the above equation will be

$$h\frac{\partial u}{\partial t} + u\frac{\partial h}{\partial t} + 2hu\frac{\partial u}{\partial x} + u^2\frac{\partial h}{\partial x} + gh\frac{\partial h}{\partial x} = 0. \tag{38}$$

The discretization for equations (37) and (38) will be individually.

From equation (37), the backward, central approximations for time and space is

$$\frac{h_j^n - h_j^{n-1}}{\Delta t} + h_j^{n-1}\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + u_j^{n-1}\frac{h_{j+1}^n - h_{j-1}^n}{2\Delta x} = 0. \tag{39}$$

When we say $h_j^n \approx h(x_j, t_n)$, it's means the height in space x_j and time t_n, and for $u_j^n \approx u(x_j, t_n)$ is the speed in space x_j and time t_n, where $t_n = n\Delta t,\ \ x_j = j\Delta x$.

Same for equation (38), by applying backward approximation for time and central approximation for space, will get

$$h_j^{n-1}\frac{u_j^n - u_j^{n-1}}{\Delta t} + u_j^{n-1}\frac{h_j^n - h_j^{n-1}}{\Delta t} + 2h_j^{n-1}u_j^{n-1}\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + u_j^{n-1^2}\frac{h_{j+1}^n - h_{j-1}^n}{2\Delta x}$$
$$+gh_j^{n-1}\frac{h_{j+1}^n - h_{j-1}^n}{2\Delta x} = 0.$$

Or simply

$$h_j^{n-1}\frac{u_j^n - u_j^{n-1}}{\Delta t} + u_j^{n-1}\frac{h_j^n - h_j^{n-1}}{\Delta t} + h_j^{n-1}u_j^{n-1}\frac{u_{j+1}^n - u_{j-1}^n}{\Delta x} + u_j^{n-1^2}\frac{h_{j+1}^n - h_{j-1}^n}{2\Delta x}$$
$$+gh_j^{n-1}\frac{h_{j+1}^n - h_{j-1}^n}{2\Delta x} = 0. \tag{40}$$

Now equations (39), and (40) are a non-linear system of difference equations, then it can be transform to linear system by multiply equation (39) by u_j^{n-1}, so will get

$$u_j^{n-1}\frac{h_j^n - h_j^{n-1}}{\Delta t} + u_j^{n-1}h_j^{n-1}\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + u_j^{n-1^2}\frac{h_{j+1}^n - h_{j-1}^n}{2\Delta x} = 0. \tag{41}$$

Subtracting equation (41) from equation (40), will have

$$h_j^{n-1}\frac{u_j^n - u_j^{n-1}}{\Delta t} + h_j^{n-1}u_j^{n-1}\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + gh_j^{n-1}\frac{h_{j+1}^n - h_{j-1}^n}{2\Delta x} = 0. \tag{42}$$

Multiply equations (39) and (42) by $2\Delta t\Delta x$

$$2\Delta x(h_j^n - h_j^{n-1}) + \Delta t h_j^{n-1}(u_{j+1}^n - u_{j-1}^n) + \Delta t u_j^{n-1}(h_{j+1}^n - h_{j-1}^n) = 0$$
$$2\Delta x h_j^{n-1}(u_j^n - u_j^{n-1}) + \Delta t h_j^{n-1}u_j^{n-1}(u_{j+1}^n - u_{j-1}^n) + \Delta t g h_j^{n-1}(h_{j+1}^n - h_{j-1}^n) = 0.$$

Moving the known terms to right-side, then the equation for mass is

$$2\Delta x h_j^n + \Delta t h_j^{n-1}(u_{j+1}^n - u_{j-1}^n) + \Delta t u_j^{n-1}(h_{j+1}^n - h_{j-1}^n) = 2\Delta x h_j^{n-1} \tag{43}$$

and the equation for momentum is

$$2\Delta x h_j^{n-1} u_j^n + \Delta t h_j^{n-1} u_j^{n-1}(u_{j+1}^n - u_{j-1}^n) + \Delta t g h_j^{n-1}(h_{j+1}^n - h_{j-1}^n) = 2\Delta x h_j^{n-1} u_j^{n-1}. \tag{44}$$

Now equations (43) and (44) are backward Euler scheme, and it is implicit and one step scheme.

Then the boundary conditions for velocity $u(x_a, t) = u(x_b, t) = 0 \ meter/second$, and for height $h(x_a, t) = h(x_b, t) = 1 \ meter$, or $u_0^n = u_{N+2}^n = 0$ and $h_0^n = h_{N+2}^n = 1$, where $x_a = 0$, and $x_b = x_{N+2}$.

For $j = 1$, substitute it in equation (43)

$$2\Delta x h_1^n + \Delta t h_1^{n-1}(u_2^n - u_0^n) + \Delta t u_1^{n-1}(h_2^n - h_0^n) = 2\Delta x h_1^{n-1}$$
$$2\Delta x h_1^n + \Delta t h_1^{n-1} u_2^n + \Delta t u_1^{n-1}(h_2^n - 1) = 2\Delta x h_1^{n-1}.$$

Move the know terms in the right-side

$$2\Delta x h_1^n + \Delta t h_1^{n-1} u_2^n + \Delta t u_1^{n-1} h_2^n = 2\Delta x h_1^{n-1} + \Delta t u_1^{n-1} \tag{45}$$

and substitute $j = 1$ in equation (44)

$$2\Delta x h_1^{n-1} u_1^n + \Delta t h_1^{n-1} u_1^{n-1}(u_2^n - u_0^n) + \Delta t g h_1^{n-1}(h_2^n - h_0^n) = 2\Delta x h_1^{n-1} u_1^{n-1}$$
$$2\Delta x h_1^{n-1} u_1^n + \Delta t h_1^{n-1} u_1^{n-1} u_2^n + \Delta t g h_1^{n-1}(h_2^n - 1) = 2\Delta x h_1^{n-1} u_1^{n-1}.$$

Move the known terms in the right-side

$$2\Delta x h_1^{n-1} u_1^n + \Delta t h_1^{n-1} u_1^{n-1} u_2^n + \Delta t g h_1^{n-1} h_2^n = 2\Delta x h_1^{n-1} u_1^{n-1} + \Delta t g h_1^{n-1}. \tag{46}$$

For $j = 2$, in equation (43)

$$2\Delta x h_2^n + \Delta t h_2^{n-1}(u_3^n - u_1^n) + \Delta t u_2^{n-1}(h_3^n - h_1^n) = 2\Delta x h_2^{n-1} \tag{47}$$

and for equation (44)

$$2\Delta x h_2^{n-1} u_2^n + \Delta t h_2^{n-1} u_2^{n-1}(u_3^n - u_1^n) + \Delta t g h_2^{n-1}(h_3^n - h_1^n) = 2\Delta x h_2^{n-1} u_2^{n-1}. \tag{48}$$

For $j = 3$, in equation (43)

$$2\Delta x h_3^n + \Delta t h_3^{n-1}(u_4^n - u_2^n) + \Delta t u_3^{n-1}(h_4^n - h_2^n) = 2\Delta x h_3^{n-1}. \tag{49}$$

and for equation (44)

$$2\Delta x h_3^{n-1}u_3^n + \Delta t h_3^{n-1}u_3^{n-1}(u_4^n - u_2^n) + \Delta t g h_3^{n-1}(h_4^n - h_2^n) = 2\Delta x h_3^{n-1}u_3^{n-1}. \tag{50}$$

For $j = x_{N+1}$, in equation (43)

$$\begin{aligned} 2\Delta x h_{x_{N+1}}^n + \Delta t h_{x_{N+1}}^{n-1}(u_{x_{N+2}}^n - u_{x_N}^n) + \Delta t u_{x_{N+1}}^{n-1}(h_{x_{N+2}}^n - h_{x_N}^n) &= 2\Delta x h_{x_{N+1}}^{n-1} \\ 2\Delta x h_{x_{N+1}}^n + \Delta t h_{x_{N+1}}^{n-1}(0 - u_{x_N}^n) + \Delta t u_{x_{N+1}}^{n-1}(1 - h_{x_N}^n) &= 2\Delta x h_{x_{N+1}}^{n-1} \\ 2\Delta x h_{x_{N+1}}^n - \Delta t h_{x_{N+1}}^{n-1}u_{x_N}^n + \Delta t u_{x_{N+1}}^{n-1} - \Delta t u_{x_{N+1}}^{n-1}h_{x_N}^n &= 2\Delta x h_{x_{N+1}}^{n-1}. \end{aligned}$$

Moving the known terms ro the right-side

$$2\Delta x h_{x_{N+1}}^n - \Delta t h_{x_{N+1}}^{n-1}u_{x_N}^n - \Delta t u_{x_{N+1}}^{n-1}h_{x_N}^n = 2\Delta x h_{x_{N+1}}^{n-1} - \Delta t u_{x_{N+1}}^{n-1} \tag{51}$$

and equation (44)

$$\begin{aligned} 2\Delta x h_{x_{N+1}}^{n-1}u_{x_{N+1}}^n + \Delta t h_{x_{N+1}}^{n-1}u_{x_{N+1}}^{n-1}(u_{x_{N+2}}^n - u_{x_N}^n) + \Delta t g h_{x_{N+1}}^{n-1}(h_{x_{N+2}}^n - h_{x_N}^n) \\ = 2\Delta x h_{x_{N+1}}^{n-1}u_{x_{N+1}}^{n-1} \\ 2\Delta x h_{x_{N+1}}^{n-1}u_{x_{N+1}}^n + \Delta t h_{x_{N+1}}^{n-1}u_{x_{N+1}}^{n-1}(0 - u_{x_N}^n) + \Delta t g h_{x_{N+1}}^{n-1}(1 - h_{x_N}^n) = 2\Delta x h_{x_{N+1}}^{n-1}u_{x_{N+1}}^{n-1}. \end{aligned}$$

So the above equation will be

$$2\Delta x h_{x_{N+1}}^{n-1}u_{x_{N+1}}^n - \Delta t h_{x_{N+1}}^{n-1}u_{x_{N+1}}^{n-1}u_{x_N}^n + \Delta t g h_{x_{N+1}}^{n-1} - \Delta t g h_{x_{N+1}}^{n-1}h_{x_N}^n = 2\Delta x h_{x_{N+1}}^{n-1}u_{x_{N+1}}^{n-1}.$$

By move the known terms to the right side will get

$$2\Delta x h_{x_{N+1}}^{n-1}u_{x_{N+1}}^n - \Delta t h_{x_{N+1}}^{n-1}u_{x_{N+1}}^{n-1}u_{x_N}^n - \Delta t g h_{x_{N+1}}^{n-1}h_{x_N}^n = 2\Delta x h_{x_{N+1}}^{n-1}u_{x_{N+1}}^{n-1} - \Delta t g h_{x_{N+1}}^{n-1}. \tag{52}$$

Now we can write the above equations (45), (46), (47), (48), (49), (50), (51), and (52) in matrix form as

$$AU^n = b \quad \text{or} \quad AU^{n+1} = b$$

$$\begin{bmatrix} 0 & p_2h_1^n & 0 & 0 & \cdots & 0 & 0 & 0 & p_1 & p_2u_1^{n-1} & 0 & 0 & \cdots & 0 & 0 & 0 \\ p_1h_1^n & p_2h_1^nu_1^n & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & p_2gh_1^n & 0 & 0 & \cdots & 0 & 0 & 0 \\ -p_2h_2^n & 0 & p_2h_2^n & 0 & \cdots & 0 & 0 & 0 & -p_2u_2^n & p_1 & p_2u_2^n & 0 & \cdots & 0 & 0 & 0 \\ -p_2h_2^nu_2^n & p_1h_2^n & p_2h_2^nu_2^n & 0 & \cdots & 0 & 0 & 0 & -p_2gh_2^n & 0 & p_2gh_2^n & 0 & \cdots & 0 & 0 & 0 \\ 0 & -p_2h_3^n & 0 & p_2h_3^n & \cdots & 0 & 0 & 0 & 0 & -p_2u_3^{n-1} & p_1x & p_2u_3^n & \cdots & 0 & 0 & 0 \\ 0 & -p_2h_3^nu_3^n & p_1h_3^n & p_2h_3^nu_3^n & \cdots & 0 & 0 & 0 & 0 & -p_2gh_3^n & 0 & p_2gh_3^n & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & -p_2h_{N+1}^n & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & -p_2u_{x_{N+1}}^n & p_1 \\ 0 & 0 & 0 & 0 & \cdots & 0 & -p_2h_{N+1}^nu_{N+1}^n & p_1h_{N+1}^n & 0 & 0 & 0 & 0 & \cdots & 0 & -p_2gh_{x_{N+1}}^n & 0 \end{bmatrix} \begin{bmatrix} u_1^{n+1} \\ u_2^{n+1} \\ u_3^{n+1} \\ u_4^{n+1} \\ \vdots \\ u_{x_{N+1}}^{n+1} \\ h_1^{n+1} \\ h_2^{n+1} \\ h_3^{n+1} \\ \vdots \\ h_{x_{N+1}}^n \end{bmatrix}$$

$$= \begin{bmatrix} p_1h_1^n + \Delta t u_1^n \\ p_1h_1^nu_1^{n-1} + p_2gh_1^n \\ p_1h_2^n \\ p_1h_2^nu_2^n \\ p_1h_3^n \\ p_1h_3^nu_3^n \\ \vdots \\ p_1h_{x_{N+1}}^n - p_2u_{x_{N+1}}^n \\ p_1h_{x_{N+1}}^nu_{x_{N+1}}^n - p_2gh_{x_{N+1}}^n \end{bmatrix}$$

Where $p_1 = 2\Delta x$, and $p_2 = \Delta t$. The above matrix is a system of $2(N+1)$ equations, now it has the linear form $A\vec{U} = \vec{b}$ which can be solved using Matlab code.

3.1.3 Forward Euler scheme

Forward Euler (FE) scheme, it given by applying forward approximation for time and central approximation for space as shown in Figure 6.

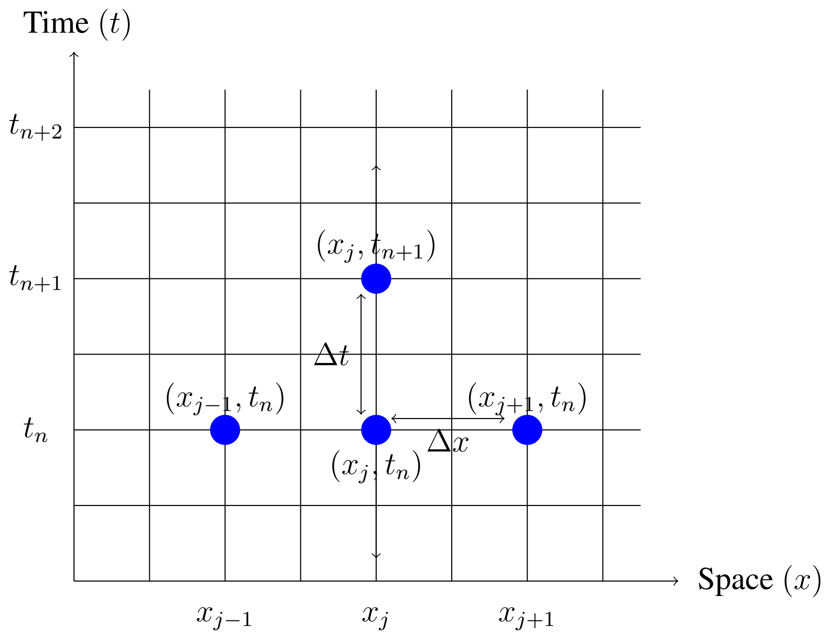

Figure 6. The mesh for Forward Euler scheme

Let's consider the system of shallow water

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) = 0 \tag{53}$$

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}(hu^2 + \frac{1}{2}gh^2) = 0. \tag{54}$$

By assuming that h and u are smooth, then equations (53) and (53) can be simplified by expanding the derivatives as

$$\frac{\partial h}{\partial t} = -\frac{\partial}{\partial x}(hu) = -h\frac{\partial u}{\partial x} - u\frac{\partial h}{\partial x} \tag{55}$$

and

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}(hu^2 + \frac{1}{2}gh^2) = 0$$
$$h\frac{\partial u}{\partial t} + u\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu^2) + \frac{1}{2}g\frac{\partial h^2}{\partial x} = 0.$$

The third term of the above equation is a derivative of multiplication of two functions, then

$$h\frac{\partial u}{\partial t} + u\frac{\partial h}{\partial t} + h\frac{\partial u^2}{\partial x} + u^2\frac{\partial h}{\partial x} + \frac{1}{2}g\frac{\partial h^2}{\partial x} = 0. \tag{56}$$

But we know that

$$\frac{\partial h^2}{\partial x} = 2h\frac{\partial h}{\partial x} \qquad \text{and} \qquad \frac{\partial u^2}{\partial x} = 2u\frac{\partial u}{\partial x}. \tag{57}$$

By substitute (57) in equation (56), will have

$$h\frac{\partial u}{\partial t} + u\frac{\partial h}{\partial t} + 2hu\frac{\partial u}{\partial x} + u^2\frac{\partial h}{\partial x} + gh\frac{\partial h}{\partial x} = 0. \tag{58}$$

From equation (55),

$$\frac{\partial h}{\partial t} = -\left(h\frac{\partial u}{\partial x} + u\frac{\partial h}{\partial x}\right) \tag{59}$$

and by substitute it in equation (58), will get

$$h\frac{\partial u}{\partial t} + u\left(-\left(h\frac{\partial u}{\partial x} + u\frac{\partial h}{\partial x}\right)\right) + 2hu\frac{\partial u}{\partial x} + u^2\frac{\partial h}{\partial x} + gh\frac{\partial h}{\partial x} = 0$$
$$h\frac{\partial u}{\partial t} - uh\frac{\partial u}{\partial x} - u^2\frac{\partial h}{\partial x} + 2hu\frac{\partial u}{\partial x} + u^2\frac{\partial h}{\partial x} + gh\frac{\partial h}{\partial x} = 0.$$

Simplify the above equation, will have

$$h\frac{\partial u}{\partial t} - uh\frac{\partial u}{\partial x} + 2uh\frac{\partial u}{\partial x} + gh\frac{\partial h}{\partial x} = 0$$
$$h\frac{\partial u}{\partial t} + uh\frac{\partial u}{\partial x} + gh\frac{\partial h}{\partial x} = 0.$$

Dividing the above equation by h, then

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x} = 0$$
$$\frac{\partial u}{\partial t} = -\left(u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x}\right).$$

The new version of the system of the shallow water equation is

$$\frac{\partial h}{\partial t} = -\left(h\frac{\partial u}{\partial x} + u\frac{\partial h}{\partial x}\right) \tag{60}$$

$$\frac{\partial u}{\partial t} = -\left(u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x}\right). \tag{61}$$

So by applying forward in time (see equation (32)), and central in space (see equation (34)), the Forward Euler (FE) scheme for equation (60) will be

$$\frac{h_j^{n+1} - h_j^n}{\Delta t} = -h_j^n\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} - u_j^n\frac{h_{j+1}^n - h_{j-1}^n}{2\Delta x} \tag{62}$$

and for equation (61)

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -u_j^n\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} - g\frac{h_{j+1}^n - h_{j-1}^n}{2\Delta x}. \tag{63}$$

In equations (62) and (63), both equations can be multiplied by Δt, and $\frac{1}{2\Delta x}$ can be used as a comment factor, so will get

$$h_j^{n+1} = h_j^n - \frac{\Delta t}{2\Delta x}\left(h_j^n(u_{j+1}^n - u_{j-1}^n) + u_j^n(h_{j+1}^n - h_{j-1}^n)\right)$$

and

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2\Delta x}\left(u_j^n(u_{j+1}^n - u_{j-1}^n) + g(h_{j+1}^n - h_{j-1}^n)\right).$$

Now the above equations can be written as

$$h_j^{n+1} = \left(\frac{\Delta t}{2\Delta x}u_j^n\right)h_{j-1}^n + h_j^n - \left(\frac{\Delta t}{2\Delta x}u_j^n\right)h_{j+1}^n + \left(\frac{\Delta t}{2\Delta x}h_j^n\right)u_{j-1}^n - \left(\frac{\Delta t}{2\Delta x}h_j^n\right)u_{j+1}^n \tag{64}$$

and

$$u_j^{n+1} = \left(\frac{g\Delta t}{2\Delta x}\right)h_{j-1}^n - \left(\frac{g\Delta t}{2\Delta x}\right)h_{j+1}^n + \left(\frac{\Delta t}{2\Delta x}u_j^n\right)u_{j-1}^n + u_j^n - \left(\frac{\Delta t}{2\Delta x}u_j^n\right)u_{j+1}^n. \tag{65}$$

Equation (64), and (65) are FE implicit and one step scheme.

For $j = 2$, substitute it in equation (64), will get

$$h_2^{n+1} = \left(\frac{\Delta t}{2\Delta x}u_2^n\right)h_1^n + h_2^n - \left(\frac{\Delta t}{2\Delta x}u_2^n\right)h_3^n + \left(\frac{\Delta t}{2\Delta x}h_2^n\right)u_1^n - \left(\frac{\Delta t}{2\Delta x}h_2^n\right)u_3^n$$

and from the boundary conditions, $h_1^n = 1$, and $u_1^n = 0$, so

$$h_2^{n+1} = \left(\frac{\Delta t}{2\Delta x}u_2^n\right) + h_2^n - \left(\frac{\Delta t}{2\Delta x}u_2^n\right)h_3^n - \left(\frac{\Delta t}{2\Delta x}h_2^n\right)u_3^n.$$

Simply, the above equation can be written as

$$h_2^{n+1} = h_2^n + \left(\frac{\Delta t}{2\Delta x}\right)u_2^n - \left(\frac{\Delta t}{2\Delta x}u_2^n\right)h_3^n - \left(\frac{\Delta t}{2\Delta x}h_2^n\right)u_3^n. \tag{66}$$

Again by substitute $j = 2$ in equation (65)

$$\begin{aligned} u_2^{n+1} &= \left(\frac{g\Delta t}{2\Delta x}\right)h_1^n - \left(\frac{g\Delta t}{2\Delta x}\right)h_3^n + \left(\frac{\Delta t}{2\Delta x}u_2^n\right)u_1^n + u_2^n - \left(\frac{\Delta t}{2\Delta x}u_2^n\right)u_3^n \\ &= \left(\frac{g\Delta t}{2\Delta x}\right) - \left(\frac{g\Delta t}{2\Delta x}\right)h_3^n + u_2^n - \left(\frac{\Delta t}{2\Delta x}u_2^n\right)u_3^n. \end{aligned}$$

Or by rewrite the above equation as

$$u_2^{n+1} = u_2^n - \left(\frac{g\Delta t}{2\Delta x}\right)h_3^n - \left(\frac{\Delta t}{2\Delta x}u_2^n\right)u_3^n + \left(\frac{g\Delta t}{2\Delta x}\right). \tag{67}$$

For $j = 3$, by substitute it in equation (64), will get

$$h_3^{n+1} = \left(\frac{\Delta t}{2\Delta x}u_3^n\right)h_2^n + h_3^n - \left(\frac{\Delta t}{2\Delta x}u_3^n\right)h_4^n + \left(\frac{\Delta t}{2\Delta x}h_3^n\right)u_2^n - \left(\frac{\Delta t}{2\Delta x}h_3^n\right)u_4^n \tag{68}$$

and in equation (65)

$$u_3^{n+1} = \left(\frac{g\Delta t}{2\Delta x}\right)h_2^n - \left(\frac{g\Delta t}{2\Delta x}\right)h_4^n + \left(\frac{\Delta t}{2\Delta x}u_3^n\right)u_2^n + u_3^n - \left(\frac{\Delta t}{2\Delta x}u_3^n\right)u_4^n. \tag{69}$$

For $j = 4$, by substitute it in equation (64), will have

$$h_4^{n+1} = \left(\frac{\Delta t}{2\Delta x}u_4^n\right)h_3^n + h_4^n - \left(\frac{\Delta t}{2\Delta x}u_4^n\right)h_5^n + \left(\frac{\Delta t}{2\Delta x}h_4^n\right)u_3^n - \left(\frac{\Delta t}{2\Delta x}h_4^n\right)u_5^n \tag{70}$$

and in equation (65)

$$u_4^{n+1} = \left(\frac{g\Delta t}{2\Delta x}\right)h_3^n - \left(\frac{g\Delta t}{2\Delta x}\right)h_5^n + \left(\frac{\Delta t}{2\Delta x}u_4^n\right)u_3^n + u_4^n - \left(\frac{\Delta t}{2\Delta x}u_4^n\right)u_5^n. \tag{71}$$

For $j = x_N$, in equation (64)

$$h_{x_N}^{n+1} = \left(\frac{\Delta t}{2\Delta x}u_{x_N}^n\right)h_{x_N-1}^n + h_{x_N}^n - \left(\frac{\Delta t}{2\Delta x}u_{x_N}^n\right)h_{x_N+1}^n + \left(\frac{\Delta t}{2\Delta x}h_{x_N}^n\right)u_{x_N-1}^n - \left(\frac{\Delta t}{2\Delta x}h_{x_N}^n\right)u_{x_N+1}^n.$$

From the boundary condition, $u_{x_{N+1}}^n = 0$, and $h_{x_{N+1}}^n = 1$, so will get

$$h_{x_N}^{n+1} = \left(\frac{\Delta t}{2\Delta x}u_{x_N}^n\right)h_{x_N-1}^n + h_{x_N}^n - \left(\frac{\Delta t}{2\Delta x}u_{x_N}^n\right) + \left(\frac{\Delta t}{2\Delta x}h_{x_N}^n\right)u_{x_N-1}^n.$$

The above equation can be rewritten as

$$h_{x_N}^{n+1} = \left(\frac{\Delta t}{2\Delta x}u_{x_N}^n\right)h_{x_N-1}^n + h_{x_N}^n + \left(\frac{\Delta t}{2\Delta x}h_{x_N}^n\right)u_{x_N-1}^n - \left(\frac{\Delta t}{2\Delta x}\right)u_{x_N}^n \qquad (72)$$

and in equation (65), by substitute $j = x_N$

$$u_{x_N}^{n+1} = \left(\frac{g\Delta t}{2\Delta x}\right)h_{x_N-1}^n - \left(\frac{g\Delta t}{2\Delta x}\right)h_{x_N+1}^n + \left(\frac{\Delta t}{2\Delta x}u_{x_N}^n\right)u_{x_N-1}^n + u_{x_N}^n - \left(\frac{\Delta t}{2\Delta x}u_{x_N}^n\right)u_{x_N+1}^n.$$

By rearrange the above equation, and from the boundary conditions, will get

$$u_{x_N}^{n+1} = \left(\frac{g\Delta t}{2\Delta x}\right)h_{x_N-1}^n + \left(\frac{\Delta t}{2\Delta x}u_{x_N}^n\right)u_{x_N-1}^n + u_{x_N}^n - \left(\frac{g\Delta t}{2\Delta x}\right). \qquad (73)$$

Now the above equations (66), (67), (68), (69), (70), (71), (72), and (73) can be written in matrix form as

$$U^{n+1} = AU^n + b$$

Or

$$
\begin{bmatrix} h_2^{n+1} \\ u_2^{n+1} \\ h_3^{n+1} \\ u_3^{n+1} \\ h_4^{n+1} \\ u_4^{n+1} \\ h_5^{n+1} \\ u_5^{n+1} \\ \vdots \\ \vdots \\ h_{x_N-1}^{n+1} \\ u_{x_N-1}^{n+1} \\ h_{x_N}^{n+1} \\ u_{x_N}^{n+1} \end{bmatrix}
=
\begin{bmatrix}
1 & gp & -pu_2^n & -ph_2^n & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 1 & -p & -pu_2^n & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
pu_3^n & ph_3^n & 1 & 0 & -pu_3^n & -ph_3^n & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
gp & gpu_3^n & 0 & 1 & -gp & -pu_3^n & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & pu_4^n & ph_4^n & 1 & 0 & -pu_4^n & -ph_4^n & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & gp & pu_4^n & 0 & 1 & -gp & -pu_4^n & 0 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & pu_{x_N}^n & ph_{x_N}^n & 1 & -p \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & gp & pu_{x_N}^n & 0 & 1
\end{bmatrix}
\begin{bmatrix} h_2^n \\ u_2^n \\ h_3^n \\ u_3^n \\ h_4^n \\ u_4^n \\ h_5^n \\ u_5^n \\ \vdots \\ \vdots \\ h_{x_N-1}^n \\ u_{x_N-1}^n \\ h_{x_N}^n \\ u_{x_N}^n \end{bmatrix}
+
\begin{bmatrix} 0 \\ gp \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ -gp \end{bmatrix}
$$

Where $p = \frac{\Delta t}{2\Delta x}$. The above matrix is a system of $2(N-1)$ equations, and it has the linear form $\vec{U}^{n+1} = A\vec{U}^n + \vec{b}$ which can be solved using Matlab code.

3.1.4 Crank-Nicolson method

The Crank-Nicolson (CN) method is one of the most important implicit methods. This method was developed by John Crank and Phyllis Nicolson in the middle of the 20^{th} century [20]. The computational effort for the CN method is higher than the other implicit method, since it is in this method we have an additional term, which is the explicit term, and the solution for the resulting system is a bit difficult than the other implicit method. But this difficulty makes the CN method the most accurate second-order method. [21]

Generally we can integrate backward and forward method as well as CN method by using single equation

$$\frac{U^{n+1} - U^n}{\Delta t} = \theta f(U^{n+1}) + (1 - \theta) f(U^n) \tag{74}$$

this approach is called θ-method approach where $\theta \in [0, 1]$. [22]

In θ-method approach, if $\theta = 0$, then equation (74), it just forward method (explicit), for $\theta = 1$, equation (74) will be backward method (implicit), and for $\theta = 0.5$, equation (74) it is CN method. So simply CN method can be described as in Figure 7 below.

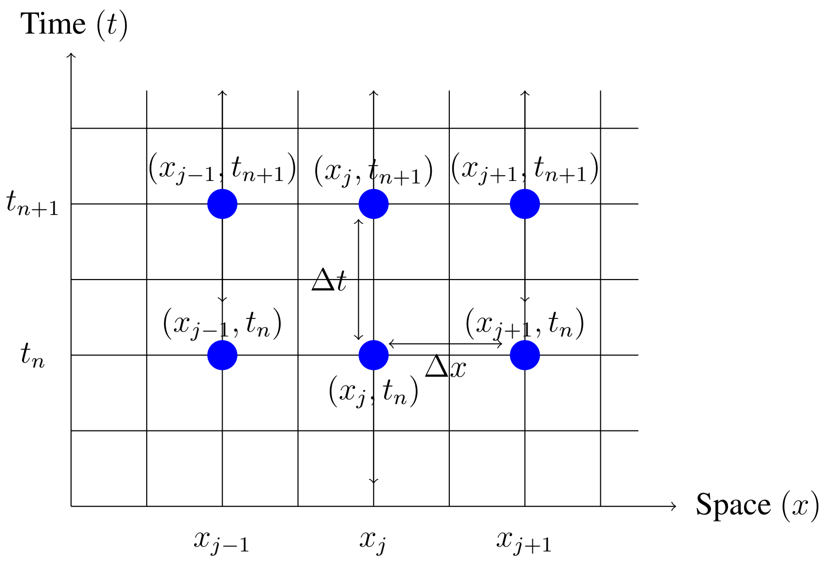

Figure 7. The mesh for Crank-Nicolson scheme

Now, following the small introduction of the CN method, the implementation of the CN method in the shallow water system will be provided below.

Crank Nicolson for the shallow water system

Recall the new version of the system of the shallow equation

$$\frac{\partial h}{\partial t} = -\left(h\frac{\partial u}{\partial x} + u\frac{\partial h}{\partial x}\right) = F_1(h, u, \frac{\partial h}{\partial x}, \frac{\partial u}{\partial x}) \tag{75}$$

$$\frac{\partial u}{\partial t} = -\left(u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x}\right) = F_2(h, u, \frac{\partial h}{\partial x}, \frac{\partial u}{\partial x}). \tag{76}$$

Then by applying Crank Nicolson (CN) in equations (75), the scheme for the height will be

$$h_j^{n+1} = h_j^n + \frac{\Delta t}{2}(F_1^{n+1} + F_1^n) \tag{77}$$

and in equation (76), the velocity scheme will be

$$u_j^{n+1} = u_j^n + \frac{\Delta t}{2}(F_2^{n+1} + F_2^n) \tag{78}$$

where F_1^n, F_2^n, are an explicit space scheme, and F_1^{n+1}, F_2^{n+1} are an implicit space scheme.

Thus, using the central space scheme and applying the CN method to the system, the first equation (77) will be

$$h_j^{n+1} = h_j^n - \frac{\Delta t}{2}\left(h_j^{n+1}\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\Delta x} + u_j^{n+1}\frac{h_{j+1}^{n+1} - h_{j-1}^{n+1}}{2\Delta x} + h_j^n\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + u_j^n\frac{h_{j+1}^n - h_{j-1}^n}{2\Delta x}\right).$$

Then, by taking $2\Delta x$ as a common factor, the above equation will be

$$h_j^{n+1} = h_j^n - \frac{\Delta t}{4\Delta x}(h_j^{n+1}(u_{j+1}^{n+1} - u_{j-1}^{n+1}) + u_j^{n+1}(h_{j+1}^{n+1} - h_{j-1}^{n+1}) + h_j^n(u_{j+1}^n - u_{j-1}^n) + u_j^n(h_{j+1}^n - h_{j-1}^n)). \tag{79}$$

And the second equation (78) will be

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{2}\left(u_j^{n+1}\frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\Delta x} + g\frac{h_{j+1}^{n+1} - h_{j-1}^{n+1}}{2\Delta x} + u_j^n\frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + g\frac{h_{j+1}^n + h_{j-1}^n}{2\Delta x}\right).$$

Again by taking $2\Delta x$ out as a common factor, will have

$$u_j^{n+1} = u_j^n - \frac{\Delta t}{4\Delta x}\left(u_j^{n+1}(u_{j+1}^{n+1} - u_{j-1}^{n+1}) + g(h_{j+1}^{n+1} - h_{j-1}^{n+1}) + u_j^n(u_{j+1}^n - u_{j-1}^n) + g(h_{j+1}^n - h_{j-1}^n)\right). \tag{80}$$

Then, by reordering equations (79) and (80), such that all variables in the time $n+1$in the left hand, and in the time n in the right hand, equation (79) will be

$$
\begin{aligned}
h_j^{n+1} + \frac{\Delta t}{4\Delta x}(h_j^{n+1}(u_{j+1}^{n+1} - u_{j-1}^{n+1}) + u_j^{n+1}(h_{j+1}^{n+1} - h_{j-1}^{n+1}) \\
= h_j^n - \frac{\Delta t}{4\Delta x}(h_j^n(u_{j+1}^n - u_{j-1}^n) + u_j^n(h_{j+1}^n - h_{j-1}^n)).
\end{aligned}
$$

Or by rearrange the above equation, will have

$$
\begin{aligned}
-\left(\frac{\Delta t}{4\Delta x}u_j^{n+1}\right)h_{j-1}^{n+1} - \left(\frac{\Delta t}{4\Delta x}h_j^{n+1}\right)u_{j-1}^{n+1} + h_j^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_j^{n+1}\right)h_{j+1}^{n+1} + \left(\frac{\Delta t}{4\Delta x}h_j^{n+1}\right)u_{j+1}^{n+1} \\
= \left(\frac{\Delta t}{4\Delta x}u_j^n\right)h_{j-1}^n + \left(\frac{\Delta t}{4\Delta x}h_j^n\right)u_{j-1}^n + h_j^n - \left(\frac{\Delta t}{4\Delta x}u_j^n\right)h_{j+1}^n - \left(\frac{\Delta t}{4\Delta x}h_j^n\right)u_{j+1}^n
\end{aligned}
\tag{81}
$$

and equation (80) will be

$$
\begin{aligned}
u_j^{n+1} + \frac{\Delta t}{4\Delta x}\left(u_j^{n+1}(u_{j+1}^{n+1} - u_{j-1}^{n+1}) + g(h_{j+1}^{n+1} - h_{j-1}^{n+1})\right) \\
= u_j^n - \frac{\Delta t}{4\Delta x}\left(u_j^n(u_{j+1}^n - u_{j-1}^n) + g(h_{j+1}^n - h_{j-1}^n)\right).
\end{aligned}
$$

Once again, we can rewrite it as

$$
\begin{aligned}
-\left(\frac{g\Delta t}{4\Delta x}\right)h_{j-1}^{n+1} - \left(\frac{\Delta t}{4\Delta x}u_j^{n+1}\right)u_{j-1}^{n+1} + u_j^{n+1} + \left(\frac{g\Delta t}{4\Delta x}\right)h_{j+1}^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_j^{n+1}\right)u_{j+1}^{n+1} \\
= \left(\frac{g\Delta t}{4\Delta x}\right)h_{j-1}^n + \left(\frac{\Delta t}{4\Delta x}u_j^n\right)u_{j-1}^n + u_j^n - \left(\frac{g\Delta t}{4\Delta x}\right)h_{j+1}^n - \left(\frac{\Delta t}{4\Delta x}u_j^n\right)u_{j+1}^n.
\end{aligned}
\tag{82}
$$

Now equations (81), and (82) are CN schemes for the shallow water model, and it nonlinear implicit scheme.

From the boundary conditions, the velocity at the beginning and the end is equal to zero ($u_1^n = u_{x_{N+1}}^n = 0\, meter/second$), and for height is equal to one ($h_1^n = h_{x_{N+1}}^n = 1\, meter$).

By substitute $j = 2$ in equation (81), will get

$$
\begin{aligned}
-\left(\frac{\Delta t}{4\Delta x}u_2^{n+1}\right)h_1^{n+1} - \left(\frac{\Delta t}{4\Delta x}h_2^{n+1}\right)u_1^{n+1} + h_2^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_2^{n+1}\right)h_3^{n+1} + \left(\frac{\Delta t}{4\Delta x}h_2^{n+1}\right)u_3^{n+1} \\
= \left(\frac{\Delta t}{4\Delta x}u_2^n\right)h_1^n + \left(\frac{\Delta t}{4\Delta x}h_2^n\right)u_1^n + h_2^n - \left(\frac{\Delta t}{4\Delta x}u_2^n\right)h_3^n - \left(\frac{\Delta t}{4\Delta x}h_2^n\right)u_3^n.
\end{aligned}
$$

Now, from the boundary conditions, the scheme for the first equation at $j = 2$ will be

$$-\left(\frac{\Delta t}{4\Delta x}u_2^{n+1}\right) + h_2^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_2^{n+1}\right)h_3^{n+1} + \left(\frac{\Delta t}{4\Delta x}h_2^{n+1}\right)u_3^{n+1} = \left(\frac{\Delta t}{4\Delta x}u_2^{n}\right) + h_2^n - \left(\frac{\Delta t}{4\Delta x}u_2^{n}\right)h_3^n - \left(\frac{\Delta t}{4\Delta x}h_2^{n}\right)u_3^n.$$

Or simply, the above equation can be rewritten as

$$h_2^{n+1} - \left(\frac{\Delta t}{4\Delta x}\right)u_2^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_2^{n+1}\right)h_3^{n+1} + \left(\frac{\Delta t}{4\Delta x}h_2^{n+1}\right)u_3^{n+1} = h_2^n + \left(\frac{\Delta t}{4\Delta x}\right)u_2^n - \left(\frac{\Delta t}{4\Delta x}u_2^{n}\right)h_3^n - \left(\frac{\Delta t}{4\Delta x}h_2^{n}\right)u_3^n. \tag{83}$$

Again by substitute $j = 2$ in equation (82), will have

$$-\left(\frac{g\Delta t}{4\Delta x}\right)h_1^{n+1} - \left(\frac{\Delta t}{4\Delta x}u_2^{n+1}\right)u_1^{n+1} + u_2^{n+1} + \left(\frac{g\Delta t}{4\Delta x}\right)h_3^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_2^{n+1}\right)u_3^{n+1}$$
$$= \left(\frac{g\Delta t}{4\Delta x}\right)h_1^n + \left(\frac{\Delta t}{4\Delta x}u_2^{n}\right)u_1^n + u_2^n - \left(\frac{g\Delta t}{4\Delta x}\right)h_3^n - \left(\frac{\Delta t}{4\Delta x}u_2^{n}\right)u_3^n$$
$$-\left(\frac{g\Delta t}{4\Delta x}\right) + u_2^{n+1} + \left(\frac{g\Delta t}{4\Delta x}\right)h_3^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_2^{n+1}\right)u_3^{n+1} = \left(\frac{g\Delta t}{4\Delta x}\right) + u_2^n - \left(\frac{g\Delta t}{4\Delta x}\right)h_3^n - \left(\frac{\Delta t}{4\Delta x}u_2^{n}\right)u_3^n.$$

By taking the first term in the right side to the left side, then the above equation will be

$$u_2^{n+1} + \left(\frac{g\Delta t}{4\Delta x}\right)h_3^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_2^{n+1}\right)u_3^{n+1} = u_2^n - \left(\frac{g\Delta t}{4\Delta x}\right)h_3^n - \left(\frac{\Delta t}{4\Delta x}u_2^{n}\right)u_3^n + \left(\frac{g\Delta t}{2\Delta x}\right). \tag{84}$$

For $j = 3$ in equation (81)

$$-\left(\frac{\Delta t}{4\Delta x}u_3^{n+1}\right)h_2^{n+1} - \left(\frac{\Delta t}{4\Delta x}h_3^{n+1}\right)u_2^{n+1} + h_3^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_3^{n+1}\right)h_4^{n+1} + \left(\frac{\Delta t}{4\Delta x}h_3^{n+1}\right)u_4^{n+1}$$
$$= \left(\frac{\Delta t}{4\Delta x}u_3^{n}\right)h_2^n + \left(\frac{\Delta t}{4\Delta x}h_3^{n}\right)u_2^n + h_3^n - \left(\frac{\Delta t}{4\Delta x}u_3^{n}\right)h_4^n - \left(\frac{\Delta t}{4\Delta x}h_3^{n}\right)u_4^n. \tag{85}$$

And for equation (82), when $j = 3$, will have

$$-\left(\frac{g\Delta t}{4\Delta x}\right)h_2^{n+1} - \left(\frac{\Delta t}{4\Delta x}u_3^{n+1}\right)u_2^{n+1} + u_3^{n+1} + \left(\frac{g\Delta t}{4\Delta x}\right)h_4^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_3^{n+1}\right)u_4^{n+1}$$
$$= \left(\frac{g\Delta t}{4\Delta x}\right)h_2^n + \left(\frac{\Delta t}{4\Delta x}u_3^{n}\right)u_2^n + u_3^n - \left(\frac{g\Delta t}{4\Delta x}\right)h_4^n - \left(\frac{\Delta t}{4\Delta x}u_3^{n}\right)u_4^n. \tag{86}$$

For $j = 4$ in equation (81), will get

$$-\left(\frac{\Delta t}{4\Delta x}u_4^{n+1}\right)h_3^{n+1} - \left(\frac{\Delta t}{4\Delta x}h_4^{n+1}\right)u_3^{n+1} + h_4^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_4^{n+1}\right)h_5^{n+1} + \left(\frac{\Delta t}{4\Delta x}h_3^{n+1}\right)u_5^{n+1}$$
$$= \left(\frac{\Delta t}{4\Delta x}u_4^{n}\right)h_3^{n} + \left(\frac{\Delta t}{4\Delta x}h_4^{n}\right)u_3^{n} + h_4^{n} - \left(\frac{\Delta t}{4\Delta x}u_4^{n}\right)h_5^{n} - \left(\frac{\Delta t}{4\Delta x}h_4^{n}\right)u_5^{n}. \tag{87}$$

And by substitute $j = 4$ in equation (82), will have

$$-\left(\frac{g\Delta t}{4\Delta x}\right)h_3^{n+1} - \left(\frac{\Delta t}{4\Delta x}u_4^{n+1}\right)u_3^{n+1} + u_4^{n+1} + \left(\frac{g\Delta t}{4\Delta x}\right)h_5^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_4^{n+1}\right)u_5^{n+1}$$
$$= \left(\frac{g\Delta t}{4\Delta x}\right)h_3^{n} + \left(\frac{\Delta t}{4\Delta x}u_4^{n}\right)u_3^{n} + u_4^{n} - \left(\frac{g\Delta t}{4\Delta x}\right)h_5^{n} - \left(\frac{\Delta t}{4\Delta x}u_4^{n}\right)u_5^{n}. \tag{88}$$

And for $j = x_N$ in equation (81), will have

$$-\left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n+1}\right)h_{x_{N-1}}^{n+1} - \left(\frac{\Delta t}{4\Delta x}h_{x_N}^{n+1}\right)u_{x_{N-1}}^{n+1} + h_{x_N}^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n+1}\right)h_{x_{N+1}}^{n+1} + \left(\frac{\Delta t}{4\Delta x}h_{x_N}^{n+1}\right)u_{x_{N+1}}^{n+1}$$
$$= \left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n}\right)h_{x_{N-1}}^{n} + \left(\frac{\Delta t}{4\Delta x}h_{x_N}^{n}\right)u_{x_{N-1}}^{n} + h_{x_N}^{n} - \left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n}\right)h_{x_{N+1}}^{n} - \left(\frac{\Delta t}{4\Delta x}h_{x_N}^{n}\right)u_{x_{N+1}}^{n}.$$

From boundary conditions, the height $h_{x_{N+1}}^{n+1} = h_{x_{N+1}}^{n} = 1$, and the velocity $u_{x_{N+1}}^{n+1} = u_{x_{N+1}}^{n} = 0$, so that by applying these conditions to the above equation, will have

$$-\left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n+1}\right)h_{x_{N-1}}^{n+1} - \left(\frac{\Delta t}{4\Delta x}h_{x_N}^{n+1}\right)u_{x_{N-1}}^{n+1} + h_{x_N}^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n+1}\right) = \left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n}\right)h_{x_{N-1}}^{n}$$
$$+ \left(\frac{\Delta t}{4\Delta x}h_{x_N}^{n}\right)u_{x_{N-1}}^{n} + h_{x_N}^{n} - \left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n}\right).$$

Or by rearrange the above equation as

$$-\left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n+1}\right)h_{x_{N-1}}^{n+1} - \left(\frac{\Delta t}{4\Delta x}h_{x_N}^{n+1}\right)u_{x_{N-1}}^{n+1} + h_{x_N}^{n+1} + \left(\frac{\Delta t}{4\Delta x}\right)u_{x_N}^{n+1} = \left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n}\right)h_{x_{N-1}}^{n}$$
$$+ \left(\frac{\Delta t}{4\Delta x}h_{x_N}^{n}\right)u_{x_{N-1}}^{n} + h_{x_N}^{n} - \left(\frac{\Delta t}{4\Delta x}\right)u_{x_N}^{n}. \tag{89}$$

For equation (82) by substitute $j = x_N$, will get

$$-\left(\frac{g\Delta t}{4\Delta x}\right)h_{x_{N-1}}^{n+1} - \left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n+1}\right)u_{x_{N-1}}^{n+1} + u_{x_N}^{n+1} + \left(\frac{g\Delta t}{4\Delta x}\right)h_{x_{N+1}}^{n+1} + \left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n+1}\right)u_{x_{N+1}}^{n+1}$$
$$= \left(\frac{g\Delta t}{4\Delta x}\right)h_{x_{N-1}}^{n} + \left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n}\right)u_{x_{N-1}}^{n} + u_{x_N}^{n} - \left(\frac{g\Delta t}{4\Delta x}\right)h_{x_{N+1}}^{n} - \left(\frac{\Delta t}{4\Delta x}u_{x_N}^{n}\right)u_{x_{N+1}}^{n}.$$

Applying the boundary conditions, the above equation will be

$$-\left(\frac{g\Delta t}{4\Delta x}\right)h^{n+1}_{x_{N-1}}-\left(\frac{\Delta t}{4\Delta x}u^{n+1}_{x_N}\right)u^{n+1}_{x_{N-1}}+u^{n+1}_{x_N}+\left(\frac{g\Delta t}{4\Delta x}\right)=\left(\frac{g\Delta t}{4\Delta x}\right)h^{n}_{x_{N-1}}$$
$$+\left(\frac{\Delta t}{4\Delta x}u^{n}_{x_N}\right)u^{n}_{x_{N-1}}+u^{n}_{x_N}-\left(\frac{g\Delta t}{4\Delta x}\right)$$

Then by moving the first term in the right side to the left side

$$-\left(\frac{g\Delta t}{4\Delta x}\right)h^{n+1}_{x_{N-1}}-\left(\frac{\Delta t}{4\Delta x}u^{n+1}_{x_N}\right)u^{n+1}_{x_{N-1}}+u^{n+1}_{x_N}=\left(\frac{g\Delta t}{4\Delta x}\right)h^{n}_{x_{N-1}}+\left(\frac{\Delta t}{4\Delta x}u^{n}_{x_N}\right)u^{n}_{x_{N-1}}$$
$$+u^{n}_{x_N}-\left(\frac{g\Delta t}{2\Delta x}\right). \tag{90}$$

So equations (83), (84), (85), (86), (87), (88), (89), and (90) which are nonlinear system of algebraic equations, can be written in matrix form as

$$A_1U^{n+1}=A_2U^n+b$$

and by putting $p=\frac{\Delta t}{4\Delta x}$, the matrices will be

$$
\begin{bmatrix}
1 & -p & pu_2^{n+1} & ph_2^{n+1} & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 1 & p & pu_2^{n+1} & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
-pu_3^{n+1} & -\frac{\Delta t}{4\Delta x}h_3^{n+1} & 1 & 0 & pu_3^{n+1} & ph_3^{n+1} & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
-gp & -pu_3^{n+1} & 0 & 1 & gp & pu_3^{n+1} & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & -pu_4^{n+1} & -ph_4^{n+1} & 1 & 0 & pu_4^{n+1} & ph_4^{n+1} & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & -gp & -pu_4^{n+1} & 0 & 1 & gp & pu_4^{n+1} & 0 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & -pu_{x_N}^{n+1} & -ph_{x_N}^{n+1} & 1 & p \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & -gp & -pu_{x_N}^{n+1} & 0 & 1
\end{bmatrix}
\begin{bmatrix}
h_2^{n+1} \\ u_2^{n+1} \\ h_3^{n+1} \\ u_3^{n+1} \\ h_4^{n+1} \\ u_4^{n+1} \\ h_5^{n+1} \\ u_5^{n+1} \\ \vdots \\ \vdots \\ \vdots \\ h_{x_N-1}^{n+1} \\ u_{x_N-1}^{n+1} \\ h_{x_N}^{n+1} \\ u_{x_N}^{n+1}
\end{bmatrix}
$$

$$
= \begin{bmatrix}
1 & p & -pu_2^n & -ph_2^n & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 1 & -gp & -pu_2^n & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
pu_3^n & ph_3^n & 1 & 0 & -pu_3^n & -ph_3^n & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
gp & pu_3^n & 0 & 1 & -gp & -pu_3^n & \cdots & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & pu_4^n & ph_4^n & 1 & 0 & -pu_4^n & -pu_4^h & 0 & \cdots & 0 & 0 & 0 & 0 \\
0 & 0 & gp & pu_4^n & 0 & 1 & -gp & -pu_4^n & 0 & \cdots & 0 & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & pu_{x_N}^n & ph_{x_N}^n & 1 & -p \\
0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & gp & pu_{x_N}^n & 0 & 1
\end{bmatrix}
\begin{bmatrix}
h_2^n \\ u_2^n \\ h_3^n \\ u_3^n \\ h_4^n \\ u_4^n \\ h_5^n \\ u_5^n \\ \vdots \\ \vdots \\ \vdots \\ h_{x_N-1}^n \\ u_{x_N-1}^n \\ h_{x_N}^n \\ u_{x_N}^n
\end{bmatrix}
+
\begin{bmatrix}
0 \\ gp \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ 0 \\ -gp
\end{bmatrix}
$$

3.2 Method of Lines

The Method of Lines (MOL) is one of the numerical methods used to solve partial differential equations (PDEs), and is one of the most commonly used numerical methods for PDEs. MOL convert PDE to a system of ordinary differential equations (ODEs) by discretizing the spatial derivative only and leaving the derivative of time as a variable, this leads to an ODE system that can be solved by using any numerical method for solving the ODE system. [23] Equation (91) describe the conversion from PDE to ODEs.

$$\frac{\partial U}{\partial t} = F(U,t) \quad \rightarrow \quad \frac{dU_i}{dt} = F_i(U_i,t) \tag{91}$$

where F represents a PDE, and F_i is the discretized space derivative. To discretize space, we can use any numerical approximations, such as finite difference approximations (i.e. forward, backward, or central approximations) or polynomials of any degree,...etc. Here is this thesis, polynomial with fifth-degree will be used to solve the system of the shallow water equation.

Method of Lines for shallow water model

In this method, as mentioned earlier, the plan was a discretized space by using a fifth-degree polynomial (five points), with this idea, a general scheme for finding the derivative of any function with respect to space was established, and the scheme was suitable for one-dimensional problems. However, since the flow of the shallow water in two directions (left and right) at the same time and this situation is a special case for the flow in one dimension, and also from the impact of the boundaries conditions, the scheme showed some limitations and instability in the result. So two points for the discretization of space were decided to used, namely central approximation.

Now let's recall the new version the system of the shallow water equation which are

$$\frac{\partial h}{\partial t} = -\left(h\frac{\partial u}{\partial x} + u\frac{\partial h}{\partial x}\right) \tag{92}$$

$$\frac{\partial u}{\partial t} = -\left(u\frac{\partial u}{\partial x} + g\frac{\partial h}{\partial x}\right). \tag{93}$$

The above system can be converted to an ODE system by using the central approximation to discreet the space as

$$\frac{dh_j}{dt} = -h_j \frac{u_{j+1} - u_{j-1}}{2\Delta x} - u_j \frac{h_{j+1} - h_{j-1}}{2\Delta x}$$

$$\frac{du_j}{dt} = -u_j^n \frac{u_{j+1} - u_{j-1}}{2\Delta x} - g \frac{h_{j+1} - h_{j-1}}{2\Delta x}.$$

By taking $2\Delta x$ as a common factor, the above equations will be

$$\frac{dh_j}{dt} = -\frac{1}{2\Delta x}(h_j(u_{j+1} - u_{j-1}) + u_j(h_{j+1} - h_{j-1})) \tag{94}$$

$$\frac{du_j}{dt} = -\frac{1}{2\Delta x}(u_j(u_{j+1} - u_{j-1}) + g(h_{j+1} - h_{j-1})). \tag{95}$$

which are the system of ODE equation.

By substitute $j = 2$ in equation (94), will have

$$\frac{dh_2}{dt} = -\frac{1}{2\Delta x}(h_2(u_3 - u_1) + u_2(h_3 - h_1)).$$

And as far as boundary conditions are concerned, the height $h_1 = 1$,and $u_1 = 0$, then will get

$$\begin{aligned}\frac{dh_2}{dt} &= -\frac{1}{2\Delta x}(h_2(u_3 - 0) + u_2(h_3 - 1)) \\ &= -\frac{1}{2\Delta x}(h_2 u_3 + u_2 h_3 - u_2).\end{aligned}$$

Or the equation above can be written as

$$\frac{dh_2}{dt} = \left(\frac{1}{2\Delta x}\right) u_2 - \left(\frac{1}{2\Delta x} u_2\right) h_3 - \left(\frac{1}{2\Delta x} h_2\right) u_3 \tag{96}$$

Also by substitute $j = 2$ is equation (95), will have

$$\begin{aligned}\frac{du_2}{dt} &= -\frac{1}{2\Delta x}(u_2(u_3 - u_1) + g(h_3 - h_1)) \\ &= -\frac{1}{2\Delta x}(u_2(u_3 - 0) + g(h_3 - 1)) \\ &= -\frac{1}{2\Delta x}(u_2 u_3 + g h_3 - g).\end{aligned}$$

Or we can rewrite as

$$\frac{du_2}{dt} = -\left(\frac{g}{2\Delta x}\right)h_3 - \left(\frac{1}{2\Delta x}u_2\right)u_3 + \left(\frac{g}{2\Delta x}\right). \tag{97}$$

For $j = 3$ in equation (94), will get

$$\frac{dh_3}{dt} = -\frac{1}{2\Delta x}(h_3(u_4 - u_2) + u_3(h_4 - h_2)).$$

The above equation can be write as

$$\frac{dh_3}{dt} = \left(\frac{1}{2\Delta x}u_3\right)h_2 + \left(\frac{1}{2\Delta x}h_3\right)u_2 - \left(\frac{1}{2\Delta x}u_3\right)h_4 - \left(\frac{1}{2\Delta x}h_3\right)u_4. \tag{98}$$

For equation (95), will have

$$\frac{du_3}{dt} = -\frac{1}{2\Delta x}(u_3(u_4 - u_2) + g(h_4 - h_2)).$$

Or by rearrange the above equation will get

$$\frac{dh_3}{dt} = \left(\frac{g}{2\Delta x}\right)h_2 + \left(\frac{1}{2\Delta x}u_3\right)u_2 - \left(\frac{g}{2\Delta x}\right)h_4 - \left(\frac{1}{2\Delta x}u_3\right)u_4. \tag{99}$$

For $j = 4$ in equation (94),

$$\frac{dh_4}{dt} = -\frac{1}{2\Delta x}(h_4(u_5 - u_3) + u_4(h_5 - h_3)).$$

The equation above can be written as

$$\frac{dh_4}{dt} = \left(\frac{1}{2\Delta x}u_4\right)h_3 + \left(\frac{1}{2\Delta x}h_4\right)u_3 - \left(\frac{1}{2\Delta x}u_4\right)h_5 - \left(\frac{1}{2\Delta x}h_4\right)u_5. \tag{100}$$

Same for equation (95), will have

$$\frac{du_4}{dt} = -\frac{1}{2\Delta x}(u_4(u_5 - u_3) + g(h_5 - h_3)).$$

By reordering the above equation, will get

$$\frac{dh_4}{dt} = \left(\frac{g}{2\Delta x}\right)h_3 + \left(\frac{1}{2\Delta x}u_4\right)u_3 - \left(\frac{g}{2\Delta x}\right)h_5 - \left(\frac{1}{2\Delta x}u_4\right)u_5. \tag{101}$$

For $j = x_N$ in equation (94)

$$\frac{dh_{x_N}}{dt} = -\frac{1}{2\Delta x}(h_{x_N}(u_{x_{N+1}} - u_{x_{N-1}}) + u_{x_N}(h_{x_{N+1}} - h_{x_{N-1}})).$$

Form the boundary conditions, $u_{x_{N+1}} = 0$, $h_{x_{N+1}} = 1$, then will have

$$\begin{aligned}\frac{dh_{x_N}}{dt} &= -\frac{1}{2\Delta x}(h_{x_N}(0 - u_{x_{N-1}}) + u_{x_N}(1 - h_{x_{N-1}})) \\ &= -\frac{1}{2\Delta x}(-h_{x_N}u_{x_{N-1}} + u^n_{x_N} - u_{x_N}h_{x_{N-1}}).\end{aligned}$$

Simply

$$\frac{dh_{x_N}}{dt} = \left(\frac{1}{2\Delta x}u_{x_N}\right)h_{x_{N-1}} + \left(\frac{1}{2\Delta x}h_{x_N}\right)u_{x_{N-1}} - \left(\frac{1}{2\Delta x}\right)u_{x_N}. \tag{102}$$

Again for equation (95) by substitute $j = x_N$ will get

$$\begin{aligned}\frac{du_{x_N}}{dt} &= -\frac{1}{2\Delta x}(u_{x_N}(u_{x_{N+1}} - u_{x_{N-1}}) + g(h_{x_{N+1}} - h_{x_{N-1}})) \\ &= -\frac{1}{2\Delta x}(u_{x_N}(0 - u_{x_{N-1}}) + g(1 - h_{x_{N-1}})) \\ &= -\frac{1}{2\Delta x}(-u_{x_N}u_{x_{N-1}} + g - gh_{x_{N-1}}).\end{aligned}$$

Or we can write it as

$$\frac{du_{x_N}}{dt} = \left(\frac{g}{2\Delta x}\right)h_{x_{N-1}} + \left(\frac{1}{2\Delta x}u_{x_N}\right)u_{x_{N-1}} - \left(\frac{g}{2\Delta x}\right). \tag{103}$$

Now equations (96), (97), (98), (99), (100), (101), (102) and (103) can be written in matrix form as

$$\frac{dU}{dt} = AU + b \tag{104}$$

Or by putting $p = \frac{1}{2\Delta x}$, the matrix form will be

$$
\frac{d}{dt}\begin{bmatrix} h_2 \\ u_2 \\ h_3 \\ u_3 \\ h_4 \\ u_4 \\ h_5 \\ u_5 \\ \vdots \\ \vdots \\ \vdots \\ h_{x_N-1} \\ u_{x_N-1} \\ h_{x_N} \\ u_{x_N} \end{bmatrix} = \begin{bmatrix} 0 & p & -u_2 & -ph_2 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & gp & -pu_2 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ pu_3 & ph_3 & 0 & 0 & -pu_3 & -ph_3 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ gp & pu_3 & 0 & 0 & -gp & -pu_3 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & pu_4 & ph_4 & 0 & 0 & -pu_4 & -ph_4 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & gp & pu_4 & 0 & 0 & -gp & -pu_4 & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & pu_{x_N} & ph_{x_N} & 0 & -p \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & \cdots & gp & pu_{x_N} & 0 & 0 \end{bmatrix} + \begin{bmatrix} 0 \\ gp \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \\ 0 \\ -gp \end{bmatrix}
$$

Now to solve this system of ODEs, Matlab ode solver will be used.

3.2.1 Matlab ODE solver

Matlab provides different version solvers for solving ODE, these solvers take the ODE, the initial value for the solution, the time span, and return the values of the derivative. The general formula for the solvers is

$$[t, y] = \text{ odeSolver } (\text{@odefun}, \text{ tspan }, y_0)$$

The solver takes the ODE system (odefun), and the span of the time that we want to find the solution on it (tspan), and initial values for the solution (y_0) as input and return the vector of the solution y with the corresponding times t. Matlab has different functions for the ODE solver, ode23, ode113, ode15s, ode23s, ode23t, ode23tb, and ode15i.

ode15s

Function ode15s is the Matlab function to solve an ODE equation, or even an ODE system. It is a Variable-Step, Variable-Order (VSVO) solver based on the Number Differentiation Formula (NDFs) of order 1 through 5.

ode23s and ode45

The functions ode45 and ode23s is a single step ODE solvers. The method behind those solvers are Runge-Kutta methods. To calculate each step we need information only from the previous step (independent from the previous steps). This method passed information from one step to the next one. For the step size h the solver uses a strategy call as FSAL. the final function value at the end of a step is used as initial value for the function at the next step. The codes for ode23 and ode45 are very similar. The only difference between the two solvers is the sets of the key parameters.

Now different schemes for the shallow water model by using different numerical methods were applied, in the next chapter, Numerical results for all these schemes will present, and also a comparison between this result will be shown.

4 Numerical Results for Shallow Water Model

The numerical results are obtained by solving the system of linear/nonlinear algebraic equations and an ODE system using different numerical methods. For each numerical model, the simulation is carried out for up to 3 sec of flow physical time. Here in this chapter, the numerical results will present and compare. These all numerical model were implemented in Matlab.

Generally, the task for all the numerical methods used is to find $\vec{U^n}$ at every time n, where

$$\vec{U^n} = [h_0^n, u_0^n, h_1^n, u_1^n, h_2^n, u_2^n, ..., h_{x_N}^n, u_{x_N}^n, h_{x_{N+1}}^n, u_{x_{N+1}}^n]^T$$

The initial conditions for height and velocity

$$\begin{aligned} u(x,0) &= 0 \text{ for all x} \in [x_a, x_b] \\ &\Longrightarrow u_j^0 = 0 \text{ for all j} \in [x_a, x_b] \\ h(x,0) &= 1 + \frac{2}{5}e^{-5x^2} \text{ for all x} \in [x_a, x_b] \\ &\Longrightarrow h_j^0 = 1 + \frac{2}{5}e^{-5x^2} \text{ for all j} \in [x_a, x_b] \end{aligned}$$

where $[x_a, x_b]$ is the interval $[-5, 5]$.

Then to find $\vec{U^1}$ we need to use $\vec{U^0}$ (which is our initial vector), and to find $\vec{U^2}$ we need to use $\vec{U^1}$ (which we calculate it in the previous step), and we continue like this until we reach to the last time ($\vec{U^n}$).

4.1 Description of the solution behavior

Analytical results for a system of shallow water equations can be found using Matlab. The solution was recorded in a time interval between zero seconds and three seconds. Figures below describe the behavior of the solution at this time interval.

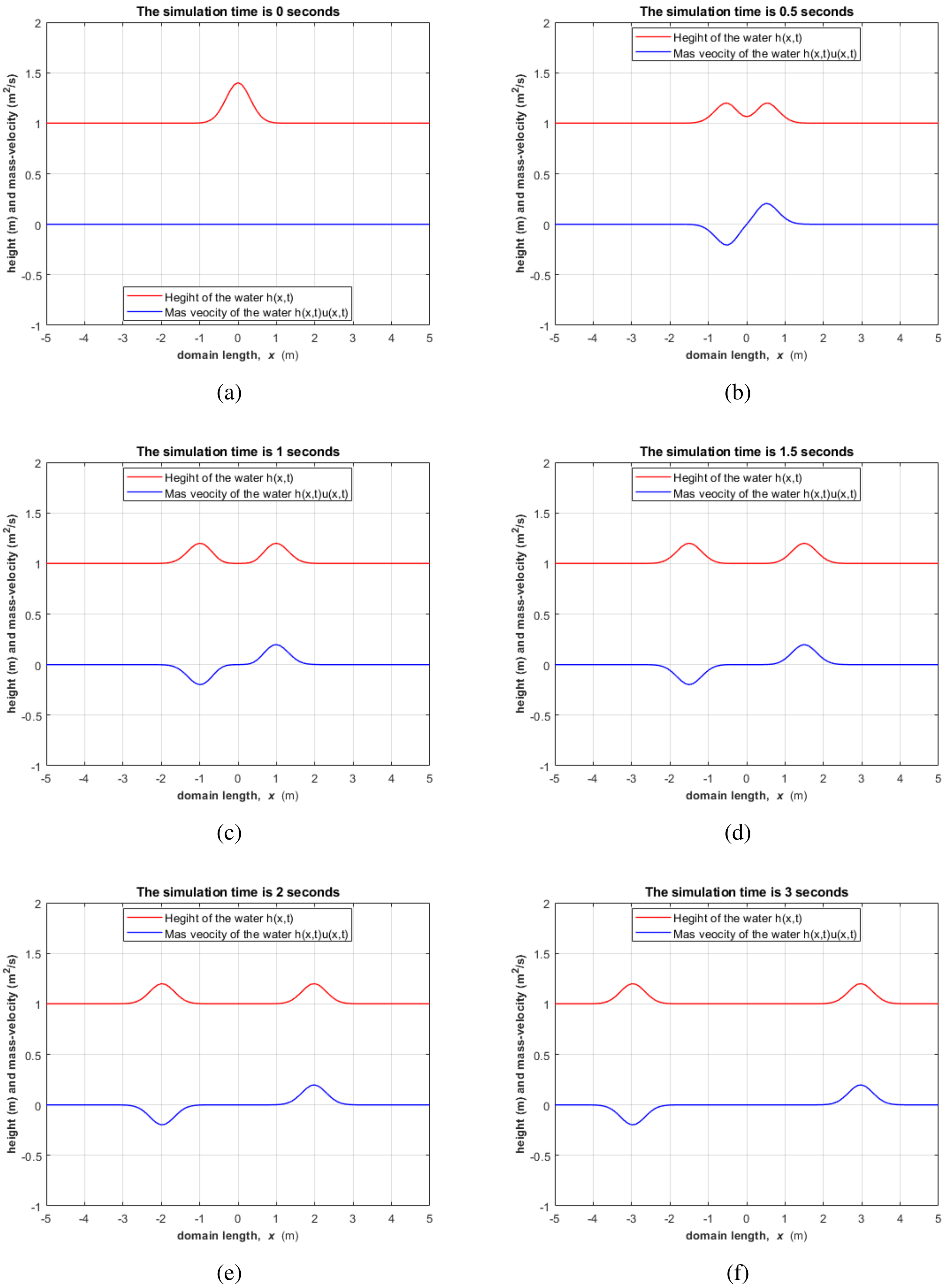

Figure 8. The analytical results for the height and mass velocity: (a) at time $t = 0s$; (b) at time $t = 0.5s$; (c) at time $t = 1s$; (d) at time $t = 1.5s$; (e) at time $t = 2s$; (f) at time $t = 3s$.

In the above figures, the height wave initially has one maximum point at the center of the space, and the mass velocity is zero for all space interval. As time increases, the height wave starts to separate in the middle to have a maximum of two points. The gap between these maximum points increases with time until it reaches the boundaries in the third second. The mass velocity begins to divide in the range $x \in [-1, 1]$ at the time $t = 0, 5s$, the wave decreases to the minimum point, and then increase until the maximum point is reached. After that, the wave begins to decrease again to reach zero value and remains constant at that value. As time passes, the gap between the minimum and the maximum points also increases in the wave of mass velocity, until it reaches the boundaries in the third seconds.

The numerical results for all four methods will be shown in the next sections, these results will be found using the Matlab code, with a number of $N = 100$ space discretization nodes in a space interval $x \in [-5, 5]$, and a time step size of $\Delta t = 0.0005$ in a time interval of three seconds.

4.2 Backward Euler results

Applying the Backward Euler (BE) scheme, which is Backward approximation in time and Central Difference in space, to the shallow water system gives us a system of a linear algebraic equation in the form of

$$A\vec{U^n} = b. \tag{105}$$

Here we want to find $\vec{U}^n$ at each time n. Matrix A is a function of the height and the velocity at time $n-1$, then the solution will be in the form of

$$\vec{U^n} = A^{-1}b. \tag{106}$$

Numerical results of the BE scheme, together with the analytical solution, are shown in Figure 9.

At the time $t = 0.5s$, the BE result showed a small difference in the local minimum of the height wave at $x = 0$. Mass velocity matches the analytical solution. With the increase in time, we can see that there is a very small difference in the maximum value and the position of the maximum value for the height wave, this difference increases with the increase in time. Also for the mass velocity , with respect to the analytical solution, the minimum and the maximum values of the wave have some differences at the same location, especially for the third seconds.

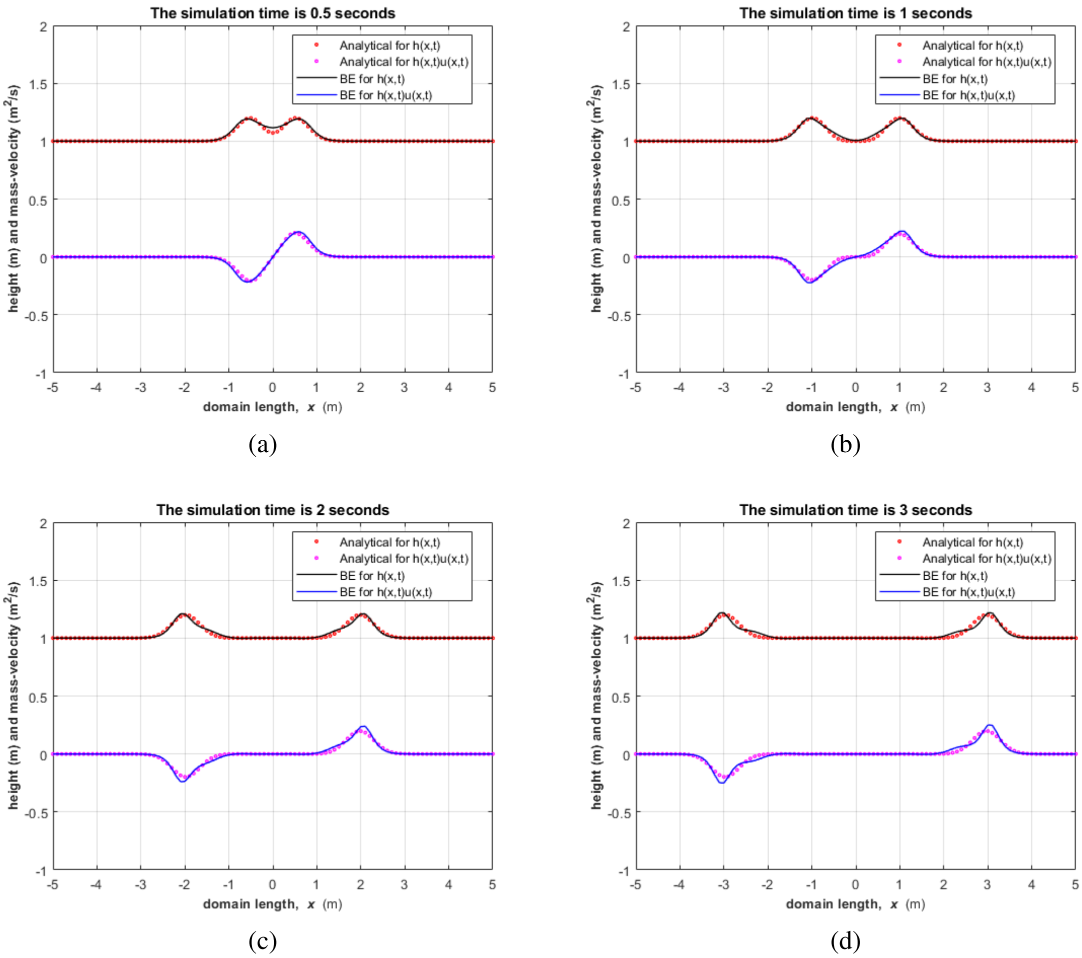

Figure 9. Results of the BE scheme compared with the analytical solution for the height and mass velocity: (a) at time $t = 0.5s$; (b) at time $t = 1s$; (c) at time $t = 2s$; (d) at time $t = 3s$.

4.3 Forward Euler results

Forward Euler, which is Forward Euler in time and Central Difference in space, gives us a system of linear algebraic equations in the form of

$$\vec{U^n} = A\vec{U^{n-1}} + b \tag{107}$$

where matrix A is a function of the height and the velocity at time $n - 1$. The results of the Forward Euler scheme are as shown below

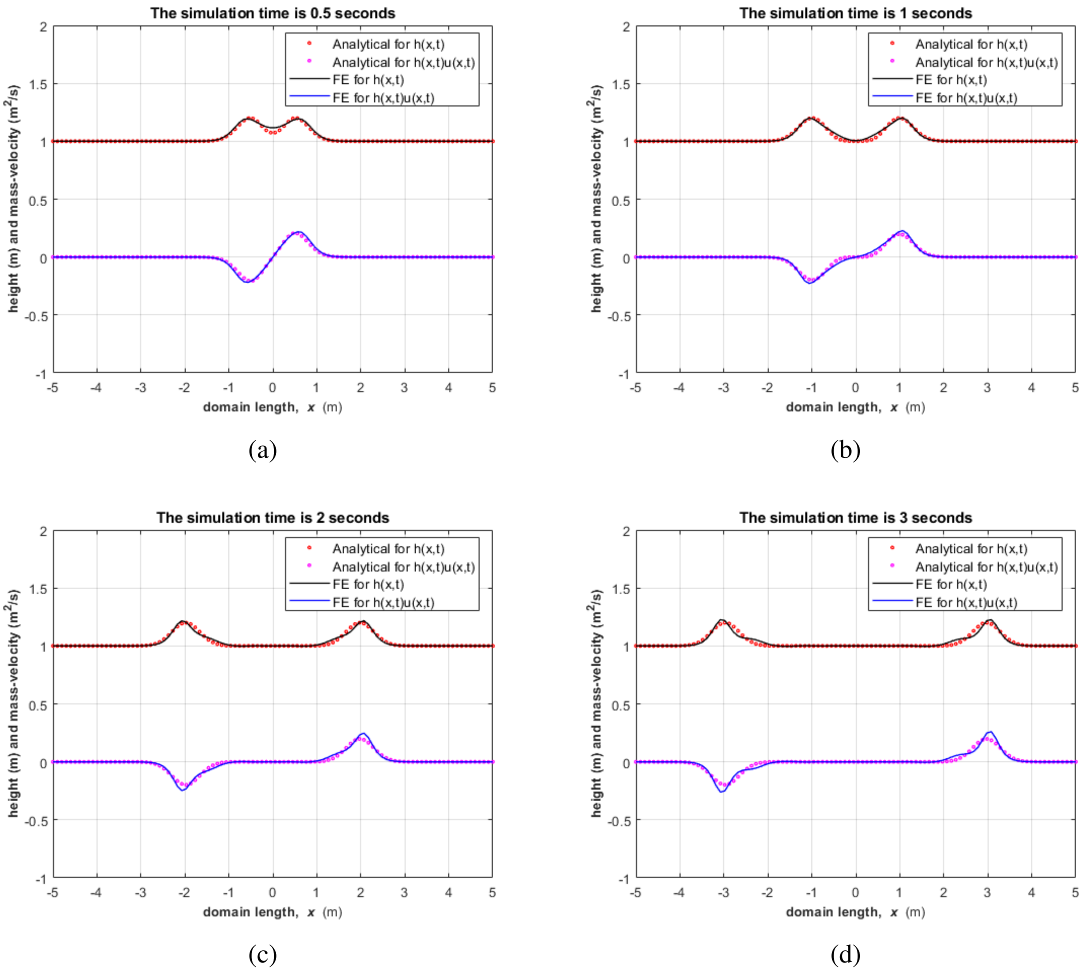

Figure 10. Results of the FE scheme compared with the analytical solution for the height and mass velocity: (a) at time $t = 0.5s$; (b) at time $t = 1s$; (c) at time $t = 2s$; (d) at time $t = 3s$.

The result of the Forward Euler scheme at the time of $t = 0.5, 1s$ is almost the same as the analytical result, a small mismatch at the local minimum (at $x = 0$)of the height wave. From the second second, the waves start not to be as smooth as the waves of the analytical solution. However the results for the FE scheme are good when we compare it with the analytical solution.

The limitation of the Forward Euler scheme

This scheme has faced the problem of instability when Courant number ($C = u\frac{\Delta t}{\Delta x}$, where Δx is space steps, and Δt is time steps) is more than one, so in order to reduce this number, we need to increase Δx, and decrease Δt at the same time

4.4 Crank-Nicolson results

By applying Crank-Nicolson scheme to the system of shallow water equations, we shall obtain a system of nonlinear algebraic equations in the form

$$A_1\vec{U^n} = A_2\vec{U^{n-1}} + b. \tag{108}$$

And the solution of this system will be

$$\vec{U^n} = A_1^{-1}(A_2\vec{U^{n-1}} + b) \tag{109}$$

where matrix A_2 is a function of the height and the velocity at time $n-1$, and matrix A_1 is a function of the height and the velocity at time n. To solve this non-linearity problem in matrix A_1 we use Picard method [24], which assumes that the values of h, and u at the time n are equal to the values at the time $n-1$.

the results of the CN scheme for the shallow water model at different times as shown in the below figures.

From the results of the CN scheme at times $t = 3s$, a small mismatch at the intervals $x \in [-3,-2]$, and $[2,3]$, and also at the maximum points for the height and the mass velocity waves, and the minimum for the mass velocity wave. However the results are good as comparison with the analytical solution.

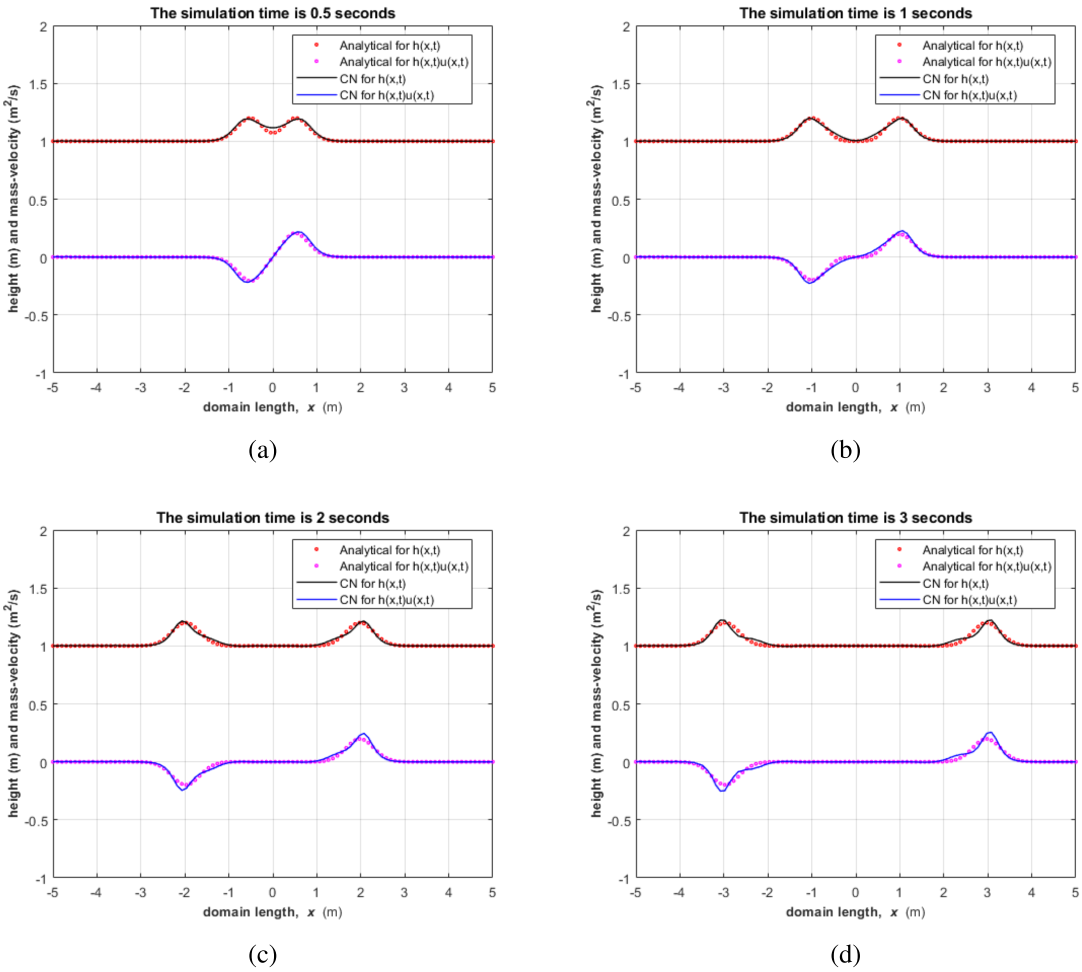

Figure 11. Results of the CN scheme compared with the analytical for the height and mass velocity: (a) at time $t = 0.5s$; (b) at time $t = 1s$; (c) at time $t = 2s$; (d) at time $t = 3s$.

4.5 Method of line results

The implementation of this method on the shallow water equation system gives us an ODE system in the form of

$$\frac{d}{dt}\vec{U} = A\vec{U} + b \tag{110}$$

After running the scheme for different Matlab's ODE solvers, namely ode15s, ode23s, and ode45s, it was found that ode23s was the best one in terms of accuracy and hence it has been used in the case of MOL technique.

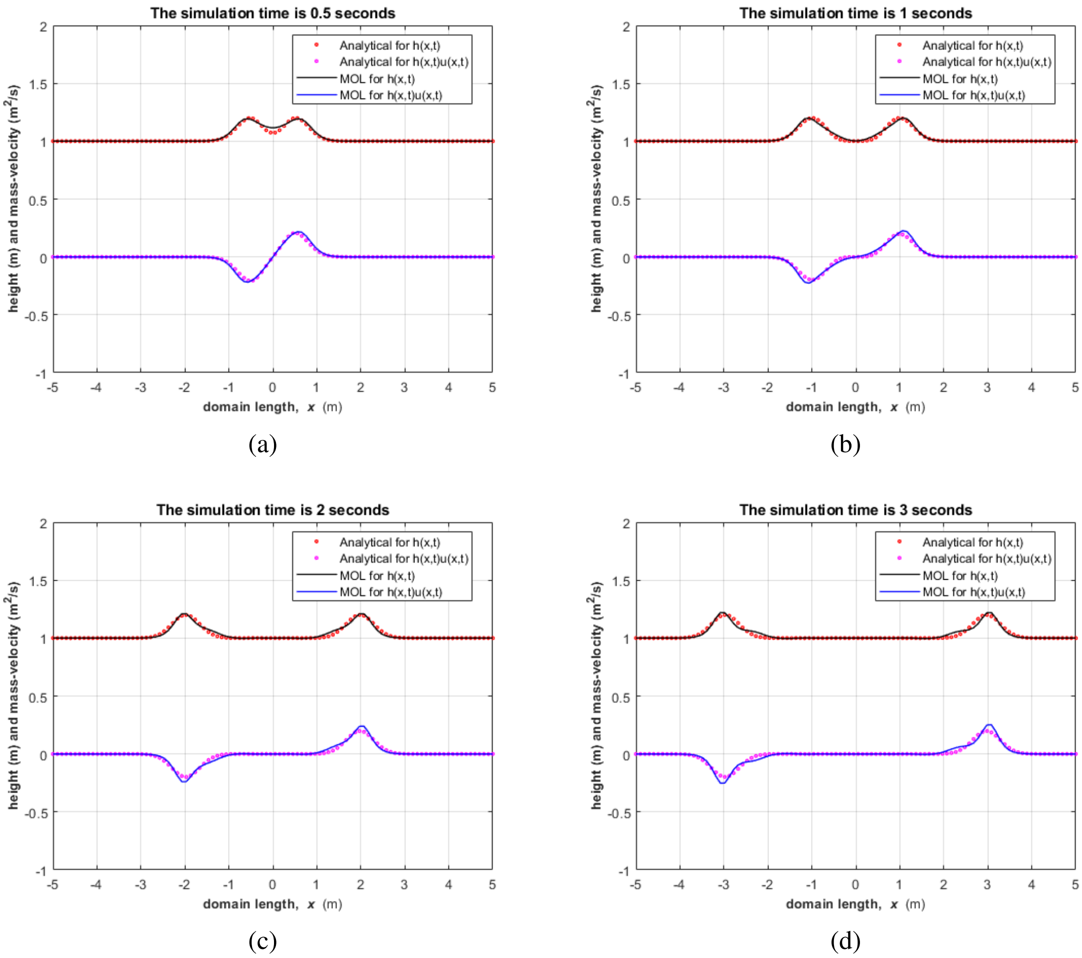

Figure 12. Results of MOL compared with the analytical for the height and mass velocity: (a) At time $t = 0.5s$; (b) At time $t = 1s$; (c) At time $t = 2s$; (d) At time $t = 3s$.

The results of MOL are similar to the analytical solution, but we can see that the waves of height and mass velocity are not smooth in the last two seconds, and the minimum and maximum points in the waves of mass velocity in the third seconds are not exactly the same as those in the analytical solution.

4.6 Comparison between the numerical results

From all the numerical results, we can see that generally, the schemes are accurate, but in the last two seconds, there are limitations, especially at the top and bottom of the waves dome. Thus, in this section, we will present the comparisons for the numerical results of all the numerical schemes and the analytical solution for the last two seconds and at specific positions to see which method is more accurate.

From the behavior of the solution, the height wave has two maximum points, and the mass velocity wave has maximum and minimum points. The comparison between the numerical methods and the analytical solution will be based on these four points with their locations.

Table 1. Comparison of different numerical results for specific points at a time $t = 2s$

	Height h $(meter)$				Mass velocity hu $(meter^2/seconds)$			
	Max 1	L Max 2	Max 1	L Max 2	Max	L Max	Min	L Min
Exact	1.1996	-2	1.1996	2	0.1976	2	-0.1976	-2
BE	1.2096	-2.1	1.2096	2.1	0.2391	2.1	-0.2391	-2.1
FE	1.2168	-2.0707	1.2168	2.0707	0.2484	2.0707	-0.2484	-2.0707
CN	1.215	-2.0707	1.215	2.2727	0.246	2.0707	-0.246	-2.0707
MOL	1.2097	-2.0707	1.2097	1.0707	0.2392	2.0707	-0.2392	2.0707

where: L max 1 and L Max 2 are the locations of the first and the second maximum points of the height wave respectively. L Max and L Min are the locations of the maximum and the minimum points of the mass velocity wave respectively.

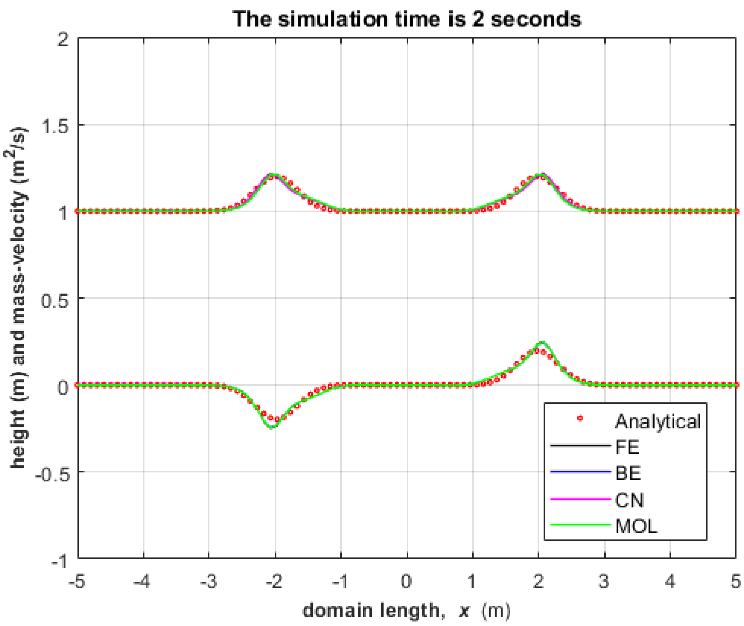

Figure 13. Comparison of different numerical results at a time $t = 2s$

From the above table, one case see that at the time of t = 2s, the height and mass velocity of the Backward Euler scheme (BE) are closer to the exact one as values, but the Method of Lines (MOL) has a position close to the analytical one. Although the results for the BE and MOL are very close to each other.

Table 2. Comparison of different numerical results for specific points at a time $t = 3s$

	Height h $(meter)$				Mass velocity hu $(meter^2/seconds)$			
	Max 1	L Max 1	Max 2	L Max 2	Max	L Max	Min	L Min
Exact	1.1991	-3	1.1991	3	0.1971	3	-0.1971	-3
BE	1.2182	-3	1.2182	3	0.2503	3	-0.2503	-3
FE	1.2257	-3.0808	1.2257	3.0808	0.2599	3.0808	-0.2599	-3.0808
CN	1.2228	-3.0808	1.2228	3.0808	0.2562	3.0808	-0.2562	-3.0808
MOL	1.2198	-3.0808	1.2198	3.0808	0.2522	3.0808	-0.2522	-3.0808

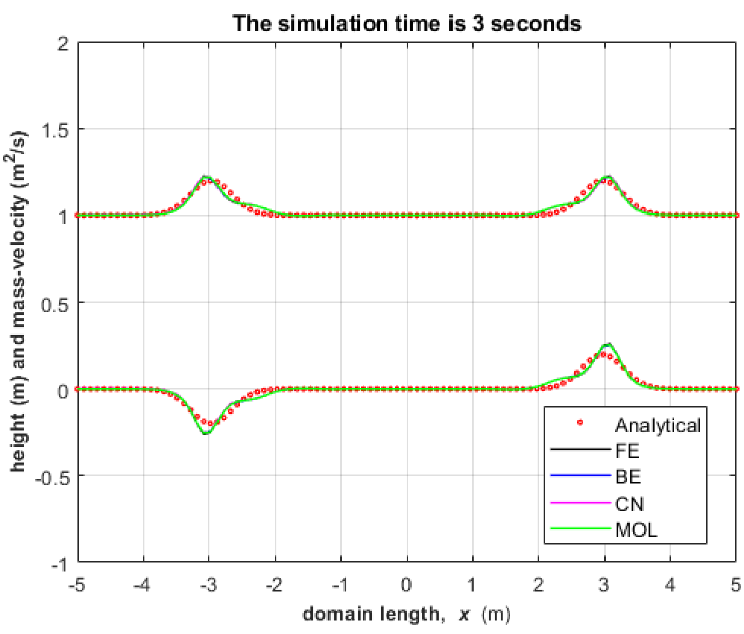

Figure 14. Comparison of different numerical results at a time $t = 3s$

The values and positions of the maximum and minimum in the results of the BE for the height and mass velocity in the third seconds are still closer to the analytical solution. Thus, from the values and plots of the numerical methods, we can generalize that the results of the BE and MOL results are closer to the analytical solution in the second and the third seconds. However, all numerical methods have good accuracy in comparison with the analytical solution.

CPU time:

We also need to consider CPU time of the scheme took to evaluate the numerical results. From Table 3, one can see that the Crank-Nicolson scheme has the height CPU time, and the Method of Lines is faster than the other numerical methods. The running time values are shown in Table 3 below.

Table 3. Comparison of CPU times for all numerical methods

	BE	FE	CN	MOL
CPU time in seconds	13.8254	20.1521	24.1790	3.3507

Finally, we can say that the result of all the numerical methods was good in terms of accuracy. The results from the BE and MOL schemes were better than than the results of the FE and CN schemes, However the MOL scheme showed a smallest running time.

4.7 Future work

This work only takes shallow water model into consideration in one dimension (1D) and without rotation, applying uniform mesh refinement. Further work would include local mesh refinement, expanding the numerical methods to 2D and ultimately 3D with rotation, and applying mesh refinement to each.

5 CONCLUSION

In this work, the system of non-linear Shallow Water Equations (SWEs) in one dimension was derived with the following assumptions; free water surface without rotation, vanishing viscosity, presence of hydrostatic pressure, and flat bottom topography with zero height. Using the Finite Difference approach, the system was solved numerically by testing a number of schemes, such as Forward Euler, Backward Euler, and Crank-Nicolson schemes for the time discretization. The Central Difference scheme was always used for the spatial discretization. In addition, the Method of Line (MOL) approach was also employed for solving the system. The discretizations gave a system of difference equations (algebraic equations) which was easy to solve numerically.

As the flow of the Shallow Water model moves in two directions (left and right) at the same time, it makes this a special case of one-dimensional flow. So the Method of Lines with a fifth-degree polynomial for the discretization of space, and together with the impact of the boundaries conditions, the scheme showed some limitations and instability in the results. Therefor, the Central Difference approximation was used for the discretization of space. In addition, the Forward Euler scheme, which is an explicit scheme was faced with the problem of the Courant number ($\frac{\Delta t}{\Delta x}u < 1$).

The numerical results produced by the tested schemes have been compared to the analytical solution. Generally, good accuracies have been found when the time is equal to 0, 1, and 1.5 seconds. However, as time increases to 2 and 3 seconds, there was a discrepancy between numerical and analytical results. The Backward Euler and the Method of Lines schemes showed better results than the Forward Euler and Crank-Nicolson schemes. Generally, all the numerical methods had results that were very close to each other in terms of accuracy, but the Method of Lines was faster in terms of CPU time.

REFERENCES

[1] Willi H.Hager Oscar Castro. *Shallow Water Hydraulics*. Springer, 2019.

[2] M.T.Capilla and A.Balaguer-Beser. A new well-balanced non-oscillatory water equations on rectangular meshes. *Journal of Computational and Applied Mathematics*, 27:62–74, 2013.

[3] Brigitte Lncquin Herve Le Dret. *Partial differential equations: Modeling, Analysis and Numerical Approximation*. International Series of Numerical Mathematics (ISNM), 2010.

[4] Hildebrand. *Introduction to Numerical Analysis*. 1974.

[5] Leader and J.Jeffery. *Numerical Analysis and Scientific Computation*. 2004.

[6] A.I.Delis and Th.Katsaounis. Numerical solution of the two-dimensional shallow water equations by the application of relaxation methods. *Applied Mathematical Modelling*, 27:754–783, 2005.

[7] Fayssal Benkhaldoun and Mohammed Seaïd. A simple finite volume method for the shallow water equations. *Journal of Computational and Applied Mathematics*, 27:58–72, 2012.

[8] S.M.Amiri, N.Talebbeydokhti, and A.Baghlani. A two-dimensional well-balanced numerical model for shallow water equations. *Scientia Iranica*, 27:97–107, 2013.

[9] A.Emad, M.Marashdeh, and Kamal Al Dawoud. Numerical simulation of one-dimensional shallow water. *International Journal of Sciences: Basic and Applied Research (IJSBAR)*, 23:196–203, 2015.

[10] J.Shipton, T.H.Gibson, and C.J.Cotte. Higher-order compatible finite element schemes for the nonlinear rotating shallow water equations on the sphere. *Journal of Computational Physics*, 27, 2018.

[11] David L.George. Numerical approximation of the nonlinear shallow water equations with topography and dry beds:a godunov-type scheme.

[12] Numerical Approximation of Non-linear Shallow Water Equations, AIMS African Institute for Mathematical Sciences (AIMS) , Nosiba Elfadil, Rwanda, 2019.

[13] Clancy. Aerodynamics. *Pitman Publishing Limited*, 27, 1975.

[14] Pedlosky and Joseph. Geophysical fluid dynamics. *Springer*, 27:10–13, 1987.

[15] Charles B. Jr. Protter, Murray H. Morrey. *Differentiation under the Integral Sign: Intermediate Calculus*. Springer, 1985.

[16] Randall LeVeque. *An Introduction to Numerical Modelling of the Atmosphere*. Cambridge Texts in Applied Mathematics, 2004.

[17] David A.Randall. *Finite Volume Methods for Hyperbolic Problems*.

[18] C.G.Mingham and D.M.Causon. *Introductory Finite Difference Methods for PDEs*.

[19] Wynand S.Verwoerd. *Matrix Methods and Differential Equations: A Practical Introduction*. 2015.

[20] J.; Nicolsoni Crank. A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type. *Proc. Camb. Phil. Soc*, 43:50–67, 1947.

[21] Tuncer Cebeci. Convective heat transfer. *Springer*.

[22] Mincheal Schafer. *Computational Engineering-Introduction to Numerical Methods*. Springer, 2006.

[23] W. E.; G. W. Griffithse Schiesser. *A Compendium of Partial Differential Equation Models: Method of Lines Analysis with Matlab*. Cambridge University Press, 2009.

[24] Hans Petter Langtangen. Svein Linge. *Finite Difference computing with PDEs: A Modern Software Approach*. Springer, 2010.