A Semi-implicit Smoothed Particle Hydrodynamics Method for the Numerical Simulation of Shallow Water Flows

Dissertation with the aim of achieving a doctoral degree
at the Faculty of Mathematics, Informatics and Natural Sciences
Department of Mathematics
of Universität Hamburg

submitted by
Adeleke Olusegun Bankole
from Ibadan, Nigeria

Hamburg
2016

Day of oral defence: February 1, 2017

The following evaluators recommend
the admission of the dissertation

Prof. Dr. Armin Iske
Department of Mathematics
University of Hamburg
Hamburg, Germany

Prof. Dr. Marc Alexander Schweitzer
Institute for Numerical Simulation
University of Bonn
Bonn, Germany

Prof. Dr. Bernd Siebert
Head of Department of Mathematics

Declaration on Oath

Hiermit erkläre ich an Eides statt, dass ich die vorliegende Dissertationsschrift selbst verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

I hereby declare, on oath, that I have written the present dissertation by my own and have not used other than the acknowledged resources and aids.

Hamburg, den 31. October 2016 signature

Abstract

This work focuses on the development of a new semi-implicit SPH scheme for the shallow water equations, following the semi-implicit finite volume and finite difference approach of Casulli [25]. The numerical solution of the shallow water equations on a staggered particle configuration has been derived and discussed.

In standard explicit numerical methods, there is often a severe limitation on the time step due to the stability restriction imposed by the CFL condition. This thesis proposes, a new semi-implicit SPH scheme, which leads to an unconditionally stable method. To this end, the discrete momentum equation is substituted into the discrete continuity equation to obtain a symmetric positive definite linear system for the free surface elevation. The resulting system is sparse which can easily be solved by a matrix-free conjugate gradient method. Once the new free surface location is known, the velocity at the new time level can directly be computed and the particle positions can subsequently be updated. This staggered semi-implicit SPH method stands out from the existing SPH schemes, the staggered approach makes the resulting system for the free surface elevation sparse.

This work involves wetting and drying, this phenomena is treated by the nonlinear algorithm proposed by Casulli [26]. We derive a mildly nonlinear system for the discrete free surface elevation from the shallow water equations by taking into consideration a correct mass balance in wet regions and in transition regions, i.e., the regions from wet particles to dry particles and those from dry particles to wet particles. Unlike in other approaches, our algorithm does not place screens or threshold values at some points to deal with the treatment of wetting and drying. Simple and yet non-trivial 1D, 2D and wetting/drying test problems for the shallow water equation are presented to validate the method. Comparisons have been made in all the cases considered with very reliable numerical reference solutions.

Zusammenfassung

In der vorliegenden Arbeit befassen wir uns mit der Entwicklung eines neuen semi-impliziten SPH (smoothed particle hydrodynamics) Verfahrens zur numerischen Lösung der Flachwassergleichungen. Dabei verwenden wir den semi-impliziten Finite Volumen und Finite Differenzen Ansatz von Casulli [25]. Die numerische Lösung auf einer verteilten Partikelkonfiguration wird hergeleitet und diskutiert.

Für klassische explizite numerische Verfahren erfordert die CFL-Stabilitätsbedingung häufig erhebliche Einschränkungen bei der Wahl des Zeitschrittes. Das in dieser Arbeit entwickelte semi-implizite SPH-Verfahren ist unbedingt (d.h. unabhängig von der Größe des Zeitschrittes) stabil. Hierzu wird die diskrete Impulsgleichung in die diskrete Kontinuitätsgleichung eingesetzt. Dies führt zu einem symmetrischen, positiv definiten linearen System für die freie Oberfläche. Das resultierende System ist dünn besetzt und kann effizient mit einem matrixfreien CG-Verfahren gelöst werden. Sobald die neue freie Oberfläche bekannt ist, kann die Geschwindigkeit zum neuen Zeitpunkt direkt berechnet werden. Die Partikelpositionen werden dann entsprechend aktualisiert. Die neuartige semi-implizite Version des hier entwickelten SPH-Verfahrens unterscheidet sich von vorherigen Varianten: mit dem verteilten Ansatz ist das System für die freie Oberfläche dünn besetzt.

Wir befassen uns außerdem mit Simulationen der Überflutung und Austrocknung (wetting and drying) von Regionen. Dies geschieht mithilfe des nichtlinearen Algorithmus von Casulli [26]. Wir leiten eine schwach nichtlineare Bedingung für die diskrete freie Oberfläche her, indem wir die korrekte Massebilanz in gefluteten Regionen und in Übergangsregionen (d.h. in Regionen, in denen ein Übergang von gefluteten zu trocken Partikeln und umgekehrt stattfindet) betrachten. Im Gegensatz zu vorherigen Ansätzen legen wir keine Schwellen beim Übergang von gefluteten zu trockenen Regionen fest. Einfache aber dennoch aussagekräftige Beispiele in 1D und 2D demonstrieren die Leistungsfähigkeit des hier entwickelten Verfahrens. Die resultierenden Algorithmen werden schließlich durch Vergleiche mit zuverlässigen Referenzlösungen validiert.

Acknowledgements

I am most grateful to God who has been my source of hope, strength, care, provision, protection and his unfailing love throughout my doctoral program. I really thank Him and praise be to His holy name.

I cannot but heartily appreciate my supervisor: Prof. Dr. Armin Iske, I appreciate your unflinching support, understanding heart, encouragement and opportunity to attend local/international workshops/conferences all these enabled me to perfectly complete this thesis. Prof. Dr. -Ing. Thomas Rung, I appreciate your supervision, corrections and your view from the engineering point of view. I am very grateful to Prof. Dr. -Ing. Michael Dumbser (University of Trento, Italy) for our inspiring collaboration over the past years.

Special thanks goes to the Lothar Collatz Graduate School (LCGS) for this wonderful initiative which I am a beneficiary of. Many thanks to all my Professors from the Department of Mathematics: Prof. Dr. Jens Struckmeier, Prof. Dr. Ingenuin Gasser from which I have benefited greatly from your wealth of knowledge and experience. Huge thanks goes to Mrs Astrid Benz, Mrs Birgit Mehrabadi, Mrs Katrin Kopp, Mrs Ruth Ellebracht and Mrs Birgit Wilcke, you are awesome.

I appreciate all my colleagues, fellow PhD students, my fantastic working group including former working group members. Many thanks goes to Dr. Claus Goetz, Dr. Stefan Heitmann, Marzia Leonardi, Fatemeh Zendehrouh that helped in partially and completely proofreading this thesis.

Lastly, a river that forgets its source will eventually run dry, this thanks goes to the two of the finest people who ever graced this world my parents; Pastor Emmanuel Bankole and Mrs Rachael Bankole, without them I would not have been this far, my siblings (Bunmi, Yinka, Yemi and Sola) for their constant prayers and encouragement, Mr. Ernest Itsuokor, my sure friend Aremu Oluwadamilare Ayopo, thank you for your prayers and encouragement. I am saying thank you, I love you all.

Contents

Abstract iii

Zusammenfassung iv

1 Introduction 1

1.1 Motivation for this Thesis . . . 1

2 Preliminaries 4

2.1 Mathematical Models for Fluid Flow . . . 4

2.2 Hyperbolic Conservation Laws . . . 8

2.3 Matrix Properties and Concepts . . . 12

2.4 Some Meshfree Particle Methods . . . 13

3 Shallow Water Equations 19

3.1 Shallow Water Flows . . . 19

3.1.1 Hydrostatic Pressure Distribution . . . 21

3.2 Derivation of the SWE . . . 21

3.3 Characteristics of the SWE . . . 24

3.3.1 Hyperbolicity . . . 24

3.3.2 Eigenstructure in Terms of Physical Variables . . . 25

3.3.3 Riemann Invariants . . . 31

4 Smoothed Particle Hydrodynamics Method 35

4.1 Introduction to SPH . . . 35

4.2 Formulations of SPH . . . 35

4.2.1 Kernel Approximation . . . 35

4.2.2 Particle Approximation . . . 37

4.3 Smoothing Functions . . . 43

5 Semi-implicit SPH Method for Shallow Water Flows 55

5.1 Introduction . . . 55

5.2 Derivation of the SISPH Method . . . 60

5.3 Some Numerical Aspects . . . 68
5.3.1 Boundary Conditions . . . 68
5.3.2 Time Integration . . . 69
5.3.3 Nearest Neighbor Search . . . 70
5.4 The Conjugate Gradient Algorithm . . . 72
5.4.1 The Matrix-free Conjugate Gradient Algorithm . . . 75

6 Wetting and Drying Semi-implicit SPH Methodology for Shallow Water Flows 77
6.1 Introduction . . . 77
6.2 Wetting and Drying Methodology . . . 78
6.2.1 Subparticle Modeling . . . 78
6.2.2 The Free Surface Equation and Mass Conservation . . . 79
6.3 Solution Algorithm . . . 80
6.3.1 Mildly Nonlinear System . . . 80
6.3.2 A Newton Method . . . 83

7 Numerical Examples 85
7.1 The 1D and 2D Shallow Water Equations . . . 85
7.1.1 Smooth Surface Wave Solution . . . 85
7.1.2 Discontinuous Solution . . . 87
7.1.3 An Oscillating Lake Problem . . . 90
7.1.4 A Collapsing Gaussian Bump . . . 92

8 Conclusions and Outlook 98
8.1 Conclusion . . . 98
8.2 Outlook . . . 99

Bibliography 108

Curriculum Vitae 109

List of Figures

2.1 Time evolution of the domain $\Omega(0)$ . . . 6

2.2 Characteristic fields for the two-dimensional shallow water equations. . . . 10

3.1 Flow Domain . . . 19

3.2 Coordinate convention for flow with free surface under gravity . . . 20

4.1 2D particle discretization under gravity . . . 35

4.2 Illustration of the kernel function W, its support domain Ω, with boundary S . . . 38

4.3 Cubic Spline Kernel Function and its First Derivative . . . 45

4.4 SPH behaviour at and near the boundaries. . . . 49

5.1 Sketch of the flow domain: the free surface (light) and the bottom bathymetry (thick). . . 62

5.2 Staggered velocity defined at the midpoint of two pair of interacting particles i and j. . . 64

5.3 Characteristic cone for the two-dimensional shallow water equations. . . . 66

5.4 Pairwise Search Technique . . . 71

5.5 Fictitious Cartesian Grid: Book-keeping Technique . . . 72

6.1 Wetting and drying hydraulic pattern . . . 78

6.2 Non-differentiability of water volume . . . 82

7.1 SISPH scheme solution for smooth surface wave . . . 86

7.2 Smooth surface wave: Larger timesteps . . . 86

7.3 Evolution of Mass Conservation in the Smooth Case . . . 87

7.4 SISPH scheme solution for a discontinuous data . . . 88

7.5 Rarefaction waves: Larger timesteps . . . 89

7.6 Evolution of Mass conservation in the Discontinuous case . . . 89

7.7 An oscillating lake . . . 90

7.8 SISPH wetting and drying solution for an oscillating lake problem . . . 91

7.9 Evolution of Mass conservation with wetting and drying . . . 92

7.10 SISPH three-dimensional view of the smooth surface wave propagation example . . . 94

7.11 SISPH planar view of the smooth surface wave propagation example . . . 95

7.12 Cross section of SISPH solution . . . 96

7.13 Cross section of SISPH solution with $195,496$ particles 97

1. Introduction

1.1 Motivation for this Thesis

The mathematical models describing real life phenomena i.e., physical, biological, technological are often proposed to solve these complex processes. These models usually consists of *partial differential equations* (PDEs) which describe the problems. With the advent of technological advancement, the increase in scale of problems; analytical solutions for partial differential equations for particular cases such as very few number of degrees of freedom, simplified models with easy geometry can be computed. Therefore, there is the need to build a numerical technique for solving and better understanding of fluid flow phenomena. Computational fluid dynamics (CFD) problems are often modeled by a class of PDEs called *hyperbolic conservation laws* e.g. the shallow water equations (SWEs), Euler equations of gas dynamics. They possess different mathematical structures as compared to *parabolic* or *elliptic equations.* Conventionally, numerical techniques for solving PDEs are mainly categorized into three classes namely: Finite Difference Methods (FDM), this method approximates the solutions at grid points of the computational domain, these grid points are connected by a mesh. Finite Volume Methods (FVM), see [43, 49, 69, 74, 75] this method uses an integral formulation of the PDEs which relies on the tessellation of the computational spatial domain defined by control volumes, FVM uses cell averages rather than point values as the case in FDM. Finite Element Methods (FEM) see [20, 38] is a variational formulation of the PDEs but relies on the usage of suitable test functions. However, the above mentioned numerical techniques require a mesh for computation; this property mainly characterize these methods. But they possess some disadvantages when applied to problems with complicated geometry, large deformations, multiphase flows and problems with fragmentation. They are often plagued with the problem of maintaining the grid and constructing them, this can become very computationally costly especially for problems with multivariate approximations. In the same same spirit, the time incurred in meshing and remeshing of time-dependent computational domains become expensive and demanding.

Since, we are interested in solving some practically relevant problems but the above mentioned problems have prohibited the application of gridbased methods. Therefore, the development of *meshfree methods* which allows the solution of practically industrial problems have been of interest recently. The family of meshfree methods possesses some notable advantages namely: they can easily treat simulations of very large deformations [36, 76], this is because the nodes are an integral part of the computation and they vary with time; they can easily adapt to the varying topological structure of the continuum e.g. in problems such as underwater explosions, crack growth [15], these methods can also incorporate a priori knowledge about the local behavior of the solution in the interpolation space, in any dimension they can provide smooth higher order interpolation field, they provide an accurate representation of geometrical objects [78], they possess a good computational paradigm for multiscale problems because of there non-local interpolation property and they support flexible refinement procedure, because particles can be added where refinement is needed and vice-versa.

Meshfree particle methods are methods that need no mesh for the computation of field variables, they instead use particles for the discretization of the equations. Examples include: the Smoothed Particle Hydrodynamics method (SPH), the Finite Volume Particle Method (FVPM), Finite Point Mass Method (FPM), the Meshfree Galerkin Methods. The *Smoothed Particle Hydrodynamics* (SPH) considered as a truly meshfree approach have been designed to overcome the above mentioned drawbacks. SPH was developed by Lucy [83], Gingold and Monaghan [48] to simulate astrophysical and cosmological problems. Over the years, the SPH method has grown and was applied to model structures in the

80's and then to *free-surface* flows [89] in the 90's and consequently, in the 2000's, the method was successfully used to study fluid dynamics problems. Its wide range of applications includes geotechnical engineering problems, free-surface flows, marine problems, reservoir flushing, landslide problems see [84, 89, 95, 96, 111], and even to port hydrodynamic simulations [109]. The SPH method; with a Lagrangian description discretizes the fluid domain by a finite number of particles which can move. The particles move with the velocity of the fluid. The basic idea behind SPH is that flow quantities are smoothed by a kernel function with respect to the measure that is associated with the mass density of the flow. Flow quantities at a point are approximated as a weighted average around the set of nearby particles.

In this thesis, the *Semi-implicit Smoothed Particle Hydrodynamics* (SISPH) will be designed, developed and analysed. Our SISPH scheme follows from the semi-implicit finite volume and finite difference approach of Casulli [25]. In numerical methods, explicit and implicit schemes exists but a major problem of explicit schemes in numerical methods is their severe time step restriction, where the Courant-Friedrichs-Lewy (CFL) condition imposes the time step size in terms of the wave propagation speed and the mesh size. Hence, the major advantage of a semi-implicit approach is that stable schemes are obtained which allow large time step sizes at a reasonable computational cost. We have considered in this thesis flows which are governed by the *shallow water equations* which we can derive from the three dimensional Navier-Stokes equations with the assumption of a hydrostatic pressure distribution, see [28, 29]. In a staggered mesh-based approach for finite differences and volumes, discrete variables are often defined at different (staggered) locations. The pressure term, which is the free surface elevation is defined in the cell center while the velocity components are defined at the cell interfaces. In the momentum equation, pressure terms are due to the gradients in the free surface elevations and the velocity in the mass equation (i.e., free surface equation) are both discretized implicitly whereas the nonlinear convective terms are discretized explicitly. The semi-Lagrangian method is one of the techniques to discretize the convective terms explicitly (see [18, 56, 73]). Recently, the semi-implicit SPH scheme for the one dimensional shallow water model has been discussed and validated on some one dimensional test examples by Bankole et al. (see [7]) for details.

In recent years, some authors have worked on a semi-implicit method for particle methods. In the specific, Koshizuka and Oka [66, 67] presented the moving-particle semi-implicit method (MPS) where a deterministic interaction models for the gradient, Laplacian operators and free surfaces are presented. Incompressibility condition is imposed by setting the rate of change of density with time to zero at each time step, likewise a modified kernel function which has a unique property that the value of the kernel goes to infinity as distance between particles tends to zero; a kernel function which has been validated to avoid particle clumping. Ataie-Ashtiani and Farhadi [3, 4] worked in the same direction and presented a stable MPS method for free surface flows using a fractional step idea of discretization to split the time step into two steps. A number of authors modified, extended and improved on the MPS method of Koshika and Oka (see [62, 63, 64, 65, 119]) even more for the enhancement of performance, stability and accuracy of the MPS method.

In this thesis, the new semi-implicit *Smoothed Particle Hydrodynamics* (SPH) scheme presented by Bankole et al. [7] for the numerical solution of the shallow water equations will be extended to the shallow water equations on 2D particle configuration is therefore proposed, derived and discussed. The flow variables in this thesis are the particle free surface elevation, particle total water depth and the particle velocity. The discrete momentum equations are substituted into the discretized mass conservation equation to give a discrete equation for the free surface leading to a system in only one single scalar quantity, the free surface elevation location. The system is solved for each time step as a linear algebraic system. The components of the momentum equation at the new time level can be directly computed from the new free surface. This can be conveniently solved by a matrix-free version of the conjugate gradient (CG) algorithm [99]. Consequently, the particle velocities at the new time level are computed and the particle positions are updated. In this semi-implicit SPH method, the stability is independent of the wave celerity. Hence, a relatively large time steps can be permitted to enhance the numerical efficiency [28]. In this thesis, we have introduced a staggered velocity framework in the sense of meshfree methods. This application of a staggered velocity between particles is one of the novelty in this thesis. The staggeredness of the velocity improves the sparsity of the resulting linear system significantly. Moreover, an integral part of the resulting numerical method, the discrete free-surface equation has been treated to represent the accurate mass balance when wetting and drying are expected. The resulting system is *nonlinear* while a mass conservation and nonnegative water depths are guaranteed everywhere in the flow domain for all time steps. And a few number of iterations are needed to solve the resulting *nonlinear system*.

The thesis is organized as follows: In Chapter 2, we give the preliminaries and basics on the mathematical modeling of fluid flows based on hyperbolic conservation laws. In Chapter 3, we present the theory of shallow water equations, the derivations of the two dimensional shallow water equations and their characteristics is presented. In Chapter 4, we give the fundamental theory of the Smooth Particle Hydrodynamics method, we explain the SPH approximations of the partial differential equations involved and the constitutive models used in particle approximations. In Chapter 5, the key ideas of the proposed semi-implicit SPH scheme will be presented, derived, discussed and analyzed. Chapter 6 presents a nonlinear wetting and drying algorithm applied to the shallow water equation and the solution algorithm used in the sense of an efficient Newton-type algorithm for solving the resulting mildly nonlinear system will be presented and derived. In Chapter 7, we present numerical examples and we validate against very reliable numerical reference solutions. The thesis is rounded up in Chapter 8 with some concluding remarks and outlook for possible future work.

Parts of the results of this thesis have been accepted/published in:

1. A.O. Bankole, M. Dumbser, A. Iske, and T. Rung, *A Meshfree Semi-implicit Smoothed Particle Hydrodynamics Method for Free Surface Flow*: Accepted for publication in: Lecture Notes in Computational Science and Engineering - Meshfree Methods for Partial Differential Equations VIII, M. Griebel and Marc A. Schweitzer, editors, Springer-Verlag, 2017.

2. A.O. Bankole, M. Dumbser, A. Iske, and T. Rung. *A Wetting and Drying Semi-implicit SPH Algorithm for the Shallow Water Equations*. Proceedings of the 11th SPHERIC International Workshop, Munich, Germany, pp. 260–267, 2016.

3. A.O. Bankole, M. Dumbser, A. Iske, and T. Rung. *A Semi-implicit SPH Scheme for the Two-dimensional Shallow Water Equations*. Proceedings of the 10th SPHERIC International Workshop, Parma, Italy, pp. 252–258, 2015.

4. A.O. Bankole, M. Dumbser, A. Iske, and T. Rung. *A Semi-implicit SPH Scheme for the Shallow Water Equations*. Proceedings of the 9th SPHERIC International Workshop, Paris, France, pp. 419–424, 2014.

2. Preliminaries

In this chapter, we present the basic ideas in the mathematical modeling of fluid flows. The laws governing fluid dynamics have been well established and in particular the equations of fluid dynamics describe the motion of a general fluid. The models rely on the conservation of mass and momentum i.e, during the evolution of a fluid, properties such as mass, momentum are not destroyed during the whole process at all times. We will present the key ideas in the theory of hyperbolic conservation laws. To solve our proposed semi-implicit smoothed particle hydrodynamics (SISPH) numerical method in this thesis we need to mention some matrix properties and definitions from numerical linear algebra. We finish this chapter by outlining some meshfree numerical methods available.

2.1 Mathematical Models for Fluid Flow

In applications, we model physical processes mathematically relying and governed by some fundamental principles of conservation. Let us start with defining a particle.

Definition 2.1.1 (Particle). *Let* $\Omega_0 \subset \Omega \subset \mathbb{R}^d$ *be an open and bounded spatial domain, where* d *is the dimension in space. The elements say* $\mathbf{p}$ *in* Ω_0 *i.e.* $\mathbf{p} \in \Omega_0$ *are called particles.*

Let us consider a fluid particle which at time t_0 is located at the position $\boldsymbol{\xi} = (\xi_1, \xi_2, \xi_3)$, the same particle at time t is at position $\mathbf{x} = (x_1, x_2, x_3)$. Without loss of generality, let $t_0 = 0$. The motion of a particle in the time interval $[0, t]$ can be described by the function

$$\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}, t). \tag{2.1}$$

Equation (2.1) at any time t, describes the position in space of a particle that was in $\boldsymbol{\xi}$ at $t = 0$ where $\boldsymbol{\xi}$ is called the *material or Lagrangian coordinates* as a particular value of $\boldsymbol{\xi}$ identifies a material particle that at $t = 0$ was in $\boldsymbol{\xi}$. $\mathbf{x}$ is called the *spatial or Eulerian coordinates* as a particular value of $\mathbf{x}$ identifies a given position in space.

We assume the motion is continuous, for a given time a single particle cannot occupy two different positions and conversely, a single point in space cannot be occupied simultaneously by two particles. By inverting Equation (2.1) we obtain

$$\boldsymbol{\xi} = \boldsymbol{\xi}(\mathbf{x}, t). \tag{2.2}$$

Equation (2.2) gives the initial position (at $t = 0$) of a material particle that at time t is in $\mathbf{x}$. Mathematically inverting (2.1) can be defined as

$$J = \det\left[\frac{\partial(x_1, x_2, x_3)}{\partial(\xi_1, \xi_2, \xi_3)}\right] \tag{2.3}$$

where $J > 0$ is condition of invertibility of the Jacobian (see [1]).

Definition 2.1.2 (Particle Trajectory). *Let* $\mathbf{x}$ *in (2.1) be a parametric equation of a curve in space and* t *as the parameter. The curve that goes through the point* $\boldsymbol{\xi}$ *at* $t = 0$ *is called the particle trajectory or particle path or pathline.*

Definition 2.1.3 (Particle Velocity). *The function*

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}, t), \qquad \mathbf{x}(0) = \boldsymbol{\xi}. \tag{2.4}$$

is the particle velocity field. We obtain particle trajectories of the flow by integration of the velocity field.

Definition 2.1.4 (Steady Flow). *A flow is said to be steady if the velocity field* $\mathbf{v}$ *described by the vector field* $\mathbf{v}(\mathbf{x}, t)$ *does not depend on time.*

Remark 2.1.5 (Steady Flow). *The steadiness of flow does not imply that each material particle has a constant velocity in time as* $\mathbf{v}(\boldsymbol{\xi}, t)$ *might still depend on time.*

Definition 2.1.6 (Streamlines). *For a given velocity field* $\mathbf{v}(\mathbf{x}, t)$*, curves which are at all points in space parallel to the velocity vector are called streamlines. Mathematically, they are defined as*

$$d\mathbf{x} \times \mathbf{v} = 0 \tag{2.5}$$

where $d\mathbf{x}$ is an infinitesimal segment along the streamline

Definition 2.1.7 (Streaklines). *For a given time* t*, a streakline joins all material points that have passed through (or will pass through) a given place* $\mathbf{x}$ *at any time.*

In (2.1), let $\mathbf{x} = \mathbf{x}'$ and $\mathbf{t} = \mathbf{t}'$ represents a material point which was at point $\mathbf{x}'$ at t'. For this particle, the path coordinates are given by

$$\mathbf{x} = \mathbf{x}(\boldsymbol{\xi}(\mathbf{x}', t'), t). \tag{2.6}$$

At a given point t, t' represents the curve parameter of a curve in space which passes through the given point $\mathbf{x}'$. This curve in space is called a *streakline*. In Fig. 2.1 we observe the evolution of a particle in the initial configuration $D(0) \equiv \Omega(0)$ at time $t = 0$ to the current configuration $D(t) \equiv \Omega(t)$ at time t.

Theorem 2.1.8 (Reynold Transport Theorem). *Let* $\mathcal{F}(\mathbf{x}, t)$ *be either scalar or vector valued function, assume* $\mathcal{F} : (0, t) \times \mathbb{R}^d \to \mathbb{R}^d$ *is differentiable, with* d *fixed. Let* $\Omega(t)$ *be a material volume entirely occupied by the fluid. Then*

$$\frac{D}{Dt} \int\limits_{\Omega(t)} \mathcal{F}(\mathbf{x}, t) dV = \int\limits_{\Omega(t)} \left[\frac{\partial \mathcal{F}}{\partial t} + \nabla \cdot (\mathcal{F}\mathbf{v}) \right] dV. \tag{2.7}$$

Equation (2.7) can be formulated as

$$\frac{D}{Dt} \int\limits_{\Omega(t)} \mathcal{F}(\mathbf{x}, t) dV = \int\limits_{\Omega(t)} \frac{\partial \mathcal{F}}{\partial t} dV + \int\limits_{\partial\Omega(t)} \mathcal{F}\mathbf{v} \cdot \mathbf{n} dA$$

$\partial\Omega(t)$ is the bounding surface of the domain $\Omega(t)$ and $\mathbf{n}$ is the outer normal to the surface $\partial\Omega(t)$. It explains that the material derivative of a fluid property $\mathcal{F}$ integrated over a material control volume $\Omega(t)$ can be written as the integral of $\frac{\partial \mathcal{F}}{\partial t}$ over the volume $\Omega(t)$ plus the flux of $\mathcal{F}$ through the bounding surface $\partial\Omega(t)$. We can find the proof in [37]

The Reynold transport theorem formulated in the one-dimensional version is given as

$$\frac{D}{Dt} \int_{a(t)}^{b(t)} f(x, t) dx = \int_{a(t)}^{b(t)} (f_t + (vf)_x) dx \tag{2.8}$$

The velocity field $v(x,t)$ be a given function of position and time, x is a displacement along the line, the interval $(a(t), b(t))$ is a material interval whose endpoints satisfy the ordinary differential equations ODEs $\dot{a} = v(a,t)$, $\dot{b} = v(b,t)$ and $f(x,t)$ is any real-valued function of position and time. To derive equation (2.8), let the antiderivative of f be given as

$$F(x,t) = \int_0^x f(x',t)dx' \tag{2.9}$$

Then we have,

$$\int_{a(t)}^{b(t)} f(x,t)dx = F(b(t),t) - F(a(t),t) \tag{2.10}$$

Differentiating with respect to time t, making use of the chain rule, we have

$$\frac{D}{Dt}\int_{a(t)}^{b(t)} f(x,t)dx = F_t(b,t) - F_t(a,t) + F_x(b,t)\dot{b} - F_x(a,t)\dot{a} \tag{2.11}$$

The time differentiation of equation (2.9) gives

$$F_t(x,t) = \int_0^x f_t(x',t)dx' \tag{2.12}$$

We have

$$F_t(b,t) - F_t(a,t) = \int_0^b f_t(x,t)dx - \int_0^a f_t(x,t)dx = \int_a^b f_t dx \tag{2.13}$$

Then,

$$F_x(b,t)\dot{b} - F_x(a,t)\dot{a} = f(b,t)v(b,t) - f(a,t)v(a,t) \tag{2.14}$$

$$= \int_a^b (vf)_x dx \tag{2.15}$$

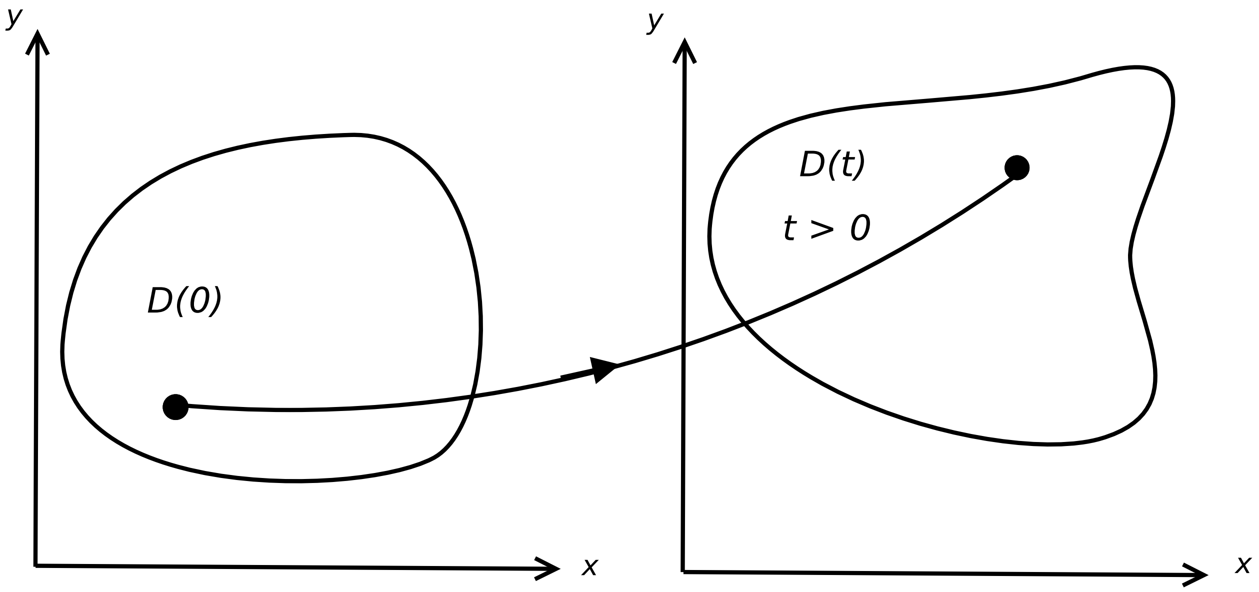


Figure 2.1: Time evolution of the domain D(0)

The first equality makes use of $F_x = f$, $\dot{a} = v(a,t)$ and $\dot{b} = v(b,t)$. The second equality uses the fundamental theorem of calculus. Thus we have the desired result in one dimension

$$\frac{D}{Dt}\int_{a(t)}^{b(t)} f(x,t)dx = \int_{a(t)}^{b(t)} (f_t + (vf)_x)dx \tag{2.16}$$

Conservation of Mass

The principle of conservation of mass imposes that the material derivative of the mass of fluid in a material volume Ω is equal to zero. Conservation of a flow quantity means the quantity is neither created nor destroyed. Let the mass of a fluid particle in V be given as

$$\int\limits_{\Omega(t)} \rho dV. \tag{2.17}$$

Applying the material derivative we have

$$\frac{D}{Dt}\int\limits_{\Omega(t)} \rho dV \tag{2.18}$$

Using Theorem 2.1.8 we have

$$\int\limits_{\Omega(t)} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v})dV = 0. \tag{2.19}$$

Because $\Omega(t)$ is an arbitrary control volume, then the following differential equation holds

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{2.20}$$

This equation (2.20) is called the *continuity equation*. In particular, when the particle density ρ is constant, the fluid is incompressible and (2.20) becomes

$$\nabla \cdot \mathbf{v} = 0 \tag{2.21}$$

This explains that the velocity field of an incompressible fluid is divergence free.

Conservation of Momentum

Newton's law states that the material derivative of the momentum of a fluid in $\Omega(t)$ is equal to sum of all external forces acting on the volume i.e., the temporal change of momentum is equal to all resultant of all acting forces. Momentum is defined as the product of mass and velocity by

$$\int\limits_{\Omega(t)} \rho \mathbf{v} dV. \tag{2.22}$$

Therefore we have:

$$\frac{D}{Dt}\int\limits_{\Omega(t)} \rho \mathbf{v} dV = \int\limits_{\Omega(t)} \rho \mathbf{f} dV + \int\limits_{\partial\Omega(t)} \mathbf{t}(\mathbf{n}) dS. \tag{2.23}$$

where $\mathbf{t}(\mathbf{n}) = \boldsymbol{\sigma}\mathbf{n}$. Applying Theorem 2.1.8 and the Gauss theorem we obtain

$$\int\limits_{\Omega(t)} \rho \frac{D\mathbf{v}}{Dt} - \rho\mathbf{f} - \nabla \cdot \boldsymbol{\sigma} dV = 0. \tag{2.24}$$

Because $\Omega(t)$ is an arbitrary control volume the following differential equation must hold

$$\rho \frac{D\mathbf{v}}{Dt} - \rho\mathbf{f} - \nabla \cdot \boldsymbol{\sigma} = 0, \tag{2.25}$$

which in spatial coordinates is given as

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v} \cdot \nabla)\mathbf{v} - \rho\mathbf{f} - \nabla \cdot \boldsymbol{\sigma} = 0. \tag{2.26}$$

$\boldsymbol{\sigma}$ is called the *Cauchy stress tensor* which also determines the character of the equations either viscous or nonviscous. Equation (2.26) holds for any continuum.

Having outlined the mathematical models for fluid flow and the fundamental laws governing the flow of fluid, we proceed by discussing the concept of hyperbolic conservation laws.

2.2 Hyperbolic Conservation Laws

In this section, we explain the basic properties of hyperbolic conservation laws and their solutions. The so called shallow water equations belongs to this class of partial differential equations, to mention a few of them: the Euler equations of gas dynamics, the magneto-hydrodynamics equations (MHD) are all of hyperbolic structure (see, e.g., [69, 74, 75] for more details). We will start with the definition of conservation laws.

Definition 2.2.1 (**Conservation Laws**). *A system of partial differential equations of the form*

$$\partial_t \mathbf{v} + \nabla \cdot \mathbf{F}(\mathbf{v}) = \mathbf{0} \qquad \forall \mathbf{x} \in \mathbb{R}^d, \quad t > 0 \tag{2.27}$$

with the initial conditions

$$\mathbf{v}(\mathbf{x}, 0) = \mathbf{v}_0(\mathbf{x}) \qquad \forall \mathbf{x} \in \mathbb{R}^d \tag{2.28}$$

where $\mathbf{v} : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^m$ *i.e.* $\mathbf{v} = (v_1, \ldots, v_m)^T \in \mathbb{R}^m$, $\mathbf{F} = (\mathbf{f}_1, \ldots, \mathbf{f}_d) : \mathbb{R}^m \to \mathbb{R}^{m \times d}$, $\mathbf{v}_0 \in \mathbf{L}^\infty(\mathbb{R}^d)$ *is called a system of conservation laws in* d *spatial dimensions*

$\mathbf{v}$ is the vector of conserved variables, $\mathbf{F}$ denotes the flux vector function, the divergence operator $\nabla \cdot \mathbf{F}(\mathbf{v})$ is given as $\sum_{j=1}^d \partial_{x_j} \mathbf{f}_j(\mathbf{v})$. By conservation, we mean that the flow quantity $\mathbf{v}$ is conserved in the control volume $\Omega(t)$ up to the flux $\mathbf{v}$ through the bounding surface $\partial\Omega(t)$. Integrating equation (2.27) and applying the divergence theorem we obtain

$$\int\limits_{\Omega(t)} \mathbf{v} dV + \sum_{j=1}^{d} \int\limits_{\partial\Omega(t)} \mathbf{f}_j(\mathbf{v}) \mathbf{t}_j(\mathbf{n}) dS = \mathbf{0}. \tag{2.29}$$

Basically, this equation explains that $\int_\Omega \mathbf{v} dV$ changes in time only due to the flux of $\mathbf{v}$ across the bounding surface $\partial\Omega(t)$.

Remark 2.2.2 (**Balance Laws**). *When the right hand side of (2.27) does not vanish, we say the partial differential equation is a balance law. It takes the form*

$$\partial_t \mathbf{v} + \nabla \cdot \mathbf{F}(\mathbf{v}) = \mathbf{S}(\mathbf{v}) \qquad \forall \mathbf{x} \in \mathbb{R}^d, \quad t > 0 \tag{2.30}$$

the vector function $\mathbf{S}(\mathbf{v})$ *is called the source term.*

Definition 2.2.3 (**Hyperbolicity**). *A system of conservation laws of the form (2.27) is called (strictly) hyperbolic if for any vector of conserved quantity* $\mathbf{v} \in \mathbb{R}^d$ *and* $\mathbf{n} = (n_1, \dots, n_d)^T \in \mathbb{R}^d$, *where* $\mathbf{n} \neq \mathbf{0}$, *the Jacobian matrix*

$$\mathbf{A}(\mathbf{v}, \mathbf{n}) = \sum_{j=1}^{d} n_j D\mathbf{f}_j(\mathbf{v}) \tag{2.31}$$

has eigenvalues $\lambda_1, \dots, \lambda_m$ *that are real and distinct with* m *linearly independent right eigenvectors* $\mathbf{R}^{(i)}$.

A major property of hyperbolic equations is that information propagates at a finite speed by the eigenvalues. For a hyperbolic system, the *characteristic speed* $\lambda_i(\mathbf{v})$ defines the *characteristic field*, called the λ_i-field. We also speak of the $\mathbf{R}_i$-field simply called the i-field. Let $\nabla\lambda_i(\mathbf{v})$ be the gradient of the eigenvalue $\lambda_i(\mathbf{v})$.

Definition 2.2.4 (**Linearly Degenerate**). *A* λ_i*- characteristic field is said to be linearly degenerate if*

$$\nabla\lambda_i(\mathbf{v}) \cdot \mathbf{R}^{(i)}(\mathbf{v}) = 0, \qquad \mathbf{v} \in \mathbb{R}^m \tag{2.32}$$

Definition 2.2.5 (**Genuinely Non-linear**). *A* λ_i*- characteristic field is said to be genuinely non-linear if*

$$\nabla\lambda_i(\mathbf{v}) \cdot \mathbf{R}^{(i)}(\mathbf{v}) \neq 0, \qquad \mathbf{v} \in \mathbb{R}^m \tag{2.33}$$

where $\mathbb{R}^m$ is the set of real-valued vectors of m components, called the *phase space or state space*. For a $m \times m$ system we speak of the *phase plane*. We proceed by giving an example of a system of hyperbolic conservation laws that will be studied in this thesis. This is the so called *shallow water equations*, these equations arise in the modeling of a wide variety of physical phenomena, such as water flows, atmospheric flows, dense gas dispersion, avalanches and also in astrophysical flows.

Example 2.2.6 (**Shallow water Equations**). *The nonlinear shallow water equations are a system of hyperbolic conservation laws. In two-dimensional case, it reads*

$$\partial_t \mathbf{v} + \nabla \cdot \mathbf{F}(\mathbf{v}) = 0, \tag{2.34}$$

with the vectors of conserved variables given as

$$\mathbf{v} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \qquad \mathbf{f_1} = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix}, \qquad \mathbf{f_2} = \begin{pmatrix} hv \\ hvu \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}, \qquad \mathbf{F} = (\mathbf{f_1}, \mathbf{f_2}),$$

where h *is the depth,* u *and* v *are the* x *and* y *components of the velocity,* g *is the constant of acceleration due to gravity.*

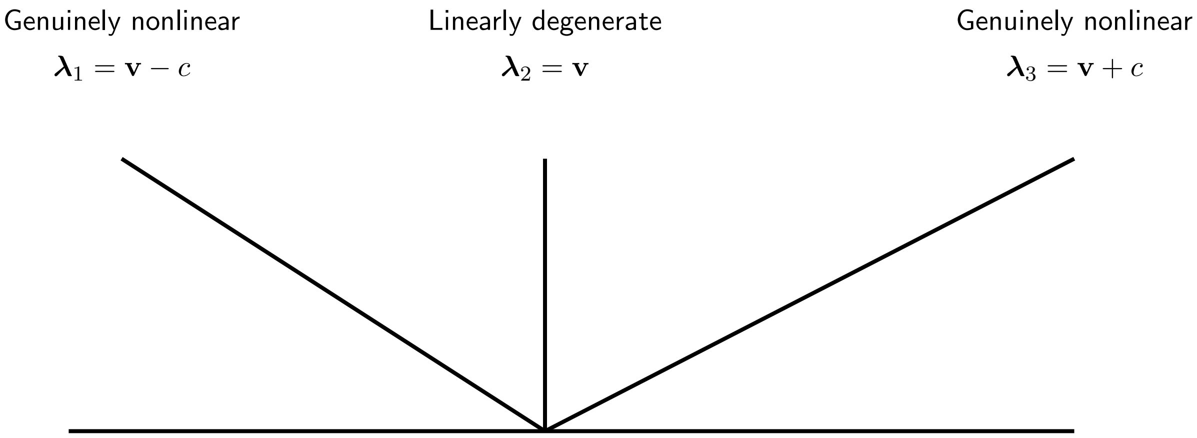

Genuinely nonlinear
$\boldsymbol{\lambda}_1 = \mathbf{v} - c$
Linearly degenerate
$\boldsymbol{\lambda}_2 = \mathbf{v}$
Genuinely nonlinear
$\boldsymbol{\lambda}_3 = \mathbf{v} + c$

Figure 2.2: Characteristic fields for the two-dimensional shallow water equations.

The eigenvalues of the Jacobian matrix $\mathbf{A}(\mathbf{v}, \mathbf{n})$ for the two-dimensional shallow water equations read

$$\lambda_1 = u - c, \qquad \lambda_2 = u, \qquad \lambda_3 = u + c, \tag{2.35}$$

$$\tilde{\lambda_1} = v - c, \qquad \tilde{\lambda_2} = v, \qquad \tilde{\lambda_3} = v + c. \tag{2.36}$$

where the term c is called the *wave celerity* defined by

$$c = \sqrt{gh}. \tag{2.37}$$

Equations (2.35) and (2.36) are the eigenvalues in the x and y directions. Eigenvalues $\lambda_1, \tilde{\lambda_1}, \lambda_3, \tilde{\lambda_3}$ are genuinely non-linear characteristic fields and $\lambda_2, \tilde{\lambda_2}$ are linearly degenerate characteristic fields as seen in Fig. 2.2. Likewise, the eigenvalues are all real, also distinct under all circumstances except for the case when $h = 0$, in which case $c = 0$ then $\lambda_1 = \lambda_2 = \lambda_3 = u$.

Definition 2.2.7 (Classical Solutions). *A function* $\mathbf{v} \in \mathbf{L}^{\infty}(\mathbb{R}^d \times \mathbb{R}_+, \mathbb{R}^m)$ *is said to be a classical solution of the hyperbolic conservation law (2.27) if*

$$\mathbf{v} \in (C^1 \cap \mathbf{L}^{\infty})(\mathbb{R}^d \times \mathbb{R}_+, \mathbb{R}) \tag{2.38}$$

and if $\mathbf{v}$ *satisfies (2.27)* $\forall\ (\mathbf{x}, t) \in \mathbb{R}^d \times \mathbb{R}_+$ *and* $\mathbf{x} \in \mathbb{R}^d$.

In the study of the solution of hyperbolic equations, the *method of characteristics* (MOC) is often employed to analytically solve for instance in a linear one-dimensional sense. However, in using the MOC and even more for smooth initial data $\mathbf{v}_0$, characteristic lines cross. Therefore, the method of characteristics is not always giving a solution, in some cases not for all time t, in some cases only up to a certain critical time and in some cases for all times. Because of this short coming, this will not be a a kind of solution we seek. Since, even smooth initial data can lead to discontinuities, then we delve to a more larger class of solutions called *weak solutions*.

Definition 2.2.8 (Weak Solutions). *A function* $\mathbf{v} \in \mathbf{L}^{\infty}(\mathbb{R}^d \times \mathbb{R}_+, \mathbb{R}^m)$ *is said to be a weak solution of the hyperbolic conservation law (2.27) if*

$$\int_{\mathbb{R}^d} \int_{\mathbb{R}_+} (\mathbf{v}\varphi_t + \mathbf{F}(\mathbf{v})\nabla\varphi) dt d\mathbf{x} + \int_{\mathbb{R}^d} \mathbf{v}_0(\mathbf{x})\varphi(\mathbf{x}, 0) d\mathbf{x} = 0 \tag{2.39}$$

holds for all test functions $\varphi \in C_0^1(\mathbb{R}^d \times \mathbb{R}_+, \mathbb{R})$.

This class of weak solutions does allow discontinuous solutions. The weak solution concept is motivated by the fact that smooth solutions are indeed weak solutions. Equation (2.27) is multiplied by the test function φ and we integrate over $\mathbf{x}$ and t, the third term in (2.39) represents the only boundary term. In general, weak solutions are not unique. To pick a physically meaningful solution from the array of non-physical solutions, the *entropy solution* is used. To pick from the alternative solutions, we need additional conditions which are not included in (2.27). The additional conditions are called *entropy conditions*. This name originates from physical models where additional conditions are based on *physical entropy*. We proceed to give an example of an entropy condition from the ground-breaking work of Lax [71].

Example 2.2.9 (Limit of Small Viscosity). *Equation (2.27) is modified into a viscous problem. We add a small diffusive term in (2.27) as follows*

$$\partial_t \mathbf{v} + \nabla \cdot \mathbf{F}(\mathbf{v}) = \varepsilon \Delta \mathbf{v} \qquad \forall \mathbf{x} \in \mathbb{R}^d, \quad t > 0, \quad , \varepsilon > 0 \tag{2.40}$$

This is a parabolic equation which gives a unique solution $\forall\ \varepsilon > 0$. *We only accept solutions to (2.27) as limits of solutions to the modified viscous problem (2.40) as* $\varepsilon \to 0$ *i.e. the correct physical solution (2.27) should coincide with the parabolic solution (2.40) as* $\varepsilon \to 0$.

Definition 2.2.10 (Entropy Solution). *A function* $\mathbf{v} \in \mathbf{L}^{\infty}(\mathbb{R}^d \times \mathbb{R}_+, \mathbb{R}^m)$ *is said to be an entropy solution of the hyperbolic conservation law (2.27) if*

$$\int_{\mathbb{R}^d} \int_{\mathbb{R}_+} (\mathbf{v}\varphi_t + \mathbf{F}(\mathbf{v})\nabla\varphi) dt d\mathbf{x} + \int_{\mathbb{R}^d} \mathbf{v}_0(\mathbf{x})\varphi(\mathbf{x}, 0) d\mathbf{x} = 0 \tag{2.41}$$

holds for all test functions $\varphi \in C_0^1(\mathbb{R}^d \times \mathbb{R}_+, \mathbb{R})$ *with compact support and if there is a constant* $C \geq 0$ *such that* $\forall\ \mathbf{x} \in \mathbb{R}$, $\mathbf{z} \in \mathbb{R}$, $z \geq 0$, $t \in \mathbb{R}$,$t \geq 0$ *the relation*

$$\mathbf{v}(\mathbf{x} + \mathbf{z}, t) - \mathbf{v}(\mathbf{x}, t) \leq C(1 + \frac{1}{t})\mathbf{z} \tag{2.42}$$

holds almost everywhere in x *and* t.

Definition 2.2.11 (Convexity). *A flux function* $\mathbf{F}$ *is said to be uniformly convex if there exists a constant* $C > 0$ *such that*

$$\mathbf{F}''(\mathbf{v}) \geq C > 0 \tag{2.43}$$

holds for all $\mathbf{v}$.

Theorem 2.2.12. *Let the flux function* $\mathbf{F}$ *be smooth, uniformly convex and let* $\mathbf{v}_0 \in \mathbf{L}^{\infty}(\mathbb{R}^d)$. *Then there exists a unique entropy solution of (2.27).*

In the next section, we give some succinct definitions and properties in matrix theory. The idea and the structure of matrices obtained will be useful in designing our solution strategy.

2.3 Matrix Properties and Concepts

More often in applications, after discretizing partial differential equations matrices eventually end up whose properties and structure must be understood. See the books of Horn, Johnson and Varga [55, 113] for more details.

Definition 2.3.1 (Spectral Radius). *Let $A = [a_{i,j}]$ be an $n \times n$ real matrix with eigenvalues $\lambda_i, 1 \leq i \leq n$. Then,*

$$\rho(A) = \max_{1 \leq i \leq n} |\lambda_i| \tag{2.44}$$

is called the spectral radius of the matrix A.

Definition 2.3.2 (Spectral Norm). *Let $A = [a_{i,j}]$ be an $n \times n$ real matrix, then*

$$||A||_2 = \sup_{\mathbf{x} \neq \mathbf{0}} \frac{||A\mathbf{x}||_2}{||\mathbf{x}||_2} \tag{2.45}$$

is called the spectral norm of the matrix A

The basic properties of the spectral norm of a matrix is analogous to the Euclidean norm of a vector $\mathbf{x}$.

Definition 2.3.3 (Reducible/Irreducible Matrices). *For $n > 2$, an $n \times n$ matrix A is called reducible if there exists an $n \times n$ permutation matrix P (a square matrix which in each row and each column has some one entry unity with all others being zero) such that*

$$PAP^T = \begin{pmatrix} A_{1,1} & A_{1,2} \\ 0 & A_{2,2} \end{pmatrix},$$

where $A_{1,1}$ is an $r \times r$ submatrix and $A_{2,2}$ is an $(n-r) \times (n-r)$ submatrix, where $1 \leq r \leq n$. If no such permutation matrix exists, then A is irreducible.

Definition 2.3.4 (Diagonally Dominant Matrices). *An $n \times n$ matrix $A = [a_{i,j}]$ is diagonally dominant if*

$$|a_{i,i}| \geq \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{i,j}| \tag{2.46}$$

for all $1 \leq i \leq n$.

An $n \times n$ matrix A is **strictly diagonally dominant** if the strict inequality in (2.46) is valid for all $1 \leq i \leq n$. In the same vein, matrix A is **irreducibly diagonally dominant** if A is irreducible and diagonally dominant, with strict inequality holding in (2.46) for at least one i.

Definition 2.3.5 (Nonnegative/Positive Matrices). *Let $A = [a_{i,j}]$, we say that $A \geq 0$ (A is nonnegative) if all its entries $a_{i,j}$ are real and nonnegative. We say that $A > 0$ (A is positive) if all its entries $a_{i,j}$ are real and positive.*

Lemma 2.3.6. *If $A \geq 0$ is an irreducible $n \times n$ matrix, then*

$$(I + A)^{n-1} > 0. \tag{2.47}$$

The proof can be found in [113]

Definition 2.3.7 (Symmetric Matrices). *A real $n \times n$ matrix $A = [a_{i,j}]$ is said to be symmetric if*

$$A^T = A. \tag{2.48}$$

Definition 2.3.8 (Positive Definite/Semidefinite Matrices). *A real $n \times n$ symmetric matrix $A = [a_{i,j}]$ is said to be positive definite if*

$$\mathbf{x}^T A\mathbf{x} > 0 \tag{2.49}$$

for all nonzero $\mathbf{x} \in \mathbb{R}^d$. It is positive semidefinite if

$$\mathbf{x}^T A\mathbf{x} \geq 0 \tag{2.50}$$

for all nonzero $\mathbf{x} \in \mathbb{R}^d$. And we say it is indefinite if there are vectors $\mathbf{y}, \mathbf{z} \in \mathbb{R}^d$ such that $\mathbf{y}^T A\mathbf{y} < 0 < \mathbf{z}^T A\mathbf{z}$.

Definition 2.3.9 (M-matrices). *A real $n \times n$ matrix $A = [a_{i,j}]$ with $a_{i,j} \leq 0$ for all $i \neq j$ is an M-matrix if A is nonsingular and $A^{-1} \geq 0$.*

Definition 2.3.10 (Stieltjes Matrices). *A real $n \times n$ matrix $A = [a_{i,j}]$ with $a_{i,j} \leq 0$ for all $i \neq j$ is a Stieltjes matrix if A is symmetric and positive definite.*

Corollary 2.3.11. *If A is a Stieltjes matrix, then A is also an M-matrix. Evenmore, A is irreducible if and only if $A^{-1} > 0$.*

Corollary 2.3.12. *If a $n \times n$ matrix $A = [a_{i,j}]$ is strictly diagonally dominant or irreducibly diagonally dominant matrix with positive real diagonal entries, then A is positive definite.*

In the next section, we will outline some meshfree particle methods for the numerical solution of partial differential equations. In particular, meshfree particle methods are numerical methods that do not have a mesh connecting the grid points in the computational domain but rather based on a scattered data interpolation strategy.

2.4 Some Meshfree Particle Methods

Though, in this thesis the SPH method is the main method of solution. In this section, we will however make some brief overview of some meshfree particle methods. A quite number of them have been developed over the years but we will outline just a few.

Reproducing Kernel Particle Method

The reproducing kernel particle method (RKPM) was developed by Liu et al. [82]. The RKPM is an integral representation, this method ensures that certain degree of consistency of the finite integral approximation. The idea of the method is that a correction function is added to the kernel function. The aim of the correction is to improve particle approximation near boundaries or stated in a mathematical sense, making particle approximation linearly or C^1 consistent near boundaries. The finite integral representation of a function together with the corrected function is given by

$$\mathbf{v}^h(\mathbf{x}) = \int\limits_{\Omega(t)} \mathbf{v}(\xi) C(\mathbf{x}, \boldsymbol{\xi}) W(\mathbf{x} - \boldsymbol{\xi}, h) d\boldsymbol{\xi} \tag{2.51}$$

where $C(\mathbf{x}, \boldsymbol{\xi})$ is the correction function. An example of a correction is a linear combination given as

$$C(\mathbf{x}, \boldsymbol{\xi}) = c_1(\mathbf{x}) + c_2(\mathbf{x})(\boldsymbol{\xi} - \mathbf{x}) \tag{2.52}$$

The coefficients $c_1(\mathbf{x})$, $c_2(\mathbf{x})$ are found by enforcing the corrected kernel to reproduce the functions $c_1(\mathbf{x})$, $c_2(\mathbf{x})$ with the aid of some moment functions. After discretizing equation (2.51), we take a summation over surrounding particles:

$$\mathbf{v}^h(\mathbf{x}) = \sum_{j=1}^{N} C(\mathbf{x}, \mathbf{x}_j) W(\mathbf{x} - \mathbf{x}_j) v_j \Delta\Omega_j = \sum_{j=1}^{N} \phi_j(\mathbf{x}) v_j \tag{2.53}$$

where $\phi_j(\mathbf{x})$ represents the RKPM shape functions given by

$$\phi_j(\mathbf{x}) = C(\mathbf{x}, \mathbf{x}_j) W(\mathbf{x} - \mathbf{x}_j) \Delta\Omega_j. \tag{2.54}$$

Moving Least Squares Method

The moving least squares method (MLS) originated in data fitting and surface construction by mathematicians. It was termed the name *local regression and loss* [39]. In the paper by Lancaster and Salkauskas [70], we can find an excellent description of the MLS method. The method has been widely employed for constructing meshfree shape functions for meshless approximations. The first group to use the MLS was Nayroles et al. [94]. For mechanics problems they used MLS approximations to construct shape functions for diffuse element method (DEM). The MLS method has two major characteristics which makes it standout namely:

- The approximated function is continuous and smooth in the entire problem domain
- It can produce an approximation with the desired order of consistency

The construction procedure is as follows: Let the approximations of $v(\mathbf{x})$ at a point in the domain be denoted by $v^h(\mathbf{x})$. Then the MLS writes the field function approximation as:

$$v^h(\mathbf{x}) = \sum_{j=1}^{m} p_j(\mathbf{x}) a_j(\mathbf{x}) \equiv \mathbf{p}^T(\mathbf{x}) \mathbf{a}(\mathbf{x}), \tag{2.55}$$

where m is the number of terms in the monomial(polynomial basis), $\mathbf{a}(\mathbf{x})$ is a vector of coefficients given by the

$$\mathbf{a}^T(x) = \{a_0(x) a_1(x) \dots a_m(x)\} \tag{2.56}$$

which are functions of x, $\mathbf{p}(\mathbf{x})$ is a vector of basis functions most often of monomials of the lowest orders to ensure minimum completeness. A complete polynomial basis of order m in 1D is given by

$$\mathbf{p}^T(x) = \{p_0(x), p_1(x), \dots, p_m(x)\} = \{1, x, x^2, \cdots, x^m\} \tag{2.57}$$

in 2D

$$\mathbf{p}^T(\mathbf{x}) = \mathbf{p}^T(x, y) = \{1, x, y, xy, x^2, y^2, \dots, x^m, y^m\} \tag{2.58}$$

and in 3D space

$$\mathbf{p}^T(\mathbf{x}) = \mathbf{p}^T(x, y, z) = \{1, x, y, z, xy, yz, zx, x^2, y^2, z^2, \dots, x^m, y^m, z^m\} \tag{2.59}$$

The local approximation is defined by

$$v^h(\mathbf{x}, \mathbf{x}_j) = \sum_{j=1}^{m} p_j(\mathbf{x}_j) a_j(\mathbf{x}) = \mathbf{p}^T(\mathbf{x}_j)\mathbf{a}(\mathbf{x}). \tag{2.60}$$

The coefficients $a_j(\mathbf{x})$ are obtained by doing a weighted least square fit for the local approximation this is done by minimizing the difference between the local approximating function and the function. This gives the quadratic form:

$$\begin{aligned} J &= \sum_{j=1}^{N} W(\mathbf{x} - \mathbf{x}_j)(v^h(\mathbf{x}, \mathbf{x}_j) - v^h(\mathbf{x}_j))^2 && (2.61) \\ &= \sum_{j=1}^{N} W(\mathbf{x} - \mathbf{x}_j) \left[\mathbf{p}^T(\mathbf{x}_j)\mathbf{a}(\mathbf{x}) - v_j\right]^2 && (2.62) \end{aligned}$$

where $W(\mathbf{x} - \mathbf{x}_j)$ is a compactly supported weight function and $v_j = v(\mathbf{x}_j)$ is the *nodal parameter*. The minimization condition to find the coefficients $\mathbf{a}(\mathbf{x})$ requires that

$$\frac{\partial J}{\partial \mathbf{a}} = \mathbf{0} \tag{2.63}$$

For more details on the MLS method, we advice to check the work by Lancaster and Salkauskas [70].

H-p Clouds Method

The h-p clouds method was introduced by Duarte and Oden [41], it is an adaptive strategy by introducing p-enrichment in a meshfree discretization. The name h-p cloud came when they attached a sequence of Legendre polynomials together with a moving least square interpolant to construct a p-version meshfree interpolant. The main idea in the h-p cloud method is the construction of hierarchical basis using the partition of unity $\Phi_I^l(\mathbf{x})$. The class of functions $\mathcal{F}_N^{l,p}$ to be constructed should have the property that for an appropriate choice of vector say $\mathbf{P}$, $\mathcal{P}_p \subset span\{\mathcal{F}_N^{l,p}\}$ where $\mathcal{P}_p$ denotes the space of polynomial of degree $\leq p$. Let $\mathcal{L}_p$ denote a set of tensor-product complete polynomials $L_{ijk} \in \mathbb{R}^3$, then L_{ijk} is given as

$$L_{ijk}(\mathbf{x}) = L_i(x_1)L_j(x_2)L_k(x_3), \qquad 0 \leq i, j, k \leq p \tag{2.64}$$

where L_i is a Lengendre polynomial of degree i in $\mathbb{R}$. The partition of unity $\Phi_I^l(\mathbf{x})$ is called $\mathcal{L}_l$ reducible if it can reproduce any element $L_{ijk} \in \mathcal{L}_l$ i.e.

$$L_{ijk}(\mathbf{x}) = \sum_{I} L_{ijk}(\mathbf{x}_I)\Phi_I^l(\mathbf{x}) \tag{2.65}$$

With the h-p clouds adaptive nature, it adds hierarchically appropriate basis elements to the original partition of unity $\{\Phi_I^l(\mathbf{x})\}$ such that the resulting basis can reproduce the polynomial of degree $p > l$. The hierarchical family of functions constructed $\mathcal{F}_N^{l,p}$ has the structure expressed as

$$\mathcal{F}_N^{l,p} = \left(\{\Phi_I^l(\mathbf{x})\} \bigcup \Phi_I^l(\mathbf{x})L_{ijk}(\mathbf{x}) : 0 \leq i, j, k \leq p, i \text{ or } j \text{ or } k > l; p \geq l\right) \tag{2.66}$$

Duarte and Oden in [41] showed that $\mathcal{F}_N^{l,p}$ can reproduce $L_{ijk} \in \mathcal{L}_p$. In a 1D sense, the h-p clouds hierarchical interpolation takes the form

$$\begin{aligned} v^h(x) &= \sum_I \Phi_I^{n+1}(x) L_{IJ}(x) b_J && (2.67)\\ &= \sum_I \Phi_I^{n+1}(x) \left(v_I L_0 + \sum_{i=1}^{l} b_{iI} L_i(x) \right) && (2.68) \end{aligned}$$

where $\Phi_I^{n+1}(x)$ is the $n+1$ order moving least square interpolant. The Legendre polynomial has been used because of their better conditioning features.

Partition of Unity Finite Element Method

The partition of unity method (PUM) was developed by Babuska and Melenk [5]. This method tries to replace the status of the finite element shape function. We define a partition of unity as follows:

Definition 2.4.1 (**Partition of Unity**). *Let* $\Omega \subset \mathbb{R}^d$ *be an open and bounded domain. Let* $\Omega_1, \Omega_2, \cdots, \Omega_N$ *be a family of open sets in* $\mathbb{R}^d$*, and*

- *The family of a open set* $\{\Omega_I\}$ *generates a covering for the domain* Ω*,*

$$\Omega \in \bigcup_I \Omega_I \qquad (2.69)$$

- *There exists a family of functions,* $\Phi_I \in C_0^s(\mathbb{R}^d)$*,* $s \geq 0$*, and supp* $\{ \ \Omega_I \ \} \in \bar{\Omega}_I$
-

$$0 \leq \Phi_I(\mathbf{x}) \leq 1, \forall \mathbf{x} \in \Omega_I \qquad (2.70)$$

- *The summation*

$$\Phi_1(\mathbf{x}) + \Phi_2(\mathbf{x}) + \cdots + \Phi_N(\mathbf{x}) = 1, \forall \mathbf{x} \in \Omega \qquad (2.71)$$

This family of generating functions $\{\Phi_I\}$ *is called a partition of unity subordinate to the open cover* $\{\Omega_I\}$*.*

Equation (2.71) gives the name called *partition of unity*. Practically, the condition in Equation (2.70) may not be satisfied and it is also possible for $\Phi_I(\mathbf{x})$ to be negative in some region in the domain. The condition in (2.70) is not necessary for the PUM method but is often the case and only necessary for other properties. A very important property with the PUM is that the set of open supports can overlap, since they do not necessarily form a sub-division of Ω, so far they generate a covering for Ω. In the work by Babuska and Melenk, they are called *patches*. Griebel and Schweitzer [51] proposed a particle-partition of unity method, that is based on operator splitting, the method of characteristics, and the generalized partition of unity method, where they applied it on all possible classes of PDEs: elliptic, parabolic and hyperbolic.

Element Free Galerkin Method

The element free Galerkin (EFG) method is a meshless method developed by Belytschko et al. [14]. This method is based on the diffuse elements methods of Nayroles et al. [94]. The method possesses some major features namely:

- The construction of the shape function is done by the moving least square (MLS) approximation
- A Galerkin weak form is done to obtain the discretized system equations
- The cells of the background mesh for integration are required to carry out the integration to compute the system matrices

In EFG, MLS interpolant is used as both the trial and test functions in a Galerkin procedure as similar to the finite element methods. The difference is how to modify the Galerkin statement to accomodate MLS interpolant. One of the main difficulties in EFG is how to impose essential boundary conditions for non-interpolating MLS shape functions. Belytschko et al. [14] made use of *Lagrange multiplier* method in changing the variational statement to enforce essential boundary conditions, other method called the *penalty method* has been used by Zhu and Atluri [120] as well.

Maximum Entropy Approximants

The maximum entropy approximants started from the maximum entropy principle of Shannon and Jaynes [57, 100] from *information theory*. The idea is approximating an unknown function and chosing basis functions that are least biased, the basis functions are viewed as discrete probability distribution. Maximum entropy basis functions, denoted by $p_j(\mathbf{x}), j = 1, \ldots, N$ with $\mathbf{x} \in \mathbb{R}^d$, where d is the space dimension, are forced to be non-negative and to satisfy the zeroth and first-order consistency conditions

$$\begin{aligned} p_j(\mathbf{x}) &\geq 0, && (2.72)\\ \sum_{j=1}^{N} p_j(\mathbf{x}) &= 1, && (2.73)\\ \sum_{j=1}^{N} p_j(\mathbf{x})\mathbf{x}_j &= \mathbf{x}, && (2.74) \end{aligned}$$

where the last equation allows to identify the vectorial weights $\mathbf{x}_j$ with the positions of the nodes associated with each basis function. Arroyo and Ortiz [2] transformed the principle of Shannon and Jaynes to obtaining local maximum entropy basis functions. The approach is defined by the node set and belong to the general class of convex approximation schemes - these are schemes based on positive shape functions and interpolate affine functions exactly. Their convex approximation schemes represents a compromise in the sense of Pareto optimality - between two competing objectives:

- *Unbiased statistical inference* based on the nodal data;
- the definition of local shape functions of *least width*.

Arroyo and Ortiz wrote the following optimization problem to select the approximants using the above mentioned requirements. For fixed $\mathbf{x}$,

$$\text{minimize} \quad \sum_{j=1}^{N} \beta_j p_j |\mathbf{x} - \mathbf{x}_j|^2 + \sum_{j=1}^{N} p_j \ln p_j \tag{2.75}$$

$$\text{subject to} \quad p_j(\mathbf{x}) \geq 0, \quad j = 1, \dots, N$$

$$\sum_{j=1}^{N} p_j(\mathbf{x}) = 1, \qquad \sum_{j=1}^{N} p_j(\mathbf{x}) \mathbf{x}_j = \mathbf{x}$$

where β_j is a non-negative nodal parameters. The first term in equation (2.75) characterizes the second moment with a non-negative width and the second term gives the entropy of the associated probability distribution function. The maximum entropy approximant method possesses a weak Kronecker-Delta property at the boundary, so it allows the imposition of essential boundary conditions, i.e, every basis function associated with a node not belonging to the boundary is zero at the boundary.

3. Shallow Water Equations

In this chapter, we want to provide a background on the theory of the two-dimensional shallow water equations (SWEs) in particular the depth-averaged version of the equations and give some insights into free surface flows. We will proceed and derive the two-dimensional shallow water equations from the Navier-Stokes equations. Since the SWEs are hyperbolic in nature we will explain the hyperbolic characteristics of the equations, we give the eigenstructure of the SWEs in physical variables and we will prove some propositions needed to understand the eigenstructure. We round up the chapter by giving the Riemann invariant property of the SWEs.

3.1 Shallow Water Flows

Physical phenomena such as tides in oceans, breaking of waves on shallow beaches, tsunami waves, surges and breaking of dam are governed by the mathematical models of the *shallow water form*. Shallow water flows are also called *free surface flows* which are mostly flowing under the influence of gravity. It is called *free* in the sense that we can attribute a large difference in the densities of fluids, i.e., the density ratio of water to air is around $1000kg/m^3$. This means the inertia of the gaseous state with low density can be ignored if compared to the liquid with higher density, so the liquid can move freely with respect to the gas. However, the gas exerts its pressure on the liquid surface. So, we can say the gas-liquid surface is free, i.e., not constrained.

A very key assumption employed in the derivation of approximate shallow water equations is the pressure distribution called *hydrostatic approximation*. This assumption states that the vertical acceleration of water particles are negligible when compared to velocity of the water particles in the horizontal direction. The equations governing the flow of fluid are derived by considering a differential elemental volume of fluid and describing them mathematically by the conservation of mass and momentum. In modeling two-dimensional shallow water flows, it can be done through: the vertically averaged modeling $(2D_{xy})$ applicable in flow circulation in a class of well-mixed estuaries, lakes and coastal embayments and, the laterally averaged modeling $(2D_{xz})$ which are applicable in flows in narrow and deep estuaries.

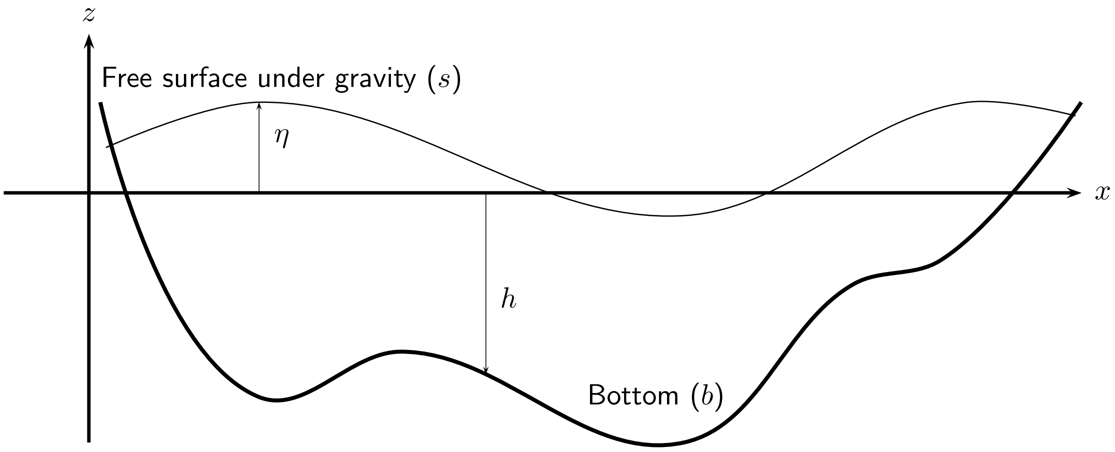


Figure 3.1: Flow Domain

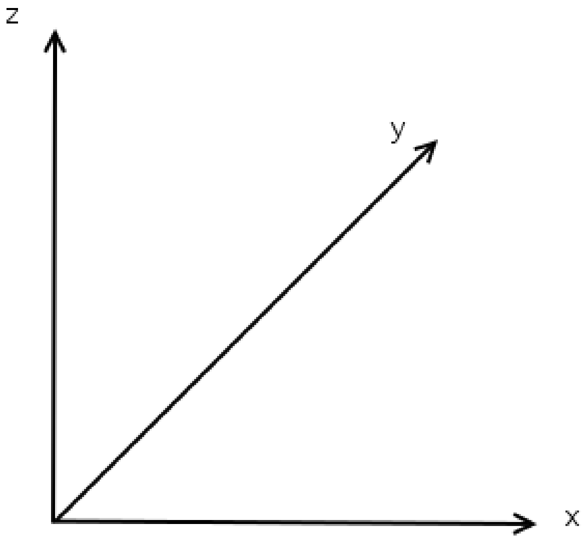


Figure 3.2: Coordinate convention for flow with free surface under gravity, $x - y$ give the horizontal plane and z give the vertical direction.

Let us consider in a three-dimensional domain the flow of water with a free surface under gravity. Figure 3.2 represents the convention for spatial coordinates; $x - y$ determines the horizontal plane and z defines the vertical direction that we have associated with the free surface elevation. In Figure 3.1 we consider the flow domain and we assume that the bottom bathymetry or profile can be expressed as single valued function

$$z = -h(x, y) \tag{3.1}$$

a similar condition at the bottom boundary is that

$$u^b h_x + v^b h_y + w^b = 0 \tag{3.2}$$

where $h(x, y)$ is the water depth measured from the undisturbed water surface, and $u^b = u(x, y, -h, t)$, $v^b = v(x, y, -h, t)$ and $w^b = w(x, y, -h, t)$ are the velocity components at the bottom. Equation (3.2) means that the velocity component that are perpendicular to the solid boundaries should vanish, so when there exists a no slip boundary condition there is no normal flow. The bottom shear stress is given as

$$\nu(u_x h_x + u_y h_y + u_z)|_{z=-h} = \gamma_b u^b \tag{3.3}$$
$$\nu(v_x h_x + v_y h_y + v_z)|_{z=-h} = \gamma_b v^b \tag{3.4}$$

where γ_b is a nonnegative bottom friction coefficient.

Also at the free surface, we assume the free surface can be expressed as a single valued function

$$z = \eta(x, y, t) \tag{3.5}$$

we call the free surface equation, also referred to as the kinematic condition of the free surface, given by

$$\eta_t + u^s \eta_x + v^s \eta_y = w^s \tag{3.6}$$

where $\eta(x, y, t)$ denotes the water surface elevation measured from the undisturbed water surface, and $u^s = u(x, y, \eta, t)$, $v^s = v(x, y, \eta, t)$ and $w^s = w(x, y, \eta, t)$ are the velocity components at the free surface. This equation means there is no relative normal flow. The surface shear stress is given as

$$\nu(-u_x\eta_x - u_y\eta_y + u_z)|_{z=\eta} = \gamma_t(u_a - u^s) \tag{3.7}$$

$$\nu(-v_x\eta_x - v_y\eta_y + v_z)|_{z=\eta} = \gamma_t(v_a - v^s) \tag{3.8}$$

where u_a, v_a are the horizontal wind velocity components and γ_t is a nonnegative wind stress coefficient.

3.1.1 Hydrostatic Pressure Distribution

In geophysical flows the vertical acceleration is often small when compared to the gravitational acceleration and to the pressure gradient in the vertical direction as in the case of our flow domain in Fig 5.1. For instance, if we consider tidal flows in the ocean the velocity in the horizontal direction is of the order of $1m/s$, while the velocity in the vertical direction is much smaller of the order of one meter per tidal cycle i.e., $10^{-5}m/s$ [27]. To this end, if the advective and viscous terms are neglected in the vertical momentum equation of the Navier-Stokes equation, we have the equation for pressure which reads

$$\frac{dp}{dz} = -\rho g. \tag{3.9}$$

Integrating equation (3.9)

$$\int_z^\eta \frac{\partial p}{\partial z} dz = -\int_z^\eta \rho g dz \tag{3.10}$$

$$p(x, y, \eta, t) - p(x, y, z, t) = -\rho g[\eta(x, y, t) - z] \tag{3.11}$$

where we have used the surface condition $p(x, y, \eta, t) = 0$, we let $p_0(x, y, \eta, t)$ denote the surface condition. The solution that satisfies (3.9) is given by the hydrostatic pressure

$$p(x, y, z, t) = p_0(x, y, \eta, t) + g[\eta(x, y, t) - z], \tag{3.12}$$

where $p_0(x, y, \eta, t)$ marks the atmospheric pressure at the free surface which without loss of generality is taken as a constant. Equation (3.12) is called the *hydrostatic pressure distribution*. Then we can have $p_x = g\eta_x$, and similarly for p_y.

Remark 3.1.1. *The pressure represents a normalized pressure, that is we mean the pressure is divided by constant density.*

3.2 Derivation of the SWE

Starting from the governing three-dimensional setting, written in primitive variables and assuming constant density, the free surface flow of an incompressible fluid are the Navier-Stokes equations (NSE) that describes the conservation of mass and momentum. The NSE written in Cartesian coordinates have the form

$$u_x + v_y + w_z = 0 \tag{3.13}$$

$$u_t + (uu)_x + (uv)_y + (uw)_z = -p_x + (\nu u_x)_x + (\nu u_y)_y + (\nu u_z)_z \tag{3.14}$$

$$v_t + (uv)_x + (vv)_y + (vw)_z = -p_y + (\nu v_x)_x + (\nu v_y)_y + (\nu v_z)_z \tag{3.15}$$

$$w_t + (uw)_x + (vw)_y + (ww)_z = -p_z + (\nu w_x)_x + (\nu w_y)_y + (\nu w_z)_z - g \tag{3.16}$$

where $u(x, y, z, t)$, $v(x, y, z, t)$ and $w(x, y, z, t)$ are the velocity components in the horizontal x, y and vertical $z-$ directions; t is the time; p is the normalized pressure, i.e., the pressure divided by constant density; g is the gravitational acceleration and ν is the viscosity coefficient.

Given the initial conditions at time $t = 0$ and the boundary conditions at the bottom bathymetry and at the free surface, the solution of the problem is given by the solution of equations (3.13 - 3.16) for the unknowns u, v, w, p. Solving equations (3.13 - 3.16) is always computationally demanding and challenging, the main difficulty in solving the full problem is associated with the free surface. This is a boundary, and boundary conditions are to be satisfied, but the position of this boundary itself is unknown and hence the domains on which the equations are to be solved are not known *a priori*. However, approximate theories that leads to simpler models should be adopted. In deriving the SWE, some assumptions could be made such as

- *Linear theory approximation*: this approximation assumes that the amplitude of the free-surface from an undisturbed position is small when compared to the characteristic length, such as wave length and thus this assumption leads to linear boundary value problems.
- *Non-linear approximation*: this approximation assumes that the water depth is small when compared to the wave length or free surface curvature and this assumption leads to non-linear initial value problems.

Even, in spite of these simplifying assumptions in deriving the non-linear shallow water model, its numerical treatment is still computationally challenging.

Now we start the derivation of the SWEs, we integrate the continuity equation i.e., $\nabla \cdot \mathbf{v} = 0$ from the bottom bathymetry to the free surface, $z = -h$ to $z = \eta$. We can apply the Leibniz integral rule since η depends on x, y, t and h depends on x, y.

$$\begin{aligned} 0 &= \int_{-h}^{\eta} \nabla \cdot \mathbf{v} dz && (3.17)\\ &= \int_{-h}^{\eta} u_x dz + \int_{-h}^{\eta} v_y dz + \int_{-h}^{\eta} w_z dz && (3.18)\\ &= \left(\int_{-h}^{\eta} u dz\right)_x - u^s \eta_x + u^b(-h)_x + \left(\int_{-h}^{\eta} v dz\right)_y - v^s \eta_y + v^b(-h)_y + w^s - w^b && (3.19) \end{aligned}$$

Using equations (3.2) and (3.6), we obtain the free surface equation

$$\eta_t + \left(\int_{-h}^{\eta} u dz\right)_x + \left(\int_{-h}^{\eta} v dz\right)_y = 0 \tag{3.20}$$

Defining the depth averaged velocities as

$$U = \frac{1}{H}\int_{-h}^{\eta} u dz, \qquad \text{and} \qquad V = \frac{1}{H}\int_{-h}^{\eta} v dz \tag{3.21}$$

and using boundary conditions to get rid of the boundary terms, we have the depth averaged continuity equation given as

$$H_t + (HU)_x + (HV)_y = 0 \tag{3.22}$$

where $H(x, y, t)$ is the total water depth written as

$$H(x, y, t) = \eta(x, y, t) + h(x, y). \tag{3.23}$$

For the momentum equation, if we integrate over the depth the left-hand side of (3.14), the $x-$momentum equation, we have

$$\begin{aligned}\int_{-h}^{\eta}(u_t+(uu)_x+(uv)_y+(uw)_z)dz &= \left(\int_{-h}^{\eta}udz\right)_t+\left(\int_{-h}^{\eta}uudz\right)_x+\left(\int_{-h}^{\eta}uvdz\right)_y\\ &\quad -u^s(\eta_t+u^s\eta_x+v^s\eta_y-w^s)+u^b(u^b(-h)_x+v^b(-h)_y-w^b)\\ &= (HU)_t+(HUU)_x+(HUV)_y\\ &\quad +\left(\int_{-h}^{\eta}(u-U)^2dz\right)_x+\left(\int_{-h}^{\eta}(u-U)(v-V)dz\right)_y \qquad (3.24)\end{aligned}$$

We should note that in equation (3.24), the product of the two integrands is not the product of the averages. We will get a similar result for the left-hand side of the $y-$momentum equation. The vertical integration of the viscous terms on the right-hand side of equation (3.14) yields

$$\begin{aligned}\int_{-h}^{\eta}((\nu u_x)_x+(\nu u_y)_y+(\nu u_z)_z)dz &= \left(\int_{-h}^{\eta}\nu u_xdz\right)_x+\left(\int_{-h}^{\eta}\nu u_ydz\right)_y\\ &\quad -\nu(u_x\eta_x+u_y\eta_y-u_z)|_{z=\eta}+\nu(u_x(-h)_x+u_y(-h)_y-u_z)|_{z=-h}\\ &= \left(\int_{-h}^{\eta}\nu U_xdz\right)_x+\left(\int_{-h}^{\eta}\nu U_ydz\right)_y\\ &\quad +\left(\int_{-h}^{\eta}\nu(u-U)_xdz\right)_x+\left(\int_{-h}^{\eta}\nu(u-U)_ydz\right)_x+\gamma_t(u_a-u^s)-\gamma_bu^b\\ &= (\tilde{\nu}HU_x)_x+(\tilde{\nu}HU_y)_y+\gamma_t(u_a-U^s)-\gamma_bU\\ &\quad +\left(\int_{-h}^{\eta}\nu(u-U)_xdz\right)_x+\left(\int_{-h}^{\eta}\nu(u-U)_ydz\right)_x\\ &\quad -\gamma_t(u_s-U)-\gamma_b(u^b-U) \qquad (3.25)\end{aligned}$$

where

$$\tilde{\nu}=\frac{1}{H}\int_{-h}^{\eta}\nu dz$$

is the depth averaged viscosity coefficient. We can obtain similar results for the $y-$ momentum equations. Then, we obtain the two-dimensional, depth averaged shallow water equations as

$$H_t+(HU)_x+(HV)_y=0 \qquad (3.26)$$

$$(HU)_t+(HUU)_x+(HUV)_y=-gH\eta_x+(\tilde{\nu}HU_x)_x+(\tilde{\nu}HU_y)_y+\gamma_tu_a-\gamma U \qquad (3.27)$$

$$(HV)_t+(HUV)_x+(HVV)_y=-gH\eta_y+(\tilde{\nu}HV_x)_x+(\tilde{\nu}HV_y)_y+\gamma_tv_a-\gamma V \qquad (3.28)$$

where the total friction coefficient is $\gamma=\gamma_b+\gamma_t$ should be nonnegative. The SWEs when written in differential conservation law form with source terms, the equations are

$$\mathbf{U}_t+\mathbf{F}(\mathbf{U})_x+\mathbf{G}(\mathbf{U})_y=\mathbf{S}(\mathbf{U}), \qquad (3.29)$$

where $\mathbf{U}$, $\mathbf{F}(\mathbf{U})$, $\mathbf{G}(\mathbf{U})$ and $\mathbf{S}(\mathbf{U})$ are the vectors of conserved variables, fluxes in the x and y directions and sources, they are given by

$$\mathbf{U} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \qquad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix},$$

$$\mathbf{G}(\mathbf{U}) = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}, \qquad \mathbf{S}(\mathbf{U}) = \begin{pmatrix} s_1 \\ s_2 \\ s_3 \end{pmatrix}.$$

$\mathbf{S}(\mathbf{U})$ is the source term vector which accounts for various physical and geometric effects. Mostly, when the bottom bed bathymetry is varying, the source term vector becomes

$$\mathbf{S} = \begin{pmatrix} 0 \\ -ghb_x \\ -ghb_y \end{pmatrix}.$$

Forces such as bottom friction, wind stresses and Coriolis forces enter into the vector $\mathbf{S}(\mathbf{U})$.

3.3 Characteristics of the SWE

In this section, we will give the characteristic properties of the shallow water equations.

3.3.1 Hyperbolicity

Hyperbolicity remains the most important property of the shallow water equations. When we consider the systems of partial differential equations, this property defines the eigenvalues of the Jacobian matrices of flux functions. For hyperbolic systems, considering special initial conditions we arrive at the so-called *Riemann problems* which divides the solution into separate waves which can be determined either analytically or approximately.

Definition 3.3.1 (**Hyperbolic System**). *A system is said to be hyperbolic if the Jacobian matrix of flux functions has m real eigenvalues and a corresponding complete set of m linearly independent eigenvectors.*

Remark 3.3.2. *For hyperbolicity, the eigenvalues are not required to be all distinct. The crucial point is that there is a complete set of linearly independent eigenvectors corresponding to the real eigenvalues.*

Definition 3.3.3 (**Strictly Hyperbolic System**). *A hyperbolic system is said to be strictly hyperbolic if all the eigenvalues of the system are distinct.*

Definition 3.3.4 (**Weakly Hyperbolic System**). *A system may have real but not distinct eigenvalues but still be hyperbolic if a complete set of linearly independent eigenvector exists. However, if all eigenvalues are real but no complete set of linearly independent eigenvectors exists then the system is called weakly hyperbolic.*

3.3.2 Eigenstructure in Terms of Physical Variables

The aim of this section is to present some theoretical results and properties of the SWEs in terms of primitive variables. This is because our proposed semi-implicit numerical method SISPH which is designed and developed in chapter 5 in this thesis is formulated in primitive variables. We can formulate the shallow water equations in terms of *physical variables* or *primitive variables*. Equation (3.29) with variable bottom bathymetry can be written as

$$h_t + uh_x + hu_x + vh_y + hv_y = 0, \tag{3.30}$$

$$u_t + uu_x + gh_x + vu_y = -gb_x, \tag{3.31}$$

$$v_t + uv_x + vv_y + gh_y = -gb_y. \tag{3.32}$$

Writing the conservative equations in non-conservative(quasi-linear) form in compact notation becomes

$$\mathbf{U}_t + \mathbf{A}(\mathbf{U})\mathbf{U}_x + \mathbf{B}(\mathbf{U})\mathbf{U}_y = \mathbf{S}, \tag{3.33}$$

where the coefficient matrices $\mathbf{A}(\mathbf{U})$ and $\mathbf{B}(\mathbf{U})$ and the vectors $\mathbf{U}$ and $\mathbf{S}$ are given as

$$\mathbf{A}(\mathbf{U}) = \begin{pmatrix} u & h & 0 \\ g & u & 0 \\ 0 & 0 & u \end{pmatrix}, \qquad \mathbf{B}(\mathbf{U}) = \begin{pmatrix} v & 0 & h \\ 0 & v & 0 \\ g & 0 & v \end{pmatrix},$$

$$\mathbf{U} = \begin{pmatrix} h \\ u \\ v \end{pmatrix}, \qquad \mathbf{S} = \begin{pmatrix} 0 \\ -gb_x \\ -gb_y \end{pmatrix}.$$

Definition 3.3.5 (Eigenvalues). *The eigenvalues of system (3.33) are the roots of the characteristic polynomial*

$$P(\lambda) \equiv |\mathbf{A} - \lambda \mathbf{I}| = 0. \tag{3.34}$$

$\mathbf{I}$ *is a* $m \times m$ *unit matrix,* λ *is a parameter,* λ_i *are eigenvalues in increasing order*

$$\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_i \leq \ldots \leq \lambda_{m-1} \leq \lambda_m. \tag{3.35}$$

Definition 3.3.6 (Right Eigenvector). *A right eigenvector* $\mathbf{R}_i$ *of* $\mathbf{A}$ *corresponding to* λ_i *is*

$$\mathbf{R}^{(i)} = \left(r_{1i},\ r_{2i}, \ldots\ r_{ii},\ \ldots,\ r_{mi}\right)^T, \tag{3.36}$$

such that

$$\mathbf{A}\mathbf{R}^{(i)} = \lambda_i \mathbf{R}^{(i)}. \tag{3.37}$$

The m *right eigenvectors corresponding to the eigenvalues (3.35) are*

$$\mathbf{R}^{(1)}, \mathbf{R}^{(2)}, \ldots, \mathbf{R}^{(i)}, \ldots, \mathbf{R}^{(m-1)}, \mathbf{R}^{(m)}. \tag{3.38}$$

Definition 3.3.7 (Left Eigenvector). *A left eigenvector* $\mathbf{L}_i$ *of* $\mathbf{A}$ *corresponding to* λ_i *is the row vector*

$$\mathbf{L}^{(i)} = \left(r_{i1},\ r_{i2}, \ldots\ r_{ii},\ \ldots,\ r_{im}\right), \tag{3.39}$$

such that

$$\mathbf{L}^{(i)}\mathbf{A} = \lambda_i \mathbf{L}^{(i)}. \tag{3.40}$$

The m *left eigenvectors corresponding to the eigenvalues (3.35) are*

$$\mathbf{L}^{(1)}, \mathbf{L}^{(2)}, \ldots, \mathbf{L}^{(i)}, \ldots, \mathbf{L}^{(m-1)}, \mathbf{L}^{(m)}. \tag{3.41}$$

Proposition 3.3.8. *The eigenvalues of* $\mathbf{A}$ *and* $\mathbf{B}$ *in equation (3.33) are respectively given by*

$$\lambda_1 = u - c, \qquad \lambda_2 = u, \qquad \lambda_3 = u + c \tag{3.42}$$

and

$$\tilde{\lambda}_1 = v - c, \qquad \tilde{\lambda}_2 = v, \qquad \tilde{\lambda}_3 = v + c. \tag{3.43}$$

Proof. The definition of the eigenvalues of the matrix $\mathbf{A}$ are given by the roots of the *characteristic polynomial*

$$|\mathbf{A} - \lambda \mathbf{I}| = 0, \tag{3.44}$$

where $|\mathbf{J}| = |\mathbf{A} - \lambda \mathbf{I}|$ denotes the determinant of matrix $\mathbf{J}$, λ is a scalar and $\mathbf{I}$ denotes the identity matrix. Evaluating the determinant in (3.44) we obtain

$$\begin{aligned} u - \lambda(u-\lambda)^2 - gh(u-\lambda) &= 0, \\ (u-\lambda)\left[(u-\lambda)^2 - gh\right] &= 0, \end{aligned}$$

where we obtain the roots to be

$$\lambda_1 = u - c, \qquad \lambda_2 = u, \qquad \lambda_3 = u + c,$$

where $c = \sqrt{gh}$. We note that the eigenvalues are real and distinct under all possible circumstances, except in the case of dry bed $h = 0$, which implies $c = 0$ and $\lambda_1 = \lambda_2 = \lambda_3 = u$. Similarly, we get the eigenvalues of $\mathbf{B}$ by solving the characteristic polynomial $|\mathbf{B} - \tilde{\lambda}\mathbf{I}| = 0$. ☐

Proposition 3.3.9. *The right eigenvectors of* $\mathbf{A}$ *in equation (3.33) are*

$$\mathbf{R}^{(1)} = \alpha_1 \begin{pmatrix} h \\ -c \\ 0 \end{pmatrix}, \qquad \mathbf{R}^{(2)} = \alpha_2 \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \qquad \mathbf{R}^{(3)} = \alpha_3 \begin{pmatrix} h \\ c \\ 0 \end{pmatrix},$$

where α_1, α_2 *and* α_3 *are scaling factors. The left eigenvectors of* $\mathbf{A}$ *in equation (3.33) are*

$$\left.\begin{aligned} \mathbf{L}^{(1)} &= \hat{\alpha_1}\,(c, \;\; -h, \;\; 0)\,, \\ \mathbf{L}^{(2)} &= \hat{\alpha_2}\,(0, \;\; 0, \;\; 1)\,, \\ \mathbf{L}^{(3)} &= \hat{\alpha_3}\,(c, \;\; h, \;\; 0)\,, \end{aligned}\right\} \tag{3.45}$$

where $\hat{\alpha_1}$, $\hat{\alpha_2}$ *and* $\hat{\alpha_3}$ *are scaling factors.*

Proof. The *right eigenvector* of the matrix $\mathbf{A}$ corresponding to an eigenvalue λ is a column vector $\mathbf{R} = (r_1, \;\; r_2, \;\; r_3)^T$ such that

$$\mathbf{A}\mathbf{R} = \lambda \mathbf{R}, \tag{3.46}$$

where we write in full notation as

$$\left.\begin{aligned} ur_1 + hr_2 &= \lambda r_1, \\ gr_1 + ur_2 &= \lambda r_2, \\ ur_3 &= \lambda r_3. \end{aligned}\right\} \tag{3.47}$$

For $\lambda = \lambda_1 = u - c$, we wish to find the right eigenvector $\mathbf{R}^{(1)}$ corresponding to it. We obtain

$$\begin{aligned} ur_1 + hr_2 &= (u-c)r_1, && (3.48)\\ gr_1 + ur_2 &= (u-c)r_2, && (3.49)\\ ur_3 &= (u-c)r_3. && (3.50) \end{aligned}$$

We should note that equations (3.48) and (3.49) are equivalent since $c = \sqrt{gh}$. This means, we have only two independent equations for the three unknowns r_1, r_2, r_3. So, we use equation (3.48) that gives a one-parameter family of solutions. Prescribing the parameter as α_1, we obtain the right eigenvector

$$\mathbf{R}^{(1)} = \alpha_1 \begin{pmatrix} h \\ -c \\ 0 \end{pmatrix}.$$

We do similar calculations for $\lambda = \lambda_2 = u$ by substituting into (3.47) to obtain $\mathbf{R}^{(2)}$ and $\lambda = \lambda_3 = u+c$ to obtain $\mathbf{R}^{(3)}$ respectively as claimed.

For the *left eigenvector* of the matrix $\mathbf{A}$ corresponding to an eigenvalue λ, the row vector $\mathbf{L} = (l_1,\ l_2,\ l_3)$ is found by solving the equation

$$\mathbf{LA} = \lambda \mathbf{L}, \tag{3.51}$$

Written in full notation as

$$\left.\begin{aligned} ul_1 + gl_2 &= \lambda l_1,\\ hl_1 + ul_2 &= \lambda l_2,\\ ul_3 &= \lambda l_3. \end{aligned}\right\} \tag{3.52}$$

For $\lambda = \lambda_1 = u - c$, We wish to find the left eigenvector $\mathbf{L}^{(1)}$ corresponding to it. We obtain

$$\begin{aligned} ul_1 + gl_2 &= (u-c)l_1, && (3.53)\\ hl_1 + ul_2 &= (u-c)l_2, && (3.54)\\ ul_3 &= (u-c)l_3. && (3.55) \end{aligned}$$

Also, we should note that (3.53) and (3.54) are equivalent since $c = \sqrt{gh}$. We have only two independent equations for the three unknowns l_1, l_2, l_3. After solving we obtain, a one-parameter family of solutions. Prescribing the parameter as $\hat{\alpha_1}$, we obtain the left eigenvector

$$\mathbf{L}^{(1)} = \hat{\alpha_1}\,(c,\ \ -h,\ \ 0)$$

We do similar calculations for $\lambda = \lambda_2 = u$ by substituting into (3.52) to obtain $\mathbf{L}^{(2)}$ and $\lambda = \lambda_3 = u+c$ to obtain $\mathbf{L}^{(3)}$ respectively as claimed. ☐

Proposition 3.3.10. *The right eigenvectors of* $\mathbf{B}$ *in equation (3.33) are*

$$\mathbf{R}^{(1)} = \beta_1 \begin{pmatrix} h \\ 0 \\ -c \end{pmatrix}, \qquad \mathbf{R}^{(2)} = \beta_2 \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \qquad \mathbf{R}^{(3)} = \beta_3 \begin{pmatrix} h \\ 0 \\ c \end{pmatrix},$$

where β_1, β_2 *and* β_3 *are scaling factors. The left eigenvectors of* $\mathbf{B}$ *in equation (3.33) are*

$$\left.\begin{aligned} \mathbf{L}^{(1)} &= \hat{\beta}_1\,(c,\ \ 0,\ \ -h)\,,\\ \mathbf{L}^{(2)} &= \hat{\beta}_2\,(0,\ \ 1,\ \ 0)\,,\\ \mathbf{L}^{(3)} &= \hat{\beta}_3\,(c,\ \ 0,\ \ h)\,, \end{aligned}\right\} \tag{3.56}$$

where $\hat{\beta}_1$, $\hat{\beta}_2$ *and* $\hat{\beta}_3$ *are scaling factors.*

Proof. The calculations for for the right eigenvectors and left eigenvectors of matrix $\mathbf{B}$ follows similar calculations as we have shown for $\mathbf{A}$ above. ☐

Remark 3.3.11 (Bi-orthonormality). *The left and right eigenvectors of the Jacobian matrix* $\mathbf{A}$ *are* bi-orthonormal, *that is they satisfy the relations*

$$\mathbf{L}^{(i)} \cdot \mathbf{R}^{(j)} = \begin{cases} 1 & \textit{if } i = j, \\ 0 & \textit{if } i \neq j. \end{cases}$$

Theorem 3.3.12. *The two-dimensional shallow water equations are strictly hyperbolic for* $c > 0$.

Proof. From the coefficient matrices $\mathbf{A}(\mathbf{U})$ and $\mathbf{B}(\mathbf{U})$, we solve the characteristic polynomial (3.44) and obtain the eigenvalues. The eigenvalues are distinct for $c = \sqrt{gh} \neq 0$, therefore the two-dimensional shallow water equations are strictly hyperbolic for $gh > 0$. ☐

Let us consider a matrix $\mathbf{C}(\mathbf{U})$ that is a *linear combination* of the two Jacobian coefficient matrices $\mathbf{A}(\mathbf{U})$ and $\mathbf{B}(\mathbf{U})$ as

$$\mathbf{C}(\mathbf{U}) = \omega_1 \mathbf{A}(\mathbf{U}) + \omega_2 \mathbf{B}(\mathbf{U}), \tag{3.57}$$

where ω_1, $\omega_2 \in \mathbb{R}$ define a non-zero vector $\boldsymbol{\omega} = [\omega_1, \omega_2]$ such that

$$|\boldsymbol{\omega}| = \sqrt{\omega_1^2 + \omega_2^2} > 0. \tag{3.58}$$

Therefore, the matrix $\mathbf{C}(\mathbf{U})$ is given by

$$\mathbf{C}(\mathbf{U}) = \omega_1 \begin{pmatrix} u & h & 0 \\ g & u & 0 \\ 0 & 0 & u \end{pmatrix} + \omega_2 \begin{pmatrix} v & 0 & h \\ 0 & v & 0 \\ g & 0 & v \end{pmatrix}$$

Then we have

$$\mathbf{C}(\mathbf{U}) = \begin{pmatrix} u\omega_1 + v\omega_2 & h\omega_1 & h\omega_2 \\ g\omega_1 & u\omega_1 + v\omega_2 & 0 \\ g\omega_2 & 0 & u\omega_1 + v\omega_2 \end{pmatrix} \tag{3.59}$$

This leads us to the following propositions

Proposition 3.3.13. *The eigenvalues of* $\mathbf{C}$ *in equation (3.59) are given by*

$$\lambda_1 = u\omega_1 + v\omega_2 - c|\boldsymbol{\omega}|, \qquad \lambda_2 = u\omega_1 + v\omega_2, \qquad \lambda_3 = u\omega_1 + v\omega_2 + c|\boldsymbol{\omega}| \tag{3.60}$$

Proof. The definition of the eigenvalues of the matrix $\mathbf{C}$ are given by the roots of the *characteristic polynomial*

$$|\mathbf{C} - \lambda \mathbf{I}| = 0, \tag{3.61}$$

Evaluating the determinant in (3.61) we obtain

$$\begin{aligned} u\omega_1 + v\omega_2 - \lambda \left[(u\omega_1 + v\omega_2 - \lambda)^2\right] - h\omega_1 \left[g\omega_1(u\omega_1 + v\omega_2 - \lambda)\right] + h\omega_2 \left[-g\omega_1(u\omega_1 + v\omega_2 - \lambda)\right] &= 0, \\ u\omega_1 + v\omega_2 - \lambda \left[(u\omega_1 + v\omega_2 - \lambda)^2 - gh\omega_1^2 - gh\omega_2^2\right] &= 0, \\ u\omega_1 + v\omega_2 - \lambda \left[(u\omega_1 + v\omega_2 - \lambda)^2 - gh(\omega_1^2 + \omega_2^2)\right] &= 0 \end{aligned}$$

where we obtain the roots to be

$$\lambda_1 = u\omega_1 + v\omega_2 - c|\boldsymbol{\omega}|, \qquad \lambda_2 = u\omega_1 + v\omega_2, \qquad \lambda_3 = u\omega_1 + v\omega_2 + c|\boldsymbol{\omega}| \tag{3.62}$$

as claimed. ☐

Proposition 3.3.14. *The right eigenvectors of* $\mathbf{C}$ *are*

$$\mathbf{R}^{(1)} = \alpha_1 \begin{pmatrix} -c|\boldsymbol{\omega}| \\ \frac{c^2}{h}\omega_1 \\ \frac{c^2}{h}\omega_2 \end{pmatrix}, \qquad \mathbf{R}^{(2)} = \alpha_2 \begin{pmatrix} 0 \\ -\omega_2 \\ \omega_1 \end{pmatrix}, \qquad \mathbf{R}^{(3)} = \alpha_3 \begin{pmatrix} \frac{c^2}{h}\omega_1 \\ c|\boldsymbol{\omega}| \\ \frac{c^2}{h}\omega_2 \end{pmatrix},$$

where α_1, α_2 *and* α_3 *are scaling factors. The left eigenvectors of* $\mathbf{C}$ *are*

$$\left.\begin{aligned} \mathbf{L}^{(1)} &= \hat{\alpha_1}\left(-c|\boldsymbol{\omega}|,\ \ h\omega_1,\ \ h\omega_2\right), \\ \mathbf{L}^{(2)} &= \hat{\alpha_2}\left(0,\ \ -\omega_2,\ \ \omega_1\right), \\ \mathbf{L}^{(3)} &= \hat{\alpha_1}\left(c|\boldsymbol{\omega}|,\ \ h\omega_1,\ \ h\omega_2\right), \end{aligned}\right\} \tag{3.63}$$

where $\hat{\alpha_1}$, $\hat{\alpha_2}$ *and* $\hat{\alpha_3}$ *are scaling factors.*

Proof. The *right eigenvector* of the matrix $\mathbf{C}$ corresponding to an an eigenvalue λ is a column vector $\mathbf{R} = (r_1,\ \ r_2,\ \ r_3)^T$ such that

$$\mathbf{C}\mathbf{R} = \lambda\mathbf{R}, \tag{3.64}$$

where we write in full notation as

$$\left.\begin{aligned} (u\omega_1 + v\omega_2)r_1 + h\omega_1 r_2 + h\omega_2 r_3 &= \lambda r_1, \\ g\omega_1 r_1 + (u\omega_1 + v\omega_2)r_2 &= \lambda r_2, \\ g\omega_2 r_1 + (u\omega_1 + v\omega_2)r_3 &= \lambda r_3. \end{aligned}\right\} \tag{3.65}$$

For $\lambda = \lambda_1 = u\omega_1 + v\omega_2 - c|\boldsymbol{\omega}|$, we find the right eigenvector $\mathbf{R}^{(1)}$ corresponding to it. We obtain

$$(u\omega_1 + v\omega_2)r_1 + h\omega_1 r_2 + h\omega_2 r_3 = (u\omega_1 + v\omega_2 - c|\boldsymbol{\omega}|)r_1, \tag{3.66}$$

$$g\omega_1 r_1 + (u\omega_1 + v\omega_2)r_2 = (u\omega_1 + v\omega_2 - c|\boldsymbol{\omega}|)r_2, \tag{3.67}$$

$$g\omega_2 r_1 + (u\omega_1 + v\omega_2)r_3 = (u\omega_1 + v\omega_2 - c|\boldsymbol{\omega}|)r_3. \tag{3.68}$$

Solving (3.67) and (3.68) we a get a one parameter family of solutions. Prescribing the parameter as α_1, we obtain the right eigenvector

$$\mathbf{R}^{(1)} = \alpha_1 \begin{pmatrix} -c|\boldsymbol{\omega}| \\ \frac{c^2}{h}\omega_1 \\ \frac{c^2}{h}\omega_2 \end{pmatrix}.$$

We do similar calculations for $\lambda = \lambda_2 = u\omega_1 + v\omega_2$ by substituting into (3.65) to obtain $\mathbf{R}^{(2)}$ and $\lambda = \lambda_3 = u\omega_1 + v\omega_2 + c|\boldsymbol{\omega}|$ to obtain $\mathbf{R}^{(3)}$ respectively as claimed.

For the *left eigenvector* of the matrix $\mathbf{C}$ corresponding to an eigenvalue λ, the row vector $\mathbf{L} = (l_1,\ \ l_2,\ \ l_3)$ by solving the equation

$$\mathbf{L}\mathbf{C} = \lambda\mathbf{L}, \tag{3.69}$$

Written in full notation as

$$\left.\begin{aligned} (u\omega_1 + v\omega_2)l_1 + g\omega_1 l_2 + g\omega_2 l_3 &= \lambda l_1, \\ h\omega_1 l_1 + (u\omega_1 + v\omega_2)l_2 &= \lambda l_2, \\ h\omega_2 l_1 + (u\omega_1 + v\omega_2)l_3 &= \lambda l_3. \end{aligned}\right\} \tag{3.70}$$

For $\lambda = \lambda_1 = u\omega_1 + v\omega_2 - c|\boldsymbol{\omega}|$, we wish to find the left eigenvector $\mathbf{L}^{(1)}$ corresponding to it. We obtain

$$\begin{aligned} (u\omega_1 + v\omega_2)l_1 + g\omega_1 l_2 + g\omega_2 l_3 &= (u\omega_1 + v\omega_2 - c|\boldsymbol{\omega}|)l_1, & (3.71)\\ h\omega_1 l_1 + (u\omega_1 + v\omega_2)l_2 &= (u\omega_1 + v\omega_2 - c|\boldsymbol{\omega}|)l_2, & (3.72)\\ h\omega_2 l_1 + (u\omega_1 + v\omega_2)l_3 &= (u\omega_1 + v\omega_2 - c|\boldsymbol{\omega}|)l_3. & (3.73) \end{aligned}$$

Using (3.72) and (3.73), we have only two independent equations for the three unknowns l_1, l_2, l_3. After solving we obtain, a one-parameter family of solutions. Prescribing the parameter as $\hat{\alpha_1}$, we obtain the left eigenvector

$$\mathbf{L}^{(1)} = \hat{\alpha_1}\left(-c|\boldsymbol{\omega}|,\;\; h\omega_1,\;\; h\omega_2\right).$$

We do similar calculations for $\lambda = \lambda_2 = u\omega_1 + v\omega_2$ by substituting into (3.70) to obtain $\mathbf{L}^{(2)}$ and $\lambda = \lambda_3 = u\omega_1 + v\omega_2 + c|\boldsymbol{\omega}|$ to obtain $\mathbf{L}^{(3)}$ respectively as claimed. ☐

Definition 3.3.15 (Hyperbolic System). *A system of m conservation laws with Jacobian coefficient matrices $\mathbf{A}(\mathbf{U})$ and $\mathbf{B}(\mathbf{U})$ is said to be hyperbolic if the matrix $\mathbf{C}(\mathbf{U})$ formed by the linear combination of the Jacobian coefficient matrices $\mathbf{A}(\mathbf{U})$ and $\mathbf{B}(\mathbf{U})$,*

$$\mathbf{C}(\mathbf{U}) = \omega_1\mathbf{A}(\mathbf{U}) + \omega_2\mathbf{B}(\mathbf{U}), \tag{3.74}$$

has m real eigenvalues for any vector $\mathbf{U}$ of conserved variables and any vector $\boldsymbol{\omega} = [\omega_1, \omega_2]$ such that $\boldsymbol{\omega} \neq \mathbf{0}$. The system is called strictly hyperbolic if in addition the eigenvalues are all distinct.

Nature of Characteristic Fields

If we consider the hyperbolic system of m conservation laws of the form

$$\mathbf{U}_t + \mathbf{A}(\mathbf{U})\mathbf{U}_x = \mathbf{0}, \tag{3.75}$$

with real eigenvalues $\lambda_i(\mathbf{U})$ and corresponding right eigenvectors $\mathbf{R}^{(i)}(\mathbf{U})$. The *characteristic speed* $\lambda_i(\mathbf{U})$ defines a *characteristic field*, the λ_i field. The *gradient* of an eigenvalue $\lambda_i(\mathbf{U})$ is given by

$$\nabla\lambda_i(\mathbf{U}) = \left(\frac{\partial}{\partial u_1}\lambda_i, \frac{\partial}{\partial u_2}\lambda_i, \cdots, \frac{\partial}{\partial u_m}\lambda_i\right)^T. \tag{3.76}$$

Proposition 3.3.16 (Nature of the characteristic fields). *For the x- split shallow water equations*

$$\mathbf{U}_t + \mathbf{A}(\mathbf{U})\mathbf{U}_x = \mathbf{0}, \tag{3.77}$$

the $\lambda_2(\mathbf{U})$ characteristic field is linearly degenerate and the $\lambda_1(\mathbf{U})$ and $\lambda_3(\mathbf{U})$ characteristic fields are genuinely nonlinear.

Proof. Firstly, we show that the $\lambda_2(\mathbf{U})$ characteristic field is linearly degenerate, we mean i.e.,

$$\nabla\lambda_2(\mathbf{U})\cdot\mathbf{R}^{(2)}(\mathbf{U}) = 0$$

for all vectors $\mathbf{U}$ written as

$$\mathbf{U} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} = \begin{pmatrix} h \\ u \\ v \end{pmatrix}$$

and therefore

$$\nabla\lambda_2 = \left(\frac{\partial}{\partial u_1}\lambda_2, \frac{\partial}{\partial u_2}\lambda_2, \frac{\partial}{\partial u_3}\lambda_2\right)^T = (0,1,0)^T. \tag{3.78}$$

Since we have established the right eigenvector to be $\mathbf{R}^{(2)}(\mathbf{U}) = \alpha_2\,(0,0,1)^T$. Clearly, the dot product $\nabla\lambda_2(\mathbf{U}) \cdot \mathbf{R}^{(2)}(\mathbf{U})$ vanishes for all vectors $\mathbf{U}$ and therefore the $\lambda_2(\mathbf{U})$ characteristic field is linearly degenerate as claimed and the first part of the proposition is proved. In the same vein for the $\lambda_1(\mathbf{U})$ characteristic field

$$\nabla\lambda_1 = \left(\frac{\partial}{\partial u_1}\lambda_1, \frac{\partial}{\partial u_2}\lambda_1, \frac{\partial}{\partial u_3}\lambda_1\right)^T = \left(-\frac{1}{2}\sqrt{\frac{g}{h}}, 1, 0\right)^T. \tag{3.79}$$

As $\mathbf{R}^{(1)}(\mathbf{U}) = (h, -c, 0)^T$, the dot product

$$\begin{aligned}\nabla\lambda_1(\mathbf{U}) \cdot \mathbf{R}^{(1)}(\mathbf{U}) &= \left(-\frac{1}{2}\sqrt{\frac{g}{h}}, 1, 0\right) \cdot (h, -c, 0)\\ &= \left(-\frac{1}{2}\sqrt{\frac{g}{h}}h - c + 0\right)\\ &= -\frac{1}{2}c - c\\ &= -\frac{3}{2}c \neq 0 \quad \forall \quad \mathbf{U} \in \mathbb{R}^m\end{aligned}$$

And lastly similarly, for the $\lambda_3(\mathbf{U})$ characteristic field

$$\nabla\lambda_3(\mathbf{U}) \cdot \mathbf{R}^{(3)}(\mathbf{U}) = \frac{3}{2}c \neq 0 \quad \forall \quad \mathbf{U} \in \mathbb{R}^m$$

Therefore the $\lambda_1(\mathbf{U})$ and $\lambda_3(\mathbf{U})$ characteristic fields are genuinely nonlinear and we complete the proof of the proposition. ☐

3.3.3 Riemann Invariants

Riemann invariants are relations that holds true, i.e., remain constant across the wave structure for rarefaction waves and contact discontinuities which leads to $m-1$ ordinary differential equations.

$$\frac{dw_1}{r_1^{(i)}} = \frac{dw_2}{r_2^{(i)}} = \frac{dw_3}{r_3^{(i)}} = \cdots = \frac{dw_m}{r_m^{(i)}}. \tag{3.80}$$

They relate ratios of changes dw_s of a quantity w_s to the respective components $r_s^{(i)}$ of the right eigenvector $\mathbf{R}^{(i)}$ corresponding to a λ_i wave family. The family of ordinary differential equations are obtained from the following proposition.

Proposition 3.3.17. *The Riemann invariants for the shallow water equations are given as* $\mathbf{u} \pm 2c$.

Proof. We consider a no wind stress, no Coriolis force, no bed variation bathymetry for the two-dimensional shallow water equations

$$\frac{\partial\eta}{\partial t} + \frac{\partial}{\partial x}((\eta+h)u) + \frac{\partial}{\partial y}((\eta+h)v) = 0, \tag{3.81}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + g\frac{\partial\eta}{\partial x} = 0, \tag{3.82}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + g\frac{\partial\eta}{\partial y} = 0. \tag{3.83}$$

We consider a 2d, i.e., in the xz plane, assume $v \equiv 0$ and $\frac{\partial}{\partial y} \equiv 0$, the shallow water equations become

$$\begin{aligned} \frac{\partial}{\partial t}(\eta + h) + u\frac{\partial}{\partial x}(\eta + h) + (\eta + h)\frac{\partial u}{\partial x} &= 0, & (3.84) \\ \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + g\frac{\partial}{\partial x}(\eta + h) &= g\frac{\partial h}{\partial x}. & (3.85) \end{aligned}$$

We define the wave celerity $c^2(x, y, t) = g(\eta + h)$. Multiply equation (3.84) by gravity g we have

$$\frac{\partial c^2}{\partial t} + u\frac{\partial c^2}{\partial x} + c^2\frac{\partial u}{\partial x} = 0 \tag{3.86}$$

which is written as

$$c\left[\frac{\partial(2c)}{\partial t} + u\frac{\partial(2c)}{\partial x} + c\frac{\partial u}{\partial x}\right] = 0. \tag{3.87}$$

Because $c^2(x, y, t) \neq 0$, from equation (3.87) we have

$$\frac{\partial(2c)}{\partial t} + u\frac{\partial(2c)}{\partial x} + c\frac{\partial u}{\partial x} = 0. \tag{3.88}$$

Similarly, in equation (3.85) we have

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + c\frac{\partial(2c)}{\partial x} = g\frac{\partial h}{\partial x}. \tag{3.89}$$

Adding and substracting equations (3.89) and (3.88) we have

$$\begin{aligned} \frac{\partial}{\partial t}(u + 2c) + u\frac{\partial}{\partial x}(u + 2c) + c\frac{\partial}{\partial x}(u + 2c) &= g\frac{\partial h}{\partial x}, & (3.90) \\ \frac{\partial}{\partial t}(u - 2c) + u\frac{\partial}{\partial x}(u - 2c) - c\frac{\partial}{\partial x}(u - 2c) &= g\frac{\partial h}{\partial x}. & (3.91) \end{aligned}$$

Equation (3.90) says that along the curves in the (x, t) plane we have

$$\frac{dx}{dt} = u + c, \tag{3.92}$$

where $u + 2c$ evolves according to

$$\frac{\partial}{\partial t}(u + 2c) + \frac{dx}{dt}\frac{\partial}{\partial x}(u + 2c) = \frac{d}{dt}(u + 2c) = g\frac{\partial h}{\partial x} \tag{3.93}$$

Also along the curves defined by

$$\frac{dx}{dt} = u - c, \tag{3.94}$$

Equation (3.91) can be written as

$$\frac{\partial}{\partial t}(u - 2c) + \frac{dx}{dt}\frac{\partial}{\partial x}(u - 2c) = \frac{d}{dt}(u - 2c) = g\frac{\partial h}{\partial x} \tag{3.95}$$

Hence, if h is constant, $u + 2c$ and $u - 2c$ are Riemann invariant, that is the functions remain constant along the curves. □

Definition 3.3.18 (Riemann problem). *Let* $\mathbf{u}_L$ *and* $\mathbf{u}_R$ *be left states and right states of* $\Omega \subset \mathbb{R}^d$*. Then, the initial value problem*

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial}{\partial x}\mathbf{f}(\mathbf{u}) = \mathbf{0}, \tag{3.96}$$

with the initial condition

$$\mathbf{u}(x,0) = \begin{cases} \mathbf{u}_L & \text{if } x \leq 0, \\ \mathbf{u}_R & \text{if } x > 0. \end{cases}$$

is called a Riemann problem.

Looking for piecewise smooth continuous functions $\mathbf{u} : (x,t) \to \mathbf{u}(x,t)$ solutions of (3.96). Godlewski and Raviart [49] shows that we restrict ourselves to *self-similar solutions* of the Riemann problem. Self-similar solutions are solutions of the form

$$\mathbf{u}(x,t) = \mathbf{s}\left(\frac{x}{t}\right). \tag{3.97}$$

If we consider *classical* self-similar solutions of (3.96), these solutions satisfy the equation

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{A}(\mathbf{u})\frac{\partial \mathbf{u}}{\partial x} = \mathbf{0}$$

in the classical sense. We should have

$$-\left(\frac{x}{t^2}\right)\mathbf{s}'\left(\frac{x}{t}\right) + \left(\frac{1}{t}\right)\mathbf{A}\left(\mathbf{s}\left(\frac{x}{t}\right)\right)\mathbf{s}'\left(\frac{x}{t}\right) = \mathbf{0},$$

if we set $\xi = \frac{x}{t}$, we have

$$\left(\mathbf{A}(\mathbf{s}(\xi)) - \xi\mathbf{I}\right)\mathbf{s}'(\xi) = \mathbf{0}.$$

Therefore, either we obtain

$$\mathbf{s}'(\xi) = \mathbf{0}$$

or $\exists$ an index $k \in 1, \ldots, p$ such that

$$\mathbf{s}'(\xi) = \alpha(\xi)\mathbf{r}_k(\mathbf{s}(\xi)), \qquad \lambda_k(\mathbf{s}(\xi)) = \xi.$$

If we differentiate the second equation with respect to ξ, we have

$$D\lambda_k(\mathbf{s}(\xi)) \cdot \mathbf{s}'(\xi) = 1,$$

substituting into the first equation, we have

$$\alpha(\xi)D\lambda_k(\mathbf{s}(\xi)) \cdot \mathbf{r}_k(\mathbf{s}(\xi)) = 1. \tag{3.98}$$

We cannot solve this equation if the kth characteristic field is linearly degenerate but only solvable if it is genuinely nonlinear. Normalizing the equation with

$$\alpha(\xi) = 1$$

Hence, we find either

$$\mathbf{s}'(\xi) = \mathbf{0}$$

or

$$\mathbf{s}'(\xi) = \mathbf{r}_k(\mathbf{s}(\xi)), \qquad \lambda_k(\mathbf{s}(\xi)) = \xi. \tag{3.99}$$

and $\mathbf{s}$ is therefore an integral curve of the field $\mathbf{r}_k$. Thus, assume that the kth characteristic field is genuinely nonlinear and that the function $\mathbf{s}$ is a solution of (3.98) with

$$\mathbf{s}(\lambda_k(\mathbf{u}_L)) = \mathbf{u}_L, \qquad \mathbf{s}(\lambda_k(\mathbf{u}_R)) = \mathbf{u}_R$$

From the above analysis the function

$$\mathbf{u}(x,t) = \begin{cases} \mathbf{u}_L & \text{if } \dfrac{x}{t} \leq \lambda_k(\mathbf{u}_L), \\ \mathbf{s}\left(\dfrac{x}{t}\right) & \text{if } \lambda_k(\mathbf{u}_L) \leq \dfrac{x}{t} \leq \lambda_k(\mathbf{u}_R), \\ \mathbf{u}_R & \text{if } \dfrac{x}{t} \geq \lambda_k(\mathbf{u}_R) \end{cases}$$

is a continuous self-similar weak solution of (3.96).

Definition 3.3.19 (k-centered simple wave). *The self-similar weak solution of (3.96) is called a k-centered simple wave or k-rarefaction wave connecting the $\mathbf{u}_L$ and $\mathbf{u}_R$ states.*

Definition 3.3.20 (k-Riemann invariant). *A smooth function $w : \Omega \to \mathbb{R}$ is called a k-Riemann invariant if it satisfies*

$$Dw(\mathbf{u}) \cdot \mathbf{r}_k(\mathbf{u}) = 0, \qquad \forall \mathbf{u} \in \Omega. \tag{3.100}$$

A k-Riemann invariant w is constant on a curve $\mathbf{v} : \xi \in \mathbb{R} \to \mathbf{v}(\xi) \in \mathbb{R}^d$ if and only if

$$\frac{d}{d\xi} w(\mathbf{v}(\xi)) = Dw(\mathbf{v}(\xi)) \cdot \mathbf{v}'(\xi) = 0, \tag{3.101}$$

which holds if $\mathbf{v}$ is an integral curve of $\mathbf{r}_k$

$$\mathbf{v}'(\xi) = \mathbf{r}_k(\mathbf{v}(\xi)). \tag{3.102}$$

The definition above implies that a k-Riemann invariant is constant along the trajectories of the vector field $\mathbf{r}_k$.

Remark 3.3.21. *When the kth field is linearly degenerate, λ_k is a k-Riemann invariant.*

Theorem 3.3.22. *On a k-rarefaction wave, all k-Riemann invariants w_k are constant, i.e., the relation*

$$w_k(\mathbf{u}_L) = w_k(\mathbf{u}_R) \tag{3.103}$$

is satisfied.

Proof. Let $\mathbf{u}$ be a $k-$rarefaction wave of the form the self-similar weak solution, let w be a $k-$Riemann invariant, let $\mathbf{s}(\frac{x}{t})$ be the integral curve of $\mathbf{r}_k$ which connects the left state $\mathbf{u}_L$ and the right state $\mathbf{u}_R$. For $t > 0$, the function $w(\mathbf{u}) : (x,t) \to w(\mathbf{u}(x,t))$ is continuous. To start with, $w(\mathbf{u})$ is constant for $\frac{x}{t} \leq \lambda_k(\mathbf{u}_L)$ and $\frac{x}{t} \geq \lambda_k(\mathbf{u}_R)$. If we derive w_k along $\mathbf{s}(\frac{x}{t})$ then we have

$$\nabla w_k(\mathbf{s}) \cdot \mathbf{s}'\left(\frac{x}{t}\right) = \nabla w_k(\mathbf{s}) \cdot \mathbf{r}_k\left(\mathbf{s}\left(\frac{x}{t}\right)\right) = 0. \tag{3.104}$$

This means that w_k is constant along the trajectories of the vector field $\mathbf{r}_k$. Also, for $\lambda_k(\mathbf{u}_L) \leq \frac{x}{t} \leq \lambda_k(\mathbf{u}_R)$, $\mathbf{u}$ is an integral curve of $\mathbf{r}_k$, $\mathbf{u}$ is continuous, w_k is a smooth function , then w_k is thus constant on the $k-$rarefaction wave and also for the left state $\mathbf{u}_L$ and the right state $\mathbf{u}_R$, which proves the result. ☐

4. Smoothed Particle Hydrodynamics Method

In this chapter, we will present introduction to the SPH method, the formulation, i.e., integral and particle approximation of SPH, we will ontline the consistency issues and present techniques to restore consistency at the particle level in the SPH method.

4.1 Introduction to SPH

The Smoothed Particle Hydrodynamics numerical method was formulated by Lucy, Gingold and Monaghan [48, 83] in the 70's. Originally, the SPH method was formulated for astrophysics problems, i.e., formation and evolution of proto-stars or galaxies. Unlike other numerical discretization techniques that discretizes continuum into a finite set of nodal points, SPH consolidates a set of discrete particles into a quasi-continuum. Because astrophysical particles moves collectively in a large scale which is comparable to the movement of a fluid, it is modeled by SPH as a quasi-fluid governed by the equations of classical hydrodynamics. In astrophysical applications the real process is discrete, a local continuous field is generated to connote the collective behavior of the discrete system to avoid singularities.

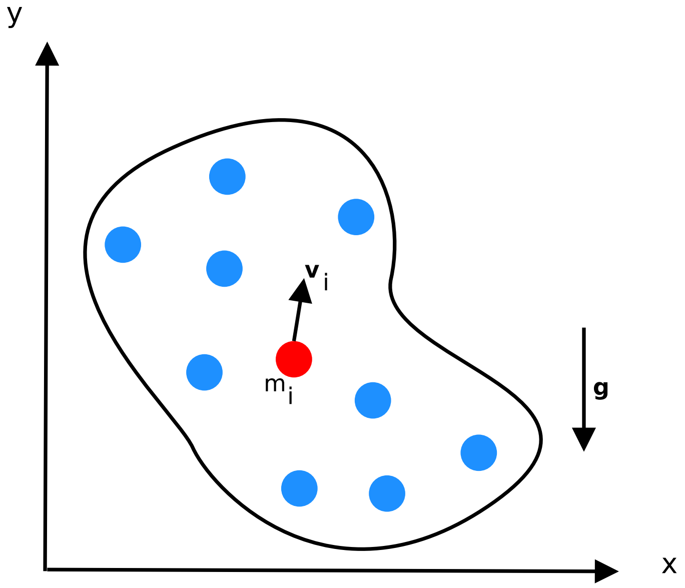


Figure 4.1: 2D particle discretization under gravity

4.2 Formulations of SPH

4.2.1 Kernel Approximation

The main idea behind the Smoothed Particle Hydrodynamics is that the method uses an integral representation of functions. Let us consider a function of $f(\mathbf{x})$ at any point $\mathbf{x} = (x, y, z)$ in space, the integral representation is given by

$$f(\mathbf{x}) = \int\limits_{\Omega} f(\mathbf{x}')\delta(\mathbf{x}-\mathbf{x}')d\mathbf{x}', \tag{4.1}$$

where $f(\mathbf{x}')$ denotes the neighboring function values, $\delta(\mathbf{x}-\mathbf{x}')$ is the Dirac delta function given by

$$\delta(\mathbf{x}-\mathbf{x}') = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{x}' \\ 0 & \text{if } \mathbf{x} \neq \mathbf{x}' \end{cases}$$

and Ω is the control volume of the integral that contains $\mathbf{x}$. We should note that the integral representation of a function as given in Equation (4.1) is exact since the Dirac delta function is used. However, it is difficult to use for numerical analysis from the mathematics point of view. The Dirac delta is strictly not a function because any real valued function that is equal to zero everywhere but a single point must have a total integral to be zero. Therefore, to this effect the Dirac delta function is replaced by a smooth kernel function $W(\mathbf{x}-\mathbf{x}', h)$ then the integral or kernel formulation reads

$$f_I(\mathbf{x}) := \int\limits_{\Omega} f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' \tag{4.2}$$

and it holds that $f_I(\mathbf{x}) \approx f(\mathbf{x})$, where W is the so-called *smoothing/kernel function*, h is called the *smoothing length* in SPH. The smoothing length defines the influence area of the smoothing function W, it controls the size of the compact support domain Ω called the *smoothing domain* or *influence domain*. This approximated integral can be termed as a finite integral representation which is conventionally called *kernel approximation*.

Kernel Approximation of Function Derivatives

We approximate the derivative of the function $f(\mathbf{x})$ by simply employing Equation (4.2) but replacing $f_I(\mathbf{x})$ by $\nabla \cdot f_I(\mathbf{x})$ which reads

$$\nabla \cdot f_I(\mathbf{x}) := \int\limits_{\Omega} \left[\nabla \cdot f(\mathbf{x}')\right] W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' \tag{4.3}$$

and it holds that $\nabla \cdot f_I(\mathbf{x}) \approx \nabla \cdot f(\mathbf{x})$. With the identity

$$\left[\nabla \cdot f(\mathbf{x}')\right] W(\mathbf{x}-\mathbf{x}', h) = \nabla \cdot \left[f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}', h)\right] - f(\mathbf{x}') \cdot \nabla W(\mathbf{x}-\mathbf{x}', h) \tag{4.4}$$

$$\nabla \cdot f_I(\mathbf{x}) \approx \int\limits_{\Omega} \nabla \cdot \left[f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}', h)\right] d\mathbf{x}' - \int\limits_{\Omega} f(\mathbf{x}') \cdot \nabla W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' \tag{4.5}$$

and applying the Gauss's theorem on the first integral in Equation (4.5)

$$\nabla \cdot f_I(\mathbf{x}) \approx \int\limits_{\partial\Omega} f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}', h) \cdot \mathbf{n} dS - \int\limits_{\Omega} f(\mathbf{x}') \cdot \nabla W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' \tag{4.6}$$

where $\mathbf{n}$ is the unit normal to the bounding surface.

With the requirement that W is compactly supported, the surface integral in equation (4.6) vanishes. Thus, we have

$$\nabla \cdot f_I(\mathbf{x}) \approx -\int\limits_{\Omega} f(\mathbf{x}') \cdot \nabla W(\mathbf{x}-\mathbf{x}', h) d\mathbf{x}' \tag{4.7}$$

From equation (4.7), it can be noted that the differentiation operator has transfered to the smoothing kernel function. So in short words from the above derivation, we can say that the SPH integral (kernel) approximation of the derivative of a function permits the gradient to be obtained from the combination of the function values and the derivative of the smoothing kernel function W, rather than the derivative of the function itself.

Proposition 4.2.1. *The SPH kernel approximation is second order accurate, or has $O(h^2)$ accuracy.*

Proof. We can show this by using the Taylor series expansion of $f(\mathbf{x}')$ around $\mathbf{x}$. From equation (4.2)

$$\begin{aligned} f_I(\mathbf{x}) &= \int\limits_{\Omega} \left[f(\mathbf{x}) + f'(\mathbf{x})(\mathbf{x}'-\mathbf{x}) + \text{HOT}((\mathbf{x}'-\mathbf{x})^2)\right] W(\mathbf{x}-\mathbf{x}', h) d\mathbf{x}' && (4.8) \\ &= f(\mathbf{x}) \int\limits_{\Omega} W(\mathbf{x}-\mathbf{x}', h) d\mathbf{x}' + f'(\mathbf{x}) \int\limits_{\Omega} (\mathbf{x}'-\mathbf{x}) W(\mathbf{x}-\mathbf{x}', h) d\mathbf{x}' + \text{HOT}((\mathbf{x}'-\mathbf{x})^2) && (4.9) \end{aligned}$$

where HOT stands for the residual higher order terms. We must take into consideration that the kernel function is an even function with respect to $\mathbf{x}$. This means that $(\mathbf{x}'-\mathbf{x})W(\mathbf{x}-\mathbf{x}', h)$ must be an odd function so therefore

$$\int\limits_{\Omega} (\mathbf{x}'-\mathbf{x}) W(\mathbf{x}-\mathbf{x}', h) d\mathbf{x}' = 0 \tag{4.10}$$

Making use of the normalization condition

$$\int\limits_{\Omega} W(\mathbf{x}-\mathbf{x}', h) d\mathbf{x}' = 1 \tag{4.11}$$

we obtained

$$f_I(\mathbf{x}) = f(\mathbf{x}) + \text{HOT}(h^2) \tag{4.12}$$

We can see that the integral or kernel approximation of a function is of second order accuracy. ☐

Remark 4.2.2. *The kernel approximation is not necessarily of second order accuracy if the smoothing function is not an even function, or if the normalization condition is not satisfied.*

4.2.2 Particle Approximation

In this section, the kernel (integral) approximations are transformed into discrete summations. Since SPH is a method that characterize the entire system by a finite number of particles which carry properties like mass, density, viscosity etc. and which occupy individual space. This is termed *particle approximation*. How do we perform this approximation?

Let the infinitesimal volume $d(\mathbf{x}')$ in the above kernel approximation be substituted by a finite volume of particle say ΔV_j. The mass of a particle j is given as

$$m_j = \rho_j \Delta V_j \tag{4.13}$$

ρ_j marks the density of particle j with $(j = 1, 2, \ldots, N)$ and N is the number of particles that is inside the support domain of the focal particle. From the integral approximation

$$\begin{aligned} f_I(\mathbf{x}) &\approx \int_\Omega f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' && (4.14)\\ &\approxeq \sum_{j=1}^{N} f(\mathbf{x}_j)W(\mathbf{x}-\mathbf{x}_j, h)\Delta V_j && (4.15)\\ &= \sum_{j=1}^{N} f(\mathbf{x}_j)W(\mathbf{x}-\mathbf{x}_j, h)\frac{1}{\rho_j}(\rho_j\Delta V_j) && (4.16)\\ &= \sum_{j=1}^{N} f(\mathbf{x}_j)W(\mathbf{x}-\mathbf{x}_j, h)\frac{1}{\rho_j}(m_j) && (4.17) \end{aligned}$$

Then we have

$$f_S(\mathbf{x}) := \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\mathbf{x}_j)W(\mathbf{x}-\mathbf{x}_j, h) \tag{4.18}$$

and it holds that $f_S(\mathbf{x}) \approx f(\mathbf{x})$, where $f_S(\mathbf{x})$ is the summation approximant at the particle level.

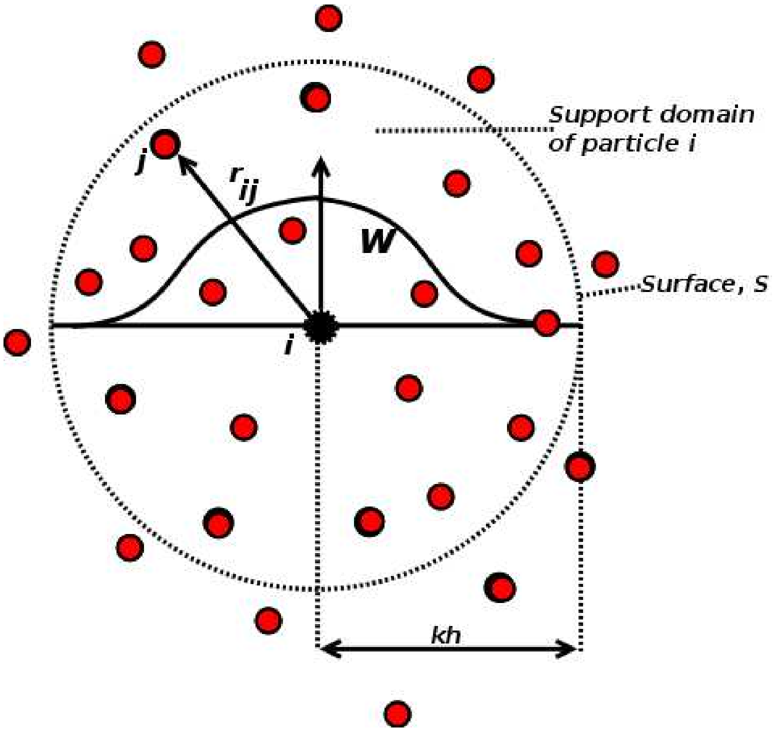


Figure 4.2: Illustration of the kernel function W, its support domain Ω, with boundary S. Neighboring particles j and focal particle i.

Similarly, the errors accrued in the summation interpolant is different, this is because approximations of integrals by summations over particles does not guarantee that these terms integrate exactly. From the summation interpolant evaluated at particle i, we do the Taylor series expansion of $f(\mathbf{x}_j) = f_j$ around

$\mathbf{x}_i$, we have

$$\sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j W_{ij} = f_i \sum_{j=1}^{N} \frac{m_j}{\rho_j} W_{ij} + \nabla f_i \cdot \sum_{j=1}^{N} \frac{m_j}{\rho_j} (\mathbf{x}_j - \mathbf{x}_i) W_{ij} + O(\mathbf{x}_j - \mathbf{x}_i)^2 \tag{4.19}$$

where $W_{ij} \equiv W(\mathbf{x}_i - \mathbf{x}_j, h)$.

Remark 4.2.3. *The summation interpolant is exact for constant functions only when the interpolant is normalized by dividing by the interpolation of unity.*

Particle Approximation of Function Derivatives

In the same spirit, the particle approximation for a function spatial derivative is given as

$$\nabla \cdot f_S(\mathbf{x}) := -\sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\mathbf{x}_j) \cdot \nabla W(\mathbf{x} - \mathbf{x}_j, h) \tag{4.20}$$

where we have taken the derivative of W with respect to the particle j, and thus we eventually have

$$\nabla \cdot f_S(\mathbf{x}_i) := -\sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\mathbf{x}_j) \cdot \nabla_i W_{ij} \tag{4.21}$$

and it holds that $\nabla \cdot f_S(\mathbf{x}_i) \approx \nabla \cdot f(\mathbf{x}_i)$, where

$$\nabla_i W_{ij} = \frac{\mathbf{x}_i - \mathbf{x}_j}{\mathbf{r}_{ij}} \frac{\partial W_{ij}}{\partial \mathbf{r}_{ij}} = \frac{\mathbf{x}_{ij}}{\mathbf{r}_{ij}} \frac{\partial W_{ij}}{\partial \mathbf{r}_{ij}}. \tag{4.22}$$

Equation (4.20) expresses that the value of the gradient of a function at particle i is approximated using the average of values of the function at all the neighboring particles in the support domain of particle i now weighted by the gradient of the smoothing function.

There are a number of approaches to derive the SPH formulation of PDEs. Benz [16] derived the SPH equations for PDEs by multiplying the terms by the smoothing function, integrate by parts over the volume and Taylor series expansions. Monaghan [88] employed a straight forward approach by placing the density inside the gradient operator, the identities are

$$\nabla \cdot f(\mathbf{x}) = \frac{1}{\rho} \left[\nabla \cdot (\rho f(\mathbf{x})) - f(\mathbf{x}) \cdot \nabla \rho \right] \tag{4.23}$$

$$\nabla \cdot f(\mathbf{x}) = \rho \left[\nabla \cdot \left(\frac{f(\mathbf{x})}{\rho} \right) + \frac{f(\mathbf{x})}{\rho^2} \cdot \nabla \rho \right] \tag{4.24}$$

These identities are now substituted into the integral representation (4.3) and evaluated at the particle i itself we obtain

$$\nabla \cdot f_S(\mathbf{x}_i) := \frac{1}{\rho} \left[\sum_{j=1}^{N} m_j \left[f(\mathbf{x}_j) - f(\mathbf{x}_i) \right] \cdot \nabla_i W_{ij} \right] \tag{4.25}$$

This approximation has the advantage that the first derivative of a constant function is zero exactly. This property is known as *zeroth-order consistency*

$$\nabla \cdot f_S(\mathbf{x}_i) := \rho_i \left[\sum_{j=1}^{N} m_j \left[\frac{f(\mathbf{x}_j)}{\rho_j^2} + \frac{f(\mathbf{x}_i)}{\rho_i^2} \right] \cdot \nabla_i W_{ij} \right] \tag{4.26}$$

A very good feature of these two equations is that the field function $f(\mathbf{x})$ appears in the form of paired particles. Equation (4.26) is not zeroth-order consistent, but has the advantage that forces leads to local momentum conservation because of the symmetric formulation.

A renormalized approximation introduced by Randles and Libersky [97], Bonet and Lok [19] and Vila [115] is given as

$$\nabla \cdot f_S(\mathbf{x}_i) := \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left[f(\mathbf{x}_j) - f(\mathbf{x}_i) \right] \mathbf{B}_i \cdot \nabla_i W_{ij} \tag{4.27}$$

where

$$\mathbf{B}_i = -\left[\sum_{j=1}^{N} \frac{m_j}{\rho_j} \mathbf{r}_{ij} \cdot \nabla_i W_{ij} \right]^{-1} \tag{4.28}$$

is the renormalized tensor where $\mathbf{r}_{ij} = \mathbf{r}_i - \mathbf{r}_j$ is the relative distance between particles i and j. Because, $\nabla_i W_{ij}$ is parallel to $\mathbf{r}_{ij}$, the renormalized tensor $\mathbf{B}$ is symmetric. This approach has the merit that it is *first-order consistent* which means by definition that the approximation can reproduce the gradient of a linear function exactly.

The errors arising from the gradient evaluation can be done in a similar fashion by expanding in Taylor series $f(\mathbf{x}')$ around $\mathbf{x}$, we have

$$\nabla f(\mathbf{x}) = \int_{\Omega} \left[f(\mathbf{x}) + (\mathbf{x}' - \mathbf{x})^{\alpha} \frac{\partial f}{\partial \mathbf{x}^{\alpha}} + \frac{1}{2} (\mathbf{x}' - \mathbf{x})^{\beta} (\mathbf{x}' - \mathbf{x})^{\gamma} \frac{\partial^2 f}{\partial \mathbf{x}^{\beta} \partial \mathbf{x}^{\gamma}} + O(\mathbf{x}' - \mathbf{x})^3 \right] \nabla W d\mathbf{x}' \tag{4.29}$$

$$\begin{aligned} &= f(\mathbf{x}) \int_{\Omega} \nabla W d\mathbf{x}' + \frac{\partial f}{\partial \mathbf{x}^{\alpha}} \int_{\Omega} (\mathbf{x}' - \mathbf{x})^{\alpha} \nabla W d\mathbf{x}' + \\ &\quad \frac{1}{2} \frac{\partial^2 f}{\partial \mathbf{x}^{\beta} \partial \mathbf{x}^{\gamma}} \int_{\Omega} (\mathbf{x}' - \mathbf{x})^{\beta} (\mathbf{x}' - \mathbf{x})^{\gamma} \nabla W d\mathbf{x}' + O[(\mathbf{x}' - \mathbf{x})^3], \end{aligned} \tag{4.30}$$

$$= \nabla f(\mathbf{x}) + \frac{1}{2} \frac{\partial^2 f}{\partial \mathbf{x}^{\beta} \partial \mathbf{x}^{\gamma}} \int_{\Omega} (\mathbf{x}' - \mathbf{x})^{\beta} (\mathbf{x}' - \mathbf{x})^{\gamma} \nabla W d\mathbf{x}' + O[(\mathbf{x}' - \mathbf{x})^3], \tag{4.31}$$

where we have used the fact that

$$\int_{\Omega} \nabla W d\mathbf{x}' = 0 \tag{4.32}$$

holds for even kernels and the second term integrates to unity, satisfying the normalization condition for even kernels. The errors in the interpolant for the gradient are therefore also second order, $O(h^2)$.

For the gradient summation interpolant, we Taylor series expand $f(\mathbf{x}_j) = f_j$ around $\mathbf{x}_i$, $f(\mathbf{x}_i) = f_i$, thus we have

$$\nabla f_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j \nabla_i W_{ij} \tag{4.33}$$

$$= f_i \sum_{j=1}^{N} \frac{m_j}{\rho_j} \nabla_i W_{ij} + \frac{\partial f_i}{\partial \mathbf{x}^{\alpha}} \sum_{j=1}^{N} \frac{m_j}{\rho_j} (\mathbf{x}_j - \mathbf{x}_i)^{\alpha} \nabla_i W_{ij} + \tag{4.34}$$

$$\frac{1}{2} \frac{\partial^2 f_i}{\partial \mathbf{x}^{\beta} \partial \mathbf{x}^{\gamma}} \sum_{j=1}^{N} \frac{m_j}{\rho_j} (\mathbf{x}_j - \mathbf{x}_i)^{\beta} (\mathbf{x}_j - \mathbf{x}_i)^{\gamma} \nabla_i W_{ij} + O[(\mathbf{x}_j - \mathbf{x}_i)^3] \tag{4.35}$$

which is also second order. The Greek indices α, β and γ denotes the coordinate directions, with repeated indices meaning Einstein summation.

Particle Approximation of the Laplacian

Second order differential operators called Laplacian are inevitable when dealing with modeling involving viscous terms, diffusion terms, temperature and to express the pressure Poisson equation when dealing with incompressible techniques. In the work of Fatehi and Manzari [45], they outlined some Laplacian approximations. To approximate the second derivative $\nabla^2 f(\mathbf{x})$ in the spirit of SPH, if we apply the approximant defined earlier for the first derivation we will obtain the particle approximation for the Laplacian

$$\nabla^2 f_S(\mathbf{x}_i) := \sum_{j=1}^{N} \frac{m_j}{\rho_j}(\nabla \cdot f_{S,j}) \cdot \nabla_i W_{ij}, \tag{4.36}$$

$$\nabla^2 f_S(\mathbf{x}_i) := \sum_{j=1}^{N} \frac{m_j}{\rho_j}\left[(\nabla \cdot f_{S,j} - \nabla \cdot f_{S,i})\right] \cdot \nabla_i W_{ij}, \tag{4.37}$$

or alternatively we have

$$\nabla^2 f_S(\mathbf{x}_i) := \sum_{j=1}^{N} \frac{m_j}{\rho_j}\left[(\nabla \cdot f_{S,j} + \nabla \cdot f_{S,i})\right] \cdot \nabla_i W_{ij}. \tag{4.38}$$

where $\nabla \cdot f_{S,i}$ and $\nabla \cdot f_{S,j}$ denotes the first order particle approximation for particle i and j. The above schemes were employed to incorporate physical viscosity in astrophysical problems [47, 116] and to solve the two dimensional heat conduction problems [58] and to study low-Reynolds number incompressible flows [19]. Flebbe et al. [47] and Watkins et al. [116] used the approximation for astrophysical problems and stated some non-physical oscillations in their solution. Using these approximations in a heat-like equation with discontinuous initial data, essentially leads to oscillatory solutions, see [44].

Another approach to construct the Laplacian is to use the second derivative of the kernel function

$$\nabla^2 f_S(\mathbf{x}_i) := \sum_{j=1}^{N} \frac{m_j}{\rho_j} f(\mathbf{x}_j)\nabla \cdot \nabla_i W_{ij}, \tag{4.39}$$

$$\nabla^2 f_S(\mathbf{x}_i) := \sum_{j=1}^{N} \frac{m_j}{\rho_j}(f(\mathbf{x}_j) - f(\mathbf{x}_i))\nabla \cdot \nabla_i W_{ij}, \tag{4.40}$$

or alternatively

$$\nabla^2 f_S(\mathbf{x}_i) := \sum_{j=1}^{N} \frac{m_j}{\rho_j}(f(\mathbf{x}_j) + f(\mathbf{x}_i))\nabla \cdot \nabla_i W_{ij}. \tag{4.41}$$

These formulations were used by Takeda et al. [104] and Chaniotis et al. [34].

Since this involves the second derivative of the kernel function, the inflection point of the kernel function must be farther than the nearest particle to have a monotonicity preserving. But because general kernel functions have inflection points between 0 and h however in a complicated flow problem this condition is hard to satisfy. This is only valid for ordered particles, see [44].

Another scheme by Brookshaw [21] uses a finite-difference-like form for the first order derivative and uses a SPH summation for the second order derivative. This form is given as

$$\nabla^2 f_S(\mathbf{x}_i) := \sum_{j=1}^{N} 2\frac{m_j}{\rho_j}\frac{f(\mathbf{x}_j) - f(\mathbf{x}_i)}{r_{ji}}\mathbf{n}_{ji} \cdot \nabla_i W_{ij}, \tag{4.42}$$

where $r_{ji} = |\mathbf{r}_{ji}|$ and $\mathbf{n}_{ji} = \frac{\mathbf{r}_{ji}}{r_{ji}}$ is the unit vector in the interparticle direction. Basa et al. [11] in their work showed that this approximation is the best among available SPH approximation of second derivatives.

Convergence and Accuracy of Particle Approximation

The convergence and accuracy of SPH approximations is largely connected to the particle distribution. Because SPH is a truly meshless method with a Lagrangian description, particles move in the computational domain and their location changes in time, hence it hard to ensure a continuous convergence rate and accuracy. Mas-Gallic and Raviart [85] showed that convergence in SPH should satisfy the conditions

$$\frac{h}{\Delta x} \to \infty \tag{4.43}$$

and

$$h \to 0 \tag{4.44}$$

with Δx given as the distance between particle i and j. Practically, the second condition is mostly used. Generally, the convergence order of SPH is of first order.

However, there are techniques available to improve the order of accuracy of SPH approximation. The Shepard normalization process, tries to ensure that the equality sums to 1 at the particle and discrete level.

$$\sum_{j=1}^{N} \frac{m_j}{\rho_j} W(\mathbf{x} - \mathbf{x}', h) = 1 \tag{4.45}$$

A term referred to as *partition of unity*. This is achieved by correcting the kernel function by

$$\widetilde{W}(\mathbf{x} - \mathbf{x}', h) = \frac{W(\mathbf{x} - \mathbf{x}', h)}{\sum_{j=1}^{N} \frac{m_j}{\rho_j} W(\mathbf{x} - \mathbf{x}', h)} \tag{4.46}$$

where $\widetilde{W}(\mathbf{x} - \mathbf{x}', h)$ is the Shepard corrected kernel. This correction ensures first order convergence rate of the SPH approximation.

Another technique for improving the convergence order and accuracy is the renormalization technique given by Randles and Libersky [97]. The aim of this method is to ensure the equality

$$\sum_{j=1}^{N} \frac{m_j}{\rho_j} \nabla W(\mathbf{x} - \mathbf{x}', h) = 0. \tag{4.47}$$

which we can term as *partition of zero*. And the approximation is corrected by

$$\nabla \cdot f_S(\mathbf{x}_i) := \sum_{j=1}^{N} \frac{m_j}{\rho_j} \left[f(\mathbf{x}_j) - f(\mathbf{x}_i)\right] \mathbf{B}_i \cdot \nabla_i W_{ij} \tag{4.48}$$

where $\mathbf{B}_i$ as stated earlier is the renormalization matrix.

4.3 Smoothing Functions

The smoothing kernel $W(\mathbf{x}-\mathbf{x}', h)$ is such that it satisfies some properties as summarized in the book of Liu and Liu [80] namely :

- **Unity:**
  The smoothing kernel function must be normalized over its support domain

  $$\int_{\Omega} W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' = 1 \qquad \text{for all } \mathbf{x} \text{ and } h > 0. \tag{4.49}$$

  This unity condition, assures the zeroth-order consistency (C^0) of the integral representation of the function at the continuous level. But this condition does not necessarily guarantee (C^0) consistency of the discrete approximation.

- **Compact support:**
  The smoothing kernel function must be compactly supported

  $$W(\mathbf{x}-\mathbf{x}', h) = 0 \quad \text{for} \quad |\mathbf{x}-\mathbf{x}'| > kh \tag{4.50}$$

  where the scaling factor k determines the spread of the smoothing function.

  The support domain of a particle at point $\mathbf{x}$ is given by $|\mathbf{x}-\mathbf{x}'| \leq kh$. This is very important to the SPH method because it enables the approximation to be done from a local representation of particles inside the smoothing domain Ω for which W is nonzero.

- **Positivity:**
  The smoothing kernel must be positive i.e.

  $$W(\mathbf{x}-\mathbf{x}', h) \geq 0 \qquad \text{for all } \mathbf{x}, \mathbf{x}' \text{ and } h > 0. \tag{4.51}$$

  for any point at $\mathbf{x}'$ within the support domain of particle at point $\mathbf{x}$. This requirement may not be necessarily mathematical as a function, however it is important from a physical point of view of phenomena. For instance, density can never be negative in fluid dynamics problems etc.

- **Decay:**
  The smoothing kernel W should be monotonically decreasing with the increase of the distance away from the target particle. This requirement explains that a force exerted by a particle on another particle decreases with the increase in distance between the two particles.

- **Dirac delta function property:**
  The smoothing function should satisfy the Dirac delta function property as the smoothing length tends to zero.

  $$\lim_{h\to 0} W(\mathbf{x}-\mathbf{x}', h) = \delta(\mathbf{x}-\mathbf{x}') \qquad \text{for all } \mathbf{x}, \mathbf{x}', \tag{4.52}$$

  where

  $$\delta(\mathbf{x}-\mathbf{x}') = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{x}' \\ 0 & \text{if } \mathbf{x} \neq \mathbf{x}' \end{cases}$$

  Though this condition enables us to observe explicitly that the SPH method converges to its exact form.

- **Symmetric property:**
  The smoothing function W should be an even function. This property explains that particles with the same distance from a target or focal particle but different positions should have equal effects on the target particle.

$$W(\mathbf{x}-\mathbf{x}',h) = W(\mathbf{x}'-\mathbf{x},h) \qquad \text{for all } h > 0 \tag{4.53}$$

- **Smoothness:**
  The kernel function W and its derivatives W' should be continuous and sufficiently smooth. This requirement is important to obtain good approximations. Because particle disorder will always occur at certain levels in SPH simulations, a continuous and smooth function is not very sensitive to this kind of disorders and this keep the errors in approximating the kernel interpolations small.

We can employ any function that possesses the above named property in the SPH method as our smoothing function. We will mention some smoothing functions

Gaussian Kernel

In the work of Monaghan [88], he stated that in finding a physical interpretation of an SPH equation, it is best to assume the smoothing function as Gaussian. The following Gaussian kernel has been used by Gingold and Monaghan [48] to simulate non-spherical stars

$$W(q,h) = K_d e^{-q^2} \tag{4.54}$$

The gradient is given as

$$W'(q,h) = -2K_d q e^{-q^2}$$

To satisfy the normalization requirement, the normalization constant K_d is given as $\frac{1}{\pi^{1/2}h}$, $\frac{1}{\pi h^2}$ and $\frac{1}{\pi^{3/2}h^3}$ in one-, two- and three- dimensional space respectively. In equation (4.54), q is the relative distance between two particles at positions $\mathbf{x}$ and $\mathbf{x}'$, $q = \frac{r}{h} = \frac{|\mathbf{x}-\mathbf{x}'|}{h}$, where r is the distance between the two particles. Though, the Gaussian kernel is sufficiently smooth even for higher orders derivatives. The downside of the Gaussian kernel is that it is not really compact, i.e., it never goes to zero theoretically except q goes to infinity. Also, it is computationally expensive to use since it will take more distance for the kernel function to approach zero. Thus, resulting in a larger support domain for particle approximation which invariably costs more time to solve.

Cubic Spline Kernel

This kernel function has been mostly widely used in SPH because it resembles the Gaussian function and most importantly, it has a more narrower compact support. So, computationally, this is optimal. We have used the cubic spline in our numerical experiments presented in this thesis. The cubic spline function is given as

$$W(q,h) = K_d \times \begin{cases} 1 - \dfrac{3}{2}q^2 + \dfrac{3}{4}q^3 & \text{for } \ 0 \le q \le 1, \\ \dfrac{1}{4}(2-q)^3 & \text{for } \ 1 < q \le 2, \\ 0 & \text{for } \ q > 2 \end{cases} \tag{4.55}$$

The normalization condition provides the constants $K_d = \frac{2}{3h}, \frac{10}{7\pi h^2}$ and $\frac{1}{\pi h^3}$ in one-, two- and three-dimensional space respectively. Its derivative is given by

$$W'(q,h) = K_d \times \begin{cases} -3q + \dfrac{9}{4}q^2 & \text{for } \ 0 \le q \le 1, \\ -\dfrac{3}{4}(2-q)^2 & \text{for } \ 1 < q \le 2, \\ 0 & \text{for } \ q > 2 \end{cases} \tag{4.56}$$

Since we have used the cubic spline in this thesis, we provide the plot and the derivative in one dimension in Figure 4.3

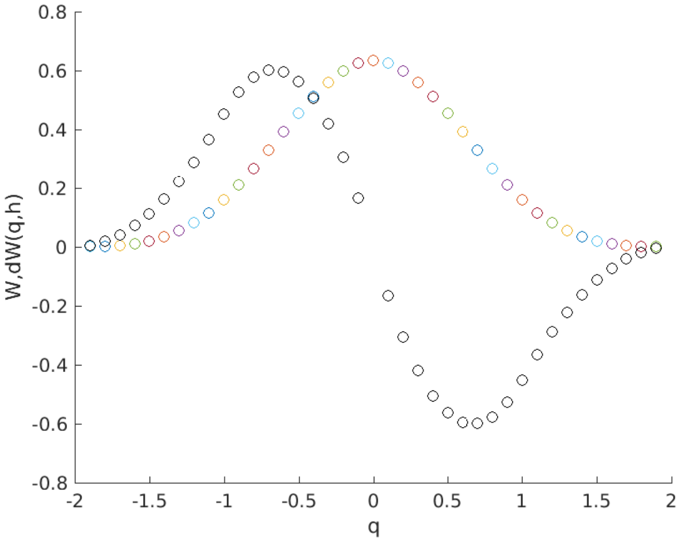


Figure 4.3: The cubic spline kernel and its first derivative, the cubic spline function in coloured dots and its first derivative in black dots.

Quartic Spline Kernel

The spline kernel of order 4 called the quartic spline kernel is given as

$$W(q,h) = K_d \times \begin{cases} \left(\dfrac{5}{2}-q\right)^4 - 5\left(\dfrac{3}{2}-q\right)^4 + 10\left(\dfrac{1}{2}-q\right)^4 & \text{for } \ 0 \le q \le 0.5, \\ \left(\dfrac{5}{2}-q\right)^4 - 5\left(\dfrac{3}{2}-q\right)^4 & \text{for } \ 0.5 < q \le 1.5, \\ \left(\dfrac{5}{2}-q\right)^4 & \text{for } \ 1.5 < q \le 2.5, \\ 0 & \text{for } \ q > 2.5 \end{cases} \tag{4.57}$$

with the normalization constant $K_d = \frac{1}{24h}, \frac{96}{1199\pi h^2}$ and $\frac{1}{20\pi h^3}$ in one-, two- and three- dimensional space respectively. Its derivative gives

$$W'(q,h) = -4K_d \times \begin{cases} \left(\frac{5}{2} - q\right)^3 - 5\left(\frac{3}{2} - q\right)^3 + 10\left(\frac{1}{2} - q\right)^3 & \text{for } 0 \le q \le 0.5, \\ \left(\frac{5}{2} - q\right)^3 - 5\left(\frac{3}{2} - q\right)^3 & \text{for } 0.5 < q \le 1.5, \\ \left(\frac{5}{2} - q\right)^3 & \text{for } 1.5 < q \le 2.5, \\ 0 & \text{for } q > 2.5 \end{cases} \tag{4.58}$$

Quintic Spline Kernel

In the same spirit, the spline kernel of order 5 called the quintic spline kernel is given as

$$W(q,h) = K_d \times \begin{cases} (3-q)^5 - 6(2-q)^5 + 15(1-q)^5 & \text{for } 0 \le q \le 1, \\ (3-q)^5 - 6(2-q)^5 & \text{for } 1 < q \le 2, \\ (3-q)^5 & \text{for } 2 < q \le 3, \\ 0 & \text{for } q > 3 \end{cases} \tag{4.59}$$

with the normalization constant $K_d = \frac{1}{120h}, \frac{7}{478\pi h^2}$ and $\frac{1}{120\pi h^3}$ in one-, two- and three- dimensional space respectively. Its derivative gives

$$W'(q,h) = -5K_d \times \begin{cases} (3-q)^4 - 6(2-q)^4 + 15(1-q)^4 & \text{for } 0 \le q \le 1, \\ (3-q)^4 - 6(2-q)^4 & \text{for } 1 < q \le 2, \\ (3-q)^4 & \text{for } 2 < q \le 3, \\ 0 & \text{for } q > 3 \end{cases} \tag{4.60}$$

Wendland Kernel

We wish to mention the Wendland kernels [117], they are based upon one algebraic equation so they are advantageously uniform and they are compactly supported. A Wendland kernel of order 4 is written as

$$W(q,h) = K_d \times \begin{cases} \left(1 - \frac{q}{2}\right)^4 (1+2q) & \text{for } 0 \le q \le 2, \\ 0 & \text{for } q > 2 \end{cases} \tag{4.61}$$

with normalization constant $K_d = \frac{3}{4h}, \frac{7}{4\pi h^2}$ and $\frac{21}{16\pi h^3}$ in one-, two- and three- dimensional space respectively. Its derivative is given by

$$W'(q,h) = K_d \times \begin{cases} -5q\left(1 - \frac{q}{2}\right)^3 & \text{for } 0 \le q \le 2, \\ 0 & \text{for } q > 2 \end{cases} \tag{4.62}$$

Consistency of SPH approximation

In numerical analysis, after performing Taylor series expansions on some smooth functions for the finite difference method, we say the approximation is *consistent* to a given order. Consistency is defined as follows by Strikwerda [103]

Definition 4.3.1 (Consistency). *A scheme $L_h u = f$ that is consistent with the differential equation $Lu = f$ is accurate (consistent) of order p if for any sufficiently smooth function v*

$$Lv - L_h v = O(h^p) \tag{4.63}$$

where p is the order of consistency.

It is necessary that $p > 0$ for convergence, and generally we require that $p \geq 1$, h is a parameter that depicts the mesh size.

The conditions of consistency, stability and convergence are related to each other, and we can find this relation in the fundamental *Equivalence Theorem of Lax*, the proof can be found in the classical book of Richtmyer and Morton [98]. *For a well-posed initial value problem and a consistent discretization scheme, stability is the necessary and sufficient condition for convergence.* This theorem shows that in order to analyze a time-dependent problem two tasks have to be performed:

- Analyze the consistency condition; this leads to the determination of the order of accuracy of the scheme and its truncation error.
- Analyze the stability properties; this leads to detailed information on the frequency distribution of the error.

And from the two steps convergence can be concluded without further analysis. Similarly, the consistency concept in finite element methods, for a FEM approximation to converge, the solution must approach the exact solution when the nodal distance approaches zero. To ensure convergence, the FEM shape function must satisfy a certain degree of consistency. The degree of consistency is characterized by the order of the polynomial that can be exactly reproduced by the approximation using the shape function. If an approximation can reproduce a constant exactly, the approximation is said to have zero-th order or C^0 consistency. In general, if an approximation can reproduce a polynomial of up to k-th order exactly, the approximation is said to have k-th order or C^k consistency.

Following the consistency idea from FEM, we argue that for an SPH kernel approximation to exactly reproduce a function, the smoothing function should satisfy some conditions, which we represent by the *polynomial reproducibility* of the kernel approximation.

Definition 4.3.2 (Reproducibility condition, completeness). *An approximation $f_S(\mathbf{x})$ is complete to order k if any polynomial up to order k can be represented exactly. Then, $f_S(\mathbf{x})$ is given by*

$$f_S(\mathbf{x}) = \sum_{j=1}^{N} f(\mathbf{x}_j) W(\mathbf{x} - \mathbf{x}_j, h). \tag{4.64}$$

If $f(\mathbf{x}_j)$ are given by a polynomial of order k, the approximation $f_S(\mathbf{x})$ should reproduce the polynomial exactly if the approximation is complete to order k.

Consistency of Integral Representation

Proposition 4.3.3 (**Constant consistency**). *For a constant (zero order polynomial) field function* $f(\mathbf{x}) = a$ *to be exactly reproduced by the SPH kernel approximation, we should have*

$$f(\mathbf{x}) = \int\limits_{\Omega} aW(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' = a \tag{4.65}$$

which implies

$$\int\limits_{\Omega} W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' = 1 \tag{4.66}$$

Clearly, we can see that the normalization condition is in fact the condition for the kernel approximation to have zero order consistency.

Proposition 4.3.4 (**Linear consistency**). *For a linear (first order polynomial) field function* $f(\mathbf{x}) = a\mathbf{x} + b$ *to be exactly reproduced by the SPH kernel approximation, we should have*

$$f(\mathbf{x}) = \int\limits_{\Omega} (a\mathbf{x}' + b)W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' = a\mathbf{x} + b \tag{4.67}$$

which given the normalization condition in equation (4.66) leads to

$$\int\limits_{\Omega} \mathbf{x}'W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' = \mathbf{x} \tag{4.68}$$

Proof. Multiplying $\mathbf{x}$ to both sides of equation (4.66), we have

$$\int\limits_{\Omega} \mathbf{x}W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' = \mathbf{x} \tag{4.69}$$

Substracting equation (4.68) from the above equation leads to

$$\int\limits_{\Omega} (\mathbf{x}-\mathbf{x}')W(\mathbf{x}-\mathbf{x}', h)d\mathbf{x}' = 0 \tag{4.70}$$

For equation (4.70) to be satisfied, the smoothing function must be even (symmetric) so that the first moment can vanish. ☐

To reproduce higher order polynomials, similar conclusions are derived [12].

Consistency of Particle Approximation

At the particle level, the above propositions derived from the continuous form of integral representation do not ensure consistency for the discrete form after particle approximation. The phenomenom where the discretized equations for conditions (4.66) and (4.70) are not satisfied is termed *particle inconsistency* [13, 92]. This means that particle approximation may not satisfy exactly the zero-th order consistency

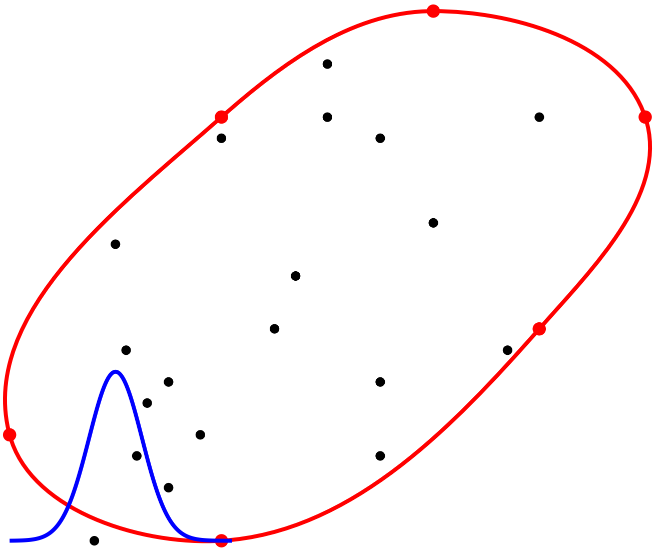


Figure 4.4: SPH behaviour at and near the boundaries.

condition near boundaries or for non-uniform particle distribution. The constant and linear consistency conditions are given as

$$\sum_{j=1}^{N} \frac{m_j}{\rho_j} W(\mathbf{x}-\mathbf{x}_j, h) = 1 \tag{4.71}$$

and

$$\sum_{j=1}^{N} \frac{m_j}{\rho_j} (\mathbf{x}-\mathbf{x}_j) W(\mathbf{x}-\mathbf{x}_j, h) = 0 \tag{4.72}$$

where N is the total number of particles in support domain for the focal particle $\mathbf{x}$. Equations (4.71) and (4.72) are not always satisfied. A very simple explanation can be seen pictorially in Fig. 4.4 for particles near or at the boundary of the computational domain where the support domain intersects with the boundary. Also, when particles are non-uniformly distributed, in this case even for interior particles whose support domain are not truncated, the constant and linear consistency conditions at the particle level will not be exactly satisfied, this is due to the unbalanced particle contribution. Generally the conditions are less than unity and will not vanish.

The reproducing conditions can be written as given below: Let the nodal values be represented by a polynomial, that is,

$$f_j = a_0 + a_1 x_j + a_2 x_j^2 + \cdots + a_k x_j^k \tag{4.73}$$

Then, the reproducing conditions (completeness) of order k are satisfied if

$$f_S(x) = \sum_{j=1}^{N} f_j W(x - x_j, h) = a_0 + a_1 x_j + a_2 x_j^2 + \cdots + a_k x_j^k \tag{4.74}$$

Because this equation must hold for arbitrary a_i, then we have the following conditions in one dimension:

$$\begin{aligned}
\sum_{j=1}^{N} W(x-x_j,h) &= 1 && (4.75)\\
\sum_{j=1}^{N} x_j W(x-x_j,h) &= x && (4.76)\\
\sum_{j=1}^{N} x_j^2 W(x-x_j,h) &= x^2 && (4.77)\\
&\dots\\
\sum_{j=1}^{N} x_j^k W(x-x_j,h) &= x^k && (4.78)
\end{aligned}$$

In two spatial dimensions, the linear reproducing conditions is given as

$$\begin{aligned}
\sum_{j=1}^{N} W(\mathbf{x}-\mathbf{x}_j,h) &= 1 && (4.79)\\
\sum_{j=1}^{N} x_j W(\mathbf{x}-\mathbf{x}_j,h) &= x && (4.80)\\
\sum_{j=1}^{N} y_j W(\mathbf{x}-\mathbf{x}_j,h) &= y && (4.81)
\end{aligned}$$

where given in indices notation

$$\sum_{j=1}^{N} x_{j\alpha} W(\mathbf{x}-\mathbf{x}_j,h) = x_\alpha \qquad (4.82)$$

and $x_0 = x_{j0} = 1$, $x_1 = x$, $x_2 = y$, $x_{j1} = x_j$, $x_{j2} = y_j$. Likewise, requiring that the derivatives of a polynomial is correctly reproduced. In two dimensions, the derivative reproducing conditions for a linear field area are given as

$$\begin{aligned}
\sum_{j=1}^{N} W_{,x}(\mathbf{x}-\mathbf{x}_j,h) &= 0, & \sum_{j=1}^{N} W_{,y}(\mathbf{x}-\mathbf{x}_j,h) = 0 && (4.83)\\
\sum_{j=1}^{N} x_j W_{,x}(\mathbf{x}-\mathbf{x}_j,h) &= 1, & \sum_{j=1}^{N} x_j W_{,y}(\mathbf{x}-\mathbf{x}_j,h) = 0 && (4.84)\\
\sum_{j=1}^{N} y_j W_{,x}(\mathbf{x}-\mathbf{x}_j,h) &= 0, & \sum_{j=1}^{N} y_j W_{,y}(\mathbf{x}-\mathbf{x}_j,h) = 1 && (4.85)
\end{aligned}$$

These are the derivative of equations (4.80), (4.81), (4.81). In general, the linear derivative reproducing conditions can be written as

$$\begin{aligned}
\sum_{j=1}^{N} W_{,k}(\mathbf{x}-\mathbf{x}_j,h) &= 0 && (4.86)\\
\sum_{j=1}^{N} x_{jm} W_{,k}(\mathbf{x}-\mathbf{x}_j,h) &= \delta_{km} && (4.87)
\end{aligned}$$

Correction Techniques

There are ways to restore consistency conditions at the particle level using kernel correction functions. Through a correction transformation, completeness in kernel approximations can be restored. The transformation is developed by enforcing the reproducibility conditions. Completeness correction of two types have evolved

- Completeness corrections of the kernel functions
- Completeness corrections of the derivatives of the kernel functions

Shepard [101] gave a correction for approximating the kernel function, this correction provides for constant reproducing conditions. Belytschko et al. [14] gave a complete correction that ensures for linear reproducing conditions. And for the correction of the derivatives the following approaches exists

- Symmetrization technique [86]
- Johnson and Beissel correction [60]
- Randles and Libersky renormalization [97]
- Correction by Krongauz and Belytschko [68]
- Chen and Beraun correction [35]
- Liu and Liu correction [81]

Correction of the Kernel Function

We need to correct the kernel function so as to ensure completeness in the kernel approximation. The resulting correction leads to the required reproducibility conditions and its derivative leads to corresponding derivative reproducibility conditions. We start with *constant completeness,* an approximation used in fitting data by Shepard [101] is given as

$$f_S(\mathbf{x}) := \sum_{j=1}^{N} f(\mathbf{x}')\widetilde{W}(\mathbf{x}-\mathbf{x}', h) \tag{4.88}$$

where

$$\widetilde{W}(\mathbf{x}-\mathbf{x}', h) = \frac{W(\mathbf{x}-\mathbf{x}', h)}{\sum_{j=1}^{N} \frac{m_j}{\rho_j} W(\mathbf{x}-\mathbf{x}', h)} \tag{4.89}$$

Proposition 4.3.5. *The Shepard functions reproduce constant functions.*

Proof. Let $f_j = c$ for all $j = 1 \cdots N$, then

$$\begin{aligned} f_S(\mathbf{x}) &= \sum_{j=1}^{N} f(\mathbf{x}')\widetilde{W}(\mathbf{x}-\mathbf{x}', h) && (4.90) \\ &= \sum_{j=1}^{N} c \cdot \widetilde{W}(\mathbf{x}-\mathbf{x}', h) && (4.91) \\ &= \frac{\sum_{j=1}^{N} W(\mathbf{x}-\mathbf{x}', h)}{\sum_{j=1}^{N} W(\mathbf{x}-\mathbf{x}', h)} c && (4.92) \\ &= c && (4.93) \end{aligned}$$

which completes the proof. □

Clearly, the Shepard functions will also correctly reproduce the gradient of a constant function, i.e., $\nabla f_S(\mathbf{x}) = 0$ and thus the derivative of a constant function is reproduced exactly. Shepard approximants are advantageous because they can be computed at a relatively low cost.

For linear completeness. This can be achieved by either using a moving least-squares (MLS) approximation or by adopting a correction to the kernel function. We will highlight the second approach, modifying the approximation by a linear function $\beta_\alpha(\mathbf{x})\mathbf{x}_\alpha$, i.e., define

$$f_S(\mathbf{x}) = \sum_{j=1}^{N} f_j \Phi_j(\mathbf{x}) \tag{4.94}$$

where

$$\Phi_j(\mathbf{x}) = \beta_\alpha(\mathbf{x})\mathbf{x}_{j\alpha} W(\mathbf{x}-\mathbf{x}', h) \tag{4.95}$$

The reproducing conditions yield the equations for the coefficients $\beta_\alpha(\mathbf{x})$

$$\sum_{j=1}^{N} (\mathbf{x}_{j\alpha} W(\mathbf{x}-\mathbf{x}', h)\mathbf{x}_{j\gamma})\beta_\alpha(\mathbf{x}) = \delta_{\alpha\gamma} \tag{4.96}$$

where for instance in two dimensions, the equation for $\beta_\alpha(\mathbf{x})$ are given by

$$\boldsymbol{\beta}(0) = \left(\sum_{j=1}^{N} W(\mathbf{x}-\mathbf{x}', h) \begin{bmatrix} 1 & x_j - x & y_j - y \\ x_j - x & (x_j - x)^2 & (x_j - x)(y_j - y) \\ y_j - y & (x_j - x)(y_j - y) & (y_j - y)^2 \end{bmatrix} \right)^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \tag{4.97}$$

By shifting the origin improves the conditioning of the matrix and thus reduces roundoff error. By construction, this approximation produces linear functions.

Correction of the Derivative of Kernel Function

We can correct the derivatives of the kernel functions, we do this by modifying the approximation to the derivatives. Derivative completeness corrections are implemented through linear transformations of the original derivatives. To achieve zero-th order completeness, the approximation must be able to reproduce the exact derivative for a constant function, that is if $f_j = 1$ for all j, then $\nabla f_S(\mathbf{x}) = 0$.

Monaghan proposed the simplest correction for the derivatives, the symmetrization technique. It is a skew symmetric form. It has been widely used in the momentum equation for stability purposes. This modification applied to standard SPH approximation for the gradient of a function gives

$$\nabla f_S(\mathbf{x}_i) := -\sum_{j=1}^{N} \frac{m_j}{\rho_j}(f_j - f_i)\nabla W_{ij} \tag{4.98}$$

Clearly we can see that when $f(\mathbf{x})$ is a constant, the $f_j = f_i$ for all i, j and the derivatives must vanish. When we observe the zero-th order approximation correction of Shepard, the Shepard correction is at any point in the domain while the symmetrization correction technique of Monaghan can only be imposed at the nodal points. Even more, in the Shepard corrections, the derivatives are integrable because they arise from a function that is well-defined. The symmetrization technique modifies the derivatives of an approximation, and thus they are usually not integrable.

In this thesis, we have made use of the Krongauz and Belytschko correction [68]. They developed a correction whereby the derivatives are reproduced exactly for any constant and linear functions. These corrected derivatives are called *pseudo-derivatives* because the corrected derivatives are not integrable. The corrected derivatives are given as $G_{ij}(\mathbf{x})$

$$\nabla f_S(\mathbf{x}) = \sum_{j=1}^{N} f_j G_{ij} \tag{4.99}$$

where G_{ij} are linear combinations of the derivatives of the Shepard functions, i.e.,

$$\mathbf{G}_j = \boldsymbol{\alpha} \cdot \nabla \widetilde{W}(\mathbf{x} - \mathbf{x}', h) \tag{4.100}$$

In two space dimensions, we have

$$G_{jx} = \alpha_{11}\widetilde{W}_{j,x}(\mathbf{x}) + \alpha_{12}\widetilde{W}_{j,y}(\mathbf{x}) \tag{4.101}$$

$$G_{jy} = \alpha_{21}\widetilde{W}_{j,x}(\mathbf{x}) + \alpha_{22}\widetilde{W}_{j,y}(\mathbf{x}) \tag{4.102}$$

Because the Shepard functions satisfy zero-th order completeness, the corrected gradients G_{ij} automatically satisfy zero-th order completeness. The coefficients α_{ij} are computed by imposing the reproducing conditions on the derivatives of linear functions. Let $f_{S,i}(\mathbf{x}) = x_i$, then the reproducing condition for the derivatives requires

$$\begin{aligned}
\nabla f_{S,i}(\mathbf{x}) &= \sum_{j=1}^{N} f_j G_{ij}(\mathbf{x}) && (4.103)\\
&= \sum_{j=1}^{N} x_{ij} G_{ij}(\mathbf{x}) && (4.104)\\
&= \sum_{j=1}^{N} \alpha_{ik} x_{ij} \widetilde{W}_{j,k}(\mathbf{x}) && (4.105)\\
&= \delta_{ij} && (4.106)
\end{aligned}$$

The above equation gives d sets of d equations in the unknowns α_{ij}, where d is the number of spatial dimensions. In two dimensions, the linear algebraic equations for the corrected derivatives are given as

$$\mathbf{A}\boldsymbol{\alpha} = \mathbf{I} \tag{4.107}$$

where $\mathbf{I}$ is the identity matrix and matrix $\mathbf{A}$ is given as

$$\mathbf{A} = \sum_{j=1}^{N} \begin{bmatrix} \widetilde{W}_{j,x}(\mathbf{x}) & \widetilde{W}_{j,y}(\mathbf{x}) \\ \widetilde{W}_{j,x}(\mathbf{x}) & \widetilde{W}_{j,y}(\mathbf{x}) \end{bmatrix} \tag{4.108}$$

This correction produces derivatives of constant and linear functions exactly. So, the gradient approximation satisfies first-order completeness. For computational reasons, the origin should be shifted to $\mathbf{x}$ for better conditioning of matrix $\mathbf{A}$ and the reduction of round-off error.

Derivatives can also be corrected with the form

$$\begin{aligned} \Phi_j &= (\alpha_{11}(\mathbf{x}) + \alpha_{12}(\mathbf{x})x_j + \alpha_{13}(\mathbf{x})y_j)\widetilde{W}_j(\mathbf{x}) && (4.109) \\ G_{jx} &= (\alpha_{21}(\mathbf{x}) + \alpha_{22}(\mathbf{x})x_j + \alpha_{23}(\mathbf{x})y_j)\widetilde{W}_j(\mathbf{x}) && (4.110) \\ G_{jy} &= (\alpha_{31}(\mathbf{x}) + \alpha_{32}(\mathbf{x})x_j + \alpha_{33}(\mathbf{x})y_j)\widetilde{W}_j(\mathbf{x}) && (4.111) \end{aligned}$$

The coefficients $\boldsymbol{\alpha}$ are obtained by equation (4.107) where $\mathbf{A}$ as

$$\mathbf{A} = \sum_{j=1}^{N} \widetilde{W}_j(\mathbf{x}) \begin{bmatrix} 1 & x_j - x & y_j - y \\ x_j - x & (x_j - x)^2 & (x_j - x)(y_j - y) \\ y_j - y & (x_j - x)(y_j - y) & (y_j - y)^2 \end{bmatrix} \tag{4.112}$$

In this case we need to invert a 3×3 matrix. But it has the advantage that linearly complete shape functions together with corrected derivatives are obtained.

Another approach to construct linearly consistent complete kernel function derivatives is given as the form

$$\begin{aligned} \Phi_j(\mathbf{x}) &= \alpha_{11}(\mathbf{x})\widetilde{W}_{j,x}(\mathbf{x}) + \alpha_{12}(\mathbf{x})\widetilde{W}_{j,y}(\mathbf{x}) + \alpha_{13}\widetilde{W}_j(\mathbf{x}) && (4.113) \\ G_{jx}(\mathbf{x}) &= \alpha_{21}(\mathbf{x})\widetilde{W}_{j,x}(\mathbf{x}) + \alpha_{22}(\mathbf{x})\widetilde{W}_{j,y}(\mathbf{x}) + \alpha_{23}\widetilde{W}_j(\mathbf{x}) && (4.114) \\ G_{jy}(\mathbf{x}) &= \alpha_{31}(\mathbf{x})\widetilde{W}_{j,x}(\mathbf{x}) + \alpha_{23}(\mathbf{x})\widetilde{W}_{j,y}(\mathbf{x}) + \alpha_{33}\widetilde{W}_j(\mathbf{x}) && (4.115) \end{aligned}$$

where $\boldsymbol{\alpha}(\mathbf{x})$ are obtained by requiring the approximation Φ_j to reproduce linear functions and the derivatives G_{jx} and G_{jy} approximations to reproduce derivatives of linear functions. Similarly, we obtain the coefficients $\boldsymbol{\alpha}$ using equation (4.107) where $\mathbf{A}$ is now given as

$$\mathbf{A} = \sum_{j=1}^{N} \begin{bmatrix} \widetilde{W}_{j,x}(\mathbf{x}) & \widetilde{W}_{j,y}(\mathbf{x}) & \widetilde{W}_j(\mathbf{x}) \\ \widetilde{W}_{j,x}(\mathbf{x})x_j & \widetilde{W}_{j,y}(\mathbf{x})x_j & \widetilde{W}_j(\mathbf{x})x_j \\ \widetilde{W}_{j,x}(\mathbf{x})y_j & \widetilde{W}_{j,y}(\mathbf{x})y_j & \widetilde{W}_j(\mathbf{x})y_j \end{bmatrix} \tag{4.116}$$

This approach involves a higher computational cost with also inverting a 3×3 matrix. The kernel functions themselves can reproduce linear functions rather than just constant functions as was the case in the earlier constructions.

5. Semi-implicit SPH Method for Shallow Water Flows

After the theoretical properties of flow phenomena, the derivation of the shallow water equations and the SPH method in the preceding chapters, we now consider the proposed semi-implicit Smoothed Particle Hydrodynamics (SISPH) method designed for the shallow water equations in the present chapter. We discuss the design, development and analysis of the SISPH method following the work of Casulli [25], we will mention its significant properties.

5.1 Introduction

In this section, we will discuss possible numerical methods that can be employed to solve the shallow water equations. For clarity into the subject, the numerical treatment of free surface flows will be presented by using simple advection-diffusion equation

$$\frac{\partial C}{\partial t} + u\frac{\partial C}{\partial x} + v\frac{\partial C}{\partial y} + w\frac{\partial C}{\partial z} = D\left[\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2}\right] - \gamma C \tag{5.1}$$

where $C(x, y, z, t)$ denote the unknown function; u, v, w are advective coefficients; D is the nonnegative diffusion coefficient; and γ is the nonnegative dissipative coefficient. We assume all these coefficients to be constants for simplicity.

Definition 5.1.1 (**CFL Stability Condition**). *The CFL stability condition expresses that the mesh ratio has to be chosen in such a way that the domain of dependence of the differential equation should be contained in the domain of dependence of the discretized equations.*

Theorem 5.1.2 (**CFL Theorem**). *The CFL condition is a necessary condition for the convergence of a numerical approximation of a partial differential equation, linear or nonlinear.*

Definition 5.1.3 (**Maximum-minimum property**). *The maximum-minimum property expresses that for any non-constant analytical solution of the advection-diffusion equation (5.1) it can only assume a positive maximum and a negative minimum at the initial time or at the boundary point.*

The solution of equation (5.1) possesses the maximum-minimum property. The maximum-minimum property is very important because mostly in the description of physical phenomena the unknown function C may represent a physical quantity which would lose its significance if assumed negative values. If the prescribed initial and boundary conditions are nonnegative, then the solution of equation (5.1) is everywhere nonnegative. This property is important when dealing with the shallow water equations because we do not want water height to be negative.

Remark 5.1.4. *The maximum-minimum property is a sufficient condition for the numerical stability of the proposed numerical method.*

We will discretize (5.1) by finite difference numerical methods, these methods possess the discrete maximum-minimum property. The aim of doing this is to bring out some cogent points about stability which is important for us to understand the goals in this thesis.

Explicit Methods

Explicit Central Difference

The explicit central difference is one of the ways to numerically solve (5.1). Discretizing the time derivative by forward differences and central differences to approximate the advective and diffusive terms in (5.1) and let Δt be the time step, Δx, Δy and Δz are the spatial increments in the x, y, z directions respectively, the resulting explicit central difference scheme is given as

$$\begin{aligned} C_{i,j,k}^{n+1} = &\left[1-\Delta t\left(\frac{2D}{\Delta x^2}+\frac{2D}{\Delta y^2}+\frac{2D}{\Delta z^2}+\gamma\right)\right]C_{i,j,k}^{n} \\ &+\Delta t\left(\frac{D}{\Delta x^2}-\frac{u}{2\Delta x}\right)C_{i+1,j,k}^{n}+\Delta t\left(\frac{D}{\Delta x^2}+\frac{u}{2\Delta x}\right)C_{i-1,j,k}^{n} \\ &+\Delta t\left(\frac{D}{\Delta y^2}-\frac{v}{2\Delta y}\right)C_{i,j+1,k}^{n}+\Delta t\left(\frac{D}{\Delta y^2}+\frac{v}{2\Delta y}\right)C_{i,j-1,k}^{n} \\ &+\Delta t\left(\frac{D}{\Delta z^2}-\frac{w}{2\Delta z}\right)C_{i,j,k+1}^{n}+\Delta t\left(\frac{D}{\Delta z^2}+\frac{w}{2\Delta z}\right)C_{i,j,k-1}^{n} \end{aligned} \tag{5.2}$$

The stability of the scheme can be analyzed as follows; It is easy to check that if the time step Δt satisfies the inequality

$$\Delta t \leq \frac{1}{\frac{2D}{\Delta x^2}+\frac{2D}{\Delta y^2}+\frac{2D}{\Delta z^2}+\gamma} \tag{5.3}$$

then the coefficient of $C_{i,j,k}^{n}$ on the right-hand side of equation (5.2) is nonnegative. But this inequality is insufficient to assure stability and maximum principle of the numerical solution. To to this effect, the coefficients of $C_{i\pm1,j,k}^{n}$, $C_{i,j\pm1,k}^{n}$ and $C_{i,j,k\pm1}^{n}$ in equation (5.2) must be nonnegative too. Eventually, this requirement leads to the following restrictions on the space

$$|u|\Delta x \leq 2D \tag{5.4}$$

$$|v|\Delta y \leq 2D \tag{5.5}$$

$$|w|\Delta z \leq 2D \tag{5.6}$$

When the inequalities including (5.3) are satisfied, the right-hand side of equation (5.2) can be regarded as a weighted average between zero and the values of C at time level t^n. Hence, $C_{i,j,k}^{n+1}$ is bounded above and below by the maximum and by the minimum, respectively, of zero and C^m. However, the inequalities are too restrictive in advection dominated diffusion problems and this cannot apply in the absence of diffusion $(D = 0)$.

Explicit Upwinding

The upwind method is a method that does not require any limitation on the mesh sizes. By taking a forward finite difference to approximate the time derivative, using upwind finite differences to approximate the advective terms and central differences to approximate the diffusive terms in equation (5.1) we obtain the explicit upwind scheme

$$\begin{aligned} C_{i,j,k}^{n+1} = & \left[1 - \Delta t \left(\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} + \frac{2D}{\Delta x^2} + \frac{2D}{\Delta y^2} + \frac{2D}{\Delta z^2} + \gamma \right) \right] C_{i,j,k}^{n} \\ & + \Delta t \left(\frac{D}{\Delta x^2} + \frac{|u| - u}{2\Delta x} \right) C_{i+1,j,k}^{n} + \Delta t \left(\frac{D}{\Delta x^2} + \frac{|u| + u}{2\Delta x} \right) C_{i-1,j,k}^{n} \\ & + \Delta t \left(\frac{D}{\Delta y^2} + \frac{|v| - v}{2\Delta y} \right) C_{i,j+1,k}^{n} + \Delta t \left(\frac{D}{\Delta y^2} + \frac{|v| + v}{2\Delta y} \right) C_{i,j-1,k}^{n} \\ & + \Delta t \left(\frac{D}{\Delta z^2} + \frac{|w| - w}{2\Delta z} \right) C_{i,j,k+1}^{n} + \Delta t \left(\frac{D}{\Delta z^2} + \frac{|w| + w}{2\Delta z} \right) C_{i,j,k-1}^{n} \end{aligned} \tag{5.7}$$

It is easy to check that if the time step Δt satisfies the inequality

$$\Delta t \leq \frac{1}{\frac{|u|}{\Delta x} + \frac{|v|}{\Delta y} + \frac{|w|}{\Delta z} + \frac{2D}{\Delta x^2} + \frac{2D}{\Delta y^2} + \frac{2D}{\Delta z^2} + \gamma} \tag{5.8}$$

then the right-hand side of equation (5.7) can be regarded as a weighted average between zero and values of C at t^n. The values for $C_{i,j,k}^{n+1}$ are bounded above and below by the maximum and minimum.

Implicit Methods

Implicit Central Difference

We can improve the stability of a numerical scheme by taking an implicit discretization of the spatial derivative contributions. Using the backward finite difference to approximate the time derivative and central differences to approximate the advective and diffusive terms, the implicit central difference scheme is given by

$$\begin{aligned} \left[1 + \Delta t \left(\frac{2D}{\Delta x^2} + \frac{2D}{\Delta y^2} + \frac{2D}{\Delta z^2} + \gamma \right) \right] C_{i,j,k}^{n+1} = C_{i,j,k}^{n} \\ + \Delta t \left(\frac{D}{\Delta x^2} - \frac{u}{2\Delta x} \right) C_{i+1,j,k}^{n+1} + \Delta t \left(\frac{D}{\Delta x^2} + \frac{u}{2\Delta x} \right) C_{i-1,j,k}^{n+1} \\ + \Delta t \left(\frac{D}{\Delta y^2} - \frac{v}{2\Delta y} \right) C_{i,j+1,k}^{n+1} + \Delta t \left(\frac{D}{\Delta y^2} + \frac{v}{2\Delta y} \right) C_{i,j-1,k}^{n+1} \\ + \Delta t \left(\frac{D}{\Delta z^2} - \frac{w}{2\Delta z} \right) C_{i,j,k+1}^{n+1} + \Delta t \left(\frac{D}{\Delta z^2} + \frac{w}{2\Delta z} \right) C_{i,j,k-1}^{n+1} \end{aligned} \tag{5.9}$$

Obviously, equation (5.9) contains seven unknowns, namely $C_{i,j,k}^{n+1}$, $C_{i\pm1,j,k}^{n+1}$, $C_{i,j\pm1,k}^{n+1}$ and $C_{i,j,k\pm1}^{n+1}$. So, with a properly prescribed boundary conditions, equation (5.9) constitute a seven-diagonal linear system with $N_x N_y N_z$ equations and $N_x N_y N_z$ unknowns. This system must be solved at every time step in order to determine the solution at the new time level t^{n+1}.

It is easy to check that the coefficient of $C_{i,j,k}^{n+1}$ on the left-hand side of equation (5.9) is always strictly positive for any time step size Δt and for any Δx, Δy and Δz. The discrete maximum principle requires that the coefficients of $C_{i\pm1,j,k}^{n+1}$, $C_{i,j\pm1,k}^{n+1}$ and $C_{i,j,k\pm1}^{n+1}$ on the right-hand side of equation (5.9) must be nonnegative. This requirement leads to the following restrictions

$$|u|\Delta x \leq 2D \tag{5.10}$$

$$|v|\Delta y \leq 2D \tag{5.11}$$

$$|w|\Delta z \leq 2D \tag{5.12}$$

When the inequalities are satisfied, $C_{i,j,k}^{n+1}$ can be expressed as a weighted average between zero and values of C. Hence, the discrete maximum principle and stability of the implicit scheme is obtained without any limitation on the time step size.

Remark 5.1.5. *The implicit central difference does not require any limitation on the time step size*

Implicit Upwinding

Similarly, the implicit upwind method can be written as

$$\begin{aligned}\left[1+\Delta t\left(\frac{|u|}{\Delta x}+\frac{|v|}{\Delta y}+\frac{|w|}{\Delta z}+\frac{2D}{\Delta x^2}+\frac{2D}{\Delta y^2}+\frac{2D}{\Delta z^2}+\gamma\right)\right]C_{i,j,k}^{n+1} &= C_{i,j,k}^{n}\\ +\Delta t\left(\frac{D}{\Delta x^2}+\frac{|u|-u}{2\Delta x}\right)C_{i+1,j,k}^{n+1}+\Delta t\left(\frac{D}{\Delta x^2}+\frac{|u|+u}{2\Delta x}\right)C_{i-1,j,k}^{n+1}&\\ +\Delta t\left(\frac{D}{\Delta y^2}+\frac{|v|-v}{2\Delta y}\right)C_{i,j+1,k}^{n+1}+\Delta t\left(\frac{D}{\Delta y^2}+\frac{|v|+v}{2\Delta y}\right)C_{i,j-1,k}^{n+1}&\\ +\Delta t\left(\frac{D}{\Delta z^2}+\frac{|w|-w}{2\Delta z}\right)C_{i,j,k+1}^{n+1}+\Delta t\left(\frac{D}{\Delta z^2}+\frac{|w|+w}{2\Delta z}\right)C_{i,j,k-1}^{n+1}& \qquad (5.13)\end{aligned}$$

We can see that for any time step Δt, the numerical approximates obtained by (5.13) admits the discrete maximum-minimum property. Hence, the scheme is unconditionally stable.

Eulerian-Lagrangian Methods

The Eulerian-Lagrangian methods performs much better in terms of both accuracy and stability of an explicit finite difference method. Since, the SPH method is a Lagrangian description, we will like to describe the advection-diffusion equation in the Lagrangian form. We write equation (5.1) in Lagrangian derivatives

$$\frac{DC}{Dt}=D\left[\frac{\partial^2 C}{\partial x^2}+\frac{\partial^2 C}{\partial y^2}+\frac{\partial^2 C}{\partial z^2}\right]-\gamma C \qquad (5.14)$$

where D/Dt denotes the substantial derivative which means the rate of change of time is considered along the streamline given by

$$\frac{Dx}{Dt}=u, \qquad \frac{Dy}{Dt}=v, \qquad \frac{Dz}{Dt}=w \qquad (5.15)$$

An explicit discretization of equation (5.14) reads

$$\begin{aligned}C_{i,j,k}^{n+1} &= \left[1-\Delta t\left(\frac{2D}{\Delta x^2}+\frac{2D}{\Delta y^2}+\frac{2D}{\Delta z^2}+\gamma\right)\right]C_{i-a,j-b,k-d}^{n}\\ &+\Delta t\frac{D}{\Delta x^2}C_{i-a+1,j-b,k-d}^{n}+\Delta t\frac{D}{\Delta x^2}C_{i-a-1,j-b,k-d}^{n}\\ &+\Delta t\frac{D}{\Delta y^2}C_{i-a,j-b+1,k-d}^{n}+\Delta t\frac{D}{\Delta y^2}C_{i-a,j-b-1,k-d}^{n}\\ &+\Delta t\frac{D}{\Delta z^2}C_{i-a,j-b,k-d+1}^{n}+\Delta t\frac{D}{\Delta z^2}C_{i-a,j-b,k-d-1}^{n} \qquad (5.16)\end{aligned}$$

where $a=u\frac{\Delta t}{\Delta x}$, $b=v\frac{\Delta t}{\Delta y}$ and $d=w\frac{\Delta t}{\Delta z}$ are the Courant numbers in the x, y and z directions respectively. In equation (5.16), the value of C at time level t^{n+1} at the node (i,j,k) is related to the value of C at time level t^n in $(i-a, j-b, k-d)$ which actually diffuses in time Δt. The equation accounts for both advection and diffusion. However, a, b and d are not integers and $(i-a, j-b, k-d)$ is not a grid point but an interpolation formula should be used to define $C^n_{i-a,j-b,k-d}$ in equation (5.16). From equation (5.16), if the time step satisfies

$$\Delta t \leq \frac{1}{\frac{2D}{\Delta x^2} + \frac{2D}{\Delta y^2} + \frac{2D}{\Delta z^2} + \gamma} \tag{5.17}$$

then the right-hand side of (5.16) is a weighted average between zero and C^n. Thus, $C^n_{i,j,k}$ is bounded above and below by the maximum and minimum respectively of zero and values of $C^n_{i-a,j-b,k-d}$. This stability restriction (5.17) can be eliminated if we use an implicit Eulerian-Lagrangian method which is given as

$$\begin{aligned}
&\frac{C^{n+1}_{i,j,k} - C^n_{i-a,j-b,k-d}}{\Delta t} \\
&\qquad = D\frac{C^{n+1}_{i+1,j,k} - 2C^{n+1}_{i,j,k} + C^{n+1}_{i-1,j,k}}{\Delta x^2} + D\frac{C^{n+1}_{i,j+1,k} - 2C^{n+1}_{i,j,k} + C^{n+1}_{i,j-1,k}}{\Delta y^2} \\
&\qquad\quad + D\frac{C^{n+1}_{i,j,k+1} - 2C^{n+1}_{i,j,k} + C^{n+1}_{i,j,k-1}}{\Delta z^2} - \gamma C^{n+1}_{i,j,k}
\end{aligned} \tag{5.18}$$

The equation constitute a linear seven-diagonal system of $N_xN_yN_z$ equations with $N_xN_yN_z$ unknowns and the system is symmetric and positive definite for any Δt.

Semi-implicit Methods

As discussed in the earlier sections for explicit and implicit methods. The difference between both methods is that explicit methods are simple, but they are possibly plagued by severe stability condition. Whereas, unconditional stability can be achieved by implicit methods but with a high cost attributed to computational complexity. In order to derive numerical methods which are stable at a minimal computational cost, an *implicitness factor* Θ is introduced. The semi-implicit approximation for (5.14) is

$$\begin{aligned}
&\frac{C^{n+1}_{i,j,k} - C^n_{i-a,j-b,k-d}}{\Delta t} \\
&\qquad = \Theta_1 D\left[\frac{C^{n+1}_{i+1,j,k} - 2C^{n+1}_{i,j,k} + C^{n+1}_{i-1,j,k}}{\Delta x^2} + \frac{C^{n+1}_{i,j+1,k} - 2C^{n+1}_{i,j,k} + C^{n+1}_{i,j-1,k}}{\Delta y^2}\right] \\
&\qquad\quad + \Theta_2 D\frac{C^{n+1}_{i,j,k+1} - 2C^{n+1}_{i,j,k} + C^{n+1}_{i,j,k-1}}{\Delta z^2} - \Theta_3\gamma C^{n+1}_{i,j,k} \\
&\qquad\quad + (1-\Theta_1)D\frac{C^n_{i-a+1,j-b,k-d} - 2C^n_{i-a,j-b,k-d} + C^n_{i-a-1,j-b,k-d}}{\Delta x^2} \\
&\qquad\quad + D\frac{C^n_{i-a,j-b+1,k-d} - 2C^n_{i-a,j-b,k-d} + C^n_{i-a,j-b-1,k-d}}{\Delta y^2} \\
&\qquad\quad + (1-\Theta_2)D\frac{C^n_{i-a,j-b,k-d+1} - 2C^n_{i-a,j-b,k-d} + C^n_{i-a,j-b,k-d-1}}{\Delta z^2} \\
&\qquad\quad - (1-\Theta_3)\gamma C^n_{i-a,j-b,k-d}
\end{aligned} \tag{5.19}$$

where Θ_1, Θ_2 and Θ_3 are the implicitness factors for the horizontal diffusion, vertical diffusion, and for the sink term respectively.

Remark 5.1.6. *In equation (5.19), if $\Theta_1 = \Theta_2 = \Theta_3 = 0$, then the scheme reduces to the explicit scheme. If $\Theta_1 = \Theta_2 = \Theta_3 = 1$, then the scheme becomes an implicit scheme*

The resulting scheme is a semi-implicit method which also possesses the maximum-minimum property if the time step Δt satisfies the inequality

$$\Delta t \leq \frac{1}{(1-\Theta_1)\left(\frac{2D}{\Delta x^2}+\frac{2D}{\Delta y^2}\right)+(1-\Theta_2)\frac{2D}{\Delta z^2}+(1-\Theta_3)\gamma} \tag{5.20}$$

With the preceding discussions made above, we can proceed into the derivation of the proposed scheme in this thesis.

5.2 Derivation of the SISPH Method

Now we consider our meshfree method. In this section, we look into the core aim in this thesis towards the derivation of the semi-implicit SPH method for the shallow water equations. There are several numerical methods that can be employed to solve the shallow water equations. These methods can be finite differences or finite elements, explicit or implicit, conservative or non-conservative or meshless methods. In this section, following the semi-implicit finite volume and finite difference approach of Casulli [25], we will delve into the derivation of the semi-implicit SPH scheme applied to the two dimensional shallow water equations.

In recent years, some authors have worked on a semi-implicit method for meshfree particle methods. In the specific, Koshizuka and Oka [66, 67] presented the moving-particle semi-implicit (MPS) method where a deterministic interaction models for the gradient, Laplacian operators and free surfaces are presented. Incompressibility condition is imposed by setting rate of change of density with time to zero at each time step, likewise a modified kernel function which has a unique property that the value of the kernel goes to infinity as distance between particles tends to zero; a kernel function which has been validated to avoid particle clumping. Ataie-Ashtiani and Farhadi [3, 4] worked in the same direction and presented a stable MPS method for free surface flows using a fractional step idea of discretization to split the time step into two steps. A number of authors modified, extended and improved on the MPS method of Koshika and Oka (see [62, 63, 64, 65, 119]) even more for the enhancement of performance, stability and accuracy of the MPS method.

A considerable amount of work has been done for both structured and unstructured meshes using finite difference, finite volume and finite element schemes [28, 29, 30, 42, 105]. As we justified in the introductory section, a major problem of explicit schemes in numerical methods is their severe time step restriction, where the Courant-Friedrichs-Lewy (CFL) condition imposes the time step size in terms of the wave propagation speed and the mesh size. Hence, the major advantage of a semi-implicit approach is that stable schemes are obtained which allow large time step sizes at a reasonable computational cost.

In a staggered mesh approach for finite differences and volumes, discrete variables are defined at different (staggered) locations. The pressure term, which is the free surface elevation is defined in the cell center while the velocity components are defined at the cell interfaces. In the momentum equations, pressure terms are due to the gradients in the free surface elevations and the velocities in the mass equation (i.e., free surface equation) are both discretized implicitly whereas the nonlinear convective terms are discretized explicitly. The semi-Lagrangian method is one of the techniques to discretize these terms explicitly, see [18, 56, 73].

In standard explicit numerical methods, there is a severe limitation due to the stability restriction imposed by the CFL condition. The restriction requires a much smaller time step size than permitted by accuracy considerations. Fully implicit discretization often leads to unconditionally stable methods but they typically lead to the simultaneous solution of a large number of coupled nonlinear equations. For accuracy, the time step cannot be chosen arbitrarily large. To this effect, a stable, efficient, robust and simple semi-implicit SPH numerical method is derived.

In this thesis the new semi-implicit *Smoothed Particle Hydrodynamics* (SPH) scheme presented by Bankole et al. [7, 8, 10] for the numerical solution of the shallow water equations on 2D particle configuration will be proposed, derived and discussed. The flow variables in the present study are the particle free surface elevation, particle total water depth and the particle velocity. The discrete momentum equations are substituted into the discretized mass conservation equation to give a discrete equation for the free surface leading to a system in only one single scalar quantity, the free surface elevation location: a unique feature that makes our method stands out from other semi-implicit approaches. The system is solved for each time step as a linear algebraic system. The components of the momentum equation at the new time level can be directly computed from the new free surface. This can be conveniently solved by a matrix-free version of the conjugate gradient (CG) algorithm [50, 99]. Consequently, the particle velocities at the new time level are computed and the particle positions are updated. In this semi-implicit SPH method, the stability is independent of the wave celerity. Hence, a relatively large time steps can be permitted to enhance the numerical efficiency [28]. The application of a staggered velocity between particles is one of the novelty of the present study. The staggered velocity improves the sparsity of the resulting linear system significantly. Moreover, an integral part of the resulting numerical scheme, the discrete free-surface equation has been treated to represent the accurate mass balance. The resulting system leads to a *nonlinear system* while a mass conservation and nonnegative water depths are guaranteed everywhere in the flow domain and for all time steps. A few number of iterations are needed to solve the resulting *nonlinear system*.

Governing Equations

The governing equations considered in this thesis are nonlinear hyperbolic conservation laws of the generic form

$$L_b(\mathbf{\Phi}) + \nabla \cdot (\boldsymbol{F}(\mathbf{\Phi}, \boldsymbol{x}, t)) = 0 \qquad \text{for } t \in \mathbb{R}^+, \mathbf{\Phi} \in \mathbb{R} \tag{5.21}$$

together with the initial condition

$$\mathbf{\Phi}(\boldsymbol{x}, 0) = \mathbf{\Phi}_0(\boldsymbol{x}) \qquad \text{for } \boldsymbol{x} \in \Omega \subset \mathbb{R}^d, \mathbf{\Phi}_0 \in \mathbb{R}$$

where L_b is the transport operator given by

$$L_b(\mathbf{\Phi}) = \frac{\partial \mathbf{\Phi}}{\partial t} + \nabla \cdot (b\mathbf{\Phi})$$

and

$$\boldsymbol{x} = (x^1, ..., x^d), \quad \boldsymbol{F} = (F^1, ..., F^d), \quad \boldsymbol{b} = (b^1, ..., b^d),$$

where $\boldsymbol{b}$ is a regular vector field in $\mathbb{R}^d$, $\boldsymbol{F}$ is a flux vector in $\mathbb{R}^d$, and $\boldsymbol{x}$ is the position.

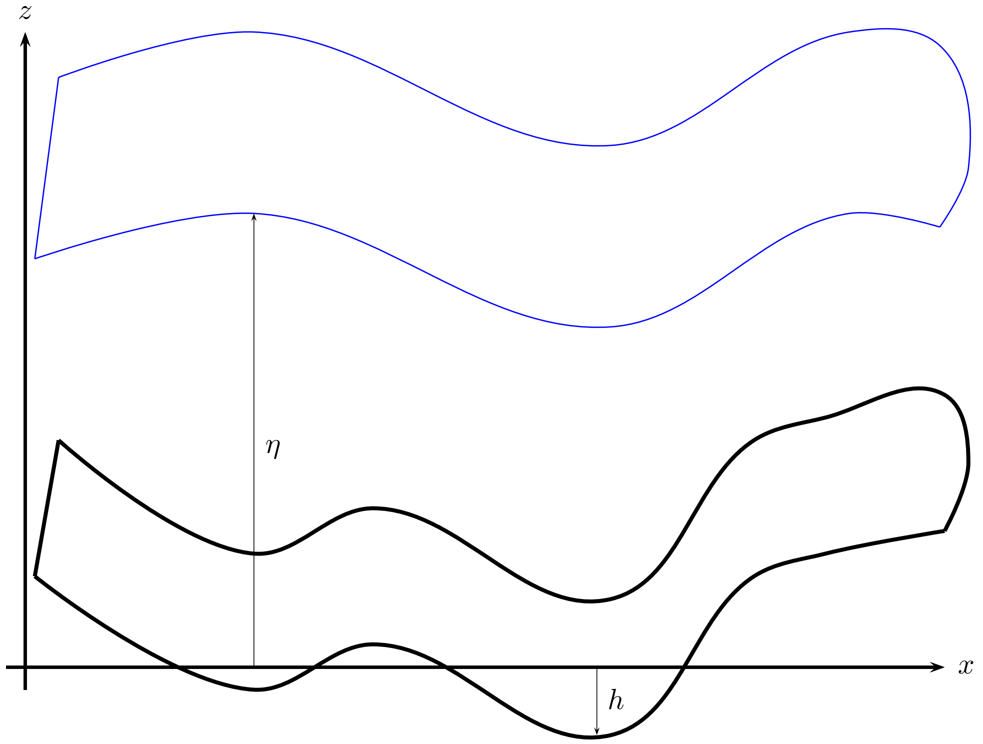


Figure 5.1: Sketch of the flow domain: the free surface (light) and the bottom bathymetry (thick).

Fig. 5.1 gives a sketch of the flow domain, i.e., the free surface elevation and the bottom bathymetry. In this configuration, the vertical variation is much smaller than the horizontal variation, as typical for rivers flowing over long distances of e.g. hundreds or thousands of kilometers. We consider the frictionless, inviscid two dimensional shallow water equations in Lagrangian derivatives, given as

$$\frac{D\eta}{Dt} + \nabla \cdot (H\boldsymbol{v}) = 0 \tag{5.22}$$

$$\frac{D\boldsymbol{v}}{Dt} + g\nabla\eta = 0 \tag{5.23}$$

$$\frac{D\boldsymbol{r}}{Dt} = \boldsymbol{v} \tag{5.24}$$

where $\eta = \eta(x, y, t)$ is the free surface location,

$$H(x, y, t) = h(x, y) + \eta(x, y, t)$$

is the total water depth with bottom bathymetry $h(x, y)$, and where $\boldsymbol{v} = v(x, y, t)$ is the particle velocity, $\boldsymbol{r} = r(x, y, t)$ the particle position, and g the gravitational acceleration.

Classical SPH Formulation

The standard SPH formulation explained in Chapter 4, discretizes the computational domain $\Omega(t)$ by a finite set of N particles, with particle positions $\boldsymbol{r}_i$. According to Gingold and Monaghan [48], the SPH discretization of the shallow water equations (5.22)-(5.24) by an explicit time discretization are given as

$$\frac{\eta_i^{n+1}-\eta_i^n}{\Delta t}+\sum_{j=1}^{N}\frac{m_j}{\rho_j}H_{ij}^n\boldsymbol{v}_j^n\nabla W_{ij}=\mathbf{0} \tag{5.25}$$

$$\frac{\boldsymbol{v}_i^{n+1}-\boldsymbol{v}_i^n}{\Delta t}+g\sum_{j=1}^{N}\frac{m_j}{\rho_j}\eta_j^n\nabla W_{ij}=\mathbf{0} \tag{5.26}$$

$$\frac{\boldsymbol{r}_i^{n+1}-\boldsymbol{r}_i^n}{\Delta t}=\boldsymbol{v}_i^n \tag{5.27}$$

where the particles are advected by (5.27), with Δt being the time step size, m_j the particle mass, ρ_j the particle density, and ∇W_{ij} is the gradient of kernel W_{ij} w.r.t. x_i. In the scheme [48, 90] of Gingold and Monaghan, $\nabla\cdot(H\boldsymbol{v})$ and $\nabla\eta$ are explicitly computed. We remark that eqns. (5.25)-(5.27) follow from a substitution of the flow variable with corresponding derivatives, using integration by parts, and the divergence theorem.

SPH formulation of Vila and Ben Moussa

In the construction of our proposed semi-implicit SPH scheme, we use the concept of Vila & Ben Moussa [93, 115], whose basic idea is to replace the centered approximation

$$(F(v_i,x_i,t)+F(v_j,x_j,t))\cdot n_{ij}$$

of (5.21) by a numerical flux $G(n_{ij},v_i,v_j)$, from a finite difference scheme in conservation form, $2G(n,u,v)$, which is required to satisfy

$$G(n(x),v,v)=F(v,x,t)\cdot n(x),$$

$$G(n,v,u)=-G(-n,u,v).$$

where the numerical viscosity $Q(n,u,v)$ and the incremental coefficient $C(n,u,v)$ are defined classically in the scalar case as

$$\begin{aligned} Q(n,u,v) &= \frac{F(u,x,t).n-2G(n,u,v)+F(v,x,t).n}{v-u}, \\ C(n,u,v) &= \frac{F(u,x,t).n-G(n,u,v)}{v-u}. \end{aligned}$$

With using this formalism, the SPH discretization of equations (5.22)-(5.24) become

$$\begin{aligned} \frac{\eta_i^{n+1}-\eta_i^n}{\Delta t}+\sum_{j=1}^{N}\frac{m_j}{\rho_j}2H_{ij}^n\boldsymbol{v}_{ij}^n\nabla W_{ij} &= \mathbf{0}, \\ \frac{\boldsymbol{v}_i^{n+1}-\boldsymbol{v}_i^n}{\Delta t}+g\sum_{j=1}^{N}\frac{m_j}{\rho_j}2\eta_{ij}^n\nabla W_{ij} &= \mathbf{0}, \\ \frac{\boldsymbol{r}_i^{n+1}-\boldsymbol{r}_i^n}{\Delta t} &= \boldsymbol{v}_i^n \end{aligned}$$

In this way, we define for a pair of particles, i and j, the free surface elevation η_i, η_j and the velocity $\boldsymbol{v}_i$, $\boldsymbol{v}_j$, respectively (see Fig. 5.2). In our approach, we, moreover, use a staggered velocity $\boldsymbol{v}_{ij}$ between two interacting particles i and j as

$$\boldsymbol{v}_{ij} = \frac{1}{2}(\boldsymbol{v}_i + \boldsymbol{v}_j) \cdot \boldsymbol{n}_{ij}$$

in the normal direction $\boldsymbol{n}_{ij}^{d=1,2}$ at the midpoint of the two interacting particles, where

$$n_{ij}^1 = \frac{x_j - x_i}{\|x_j - x_i\|} \qquad \text{and} \qquad n_{ij}^2 = \frac{y_j - y_i}{\|y_j - y_i\|}$$

for the two components of vector $\boldsymbol{n}_{ij}$. Moreover,

$$\delta_{ij}^1 = \|x_j - x_i\| \qquad \text{and} \qquad \delta_{ij}^2 = \|y_j - y_i\|$$

gives the distance between particles i and j. Since the velocities at the particles' midpoint are known, we can use kernel summation for velocity updates, i.e., between the midpoint velocities and the natural velocities.

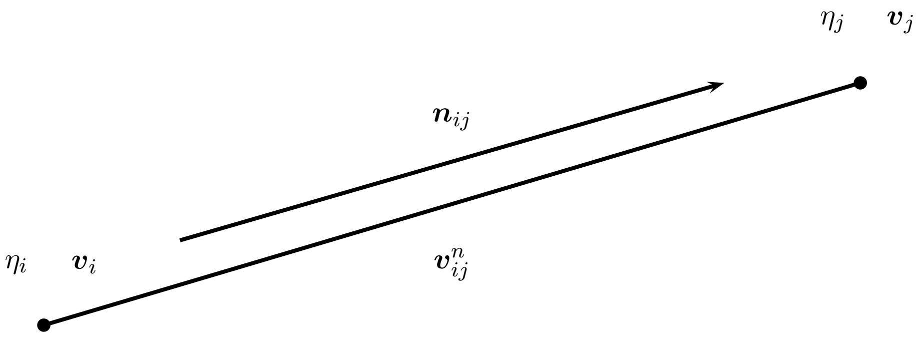


Figure 5.2: Staggered velocity defined at the midpoint of two pair of interacting particles i and j.

The Semi-implicit SPH Scheme

For the derivation of the semi-implicit SPH scheme, let us consider the governing equations (5.22)-(5.23). Writing equations (5.22)-(5.23) in a non-conservative quasi-linear form by expanding derivatives in the continuity equation and momentum equations (with assuming smooth solutions), this yields

$$u_t + uu_x + vu_y + g\eta_x = 0 \tag{5.28}$$

$$v_t + uv_x + vv_y + g\eta_y = 0 \tag{5.29}$$

$$\eta_t + u\eta_x + v\eta_y + H(u_x + v_y) = -uh_x - vh_y. \tag{5.30}$$

Rewriting (5.28)-(5.30) in matrix form, we obtain

$$\mathbf{Q}_t + \mathbf{A}\mathbf{Q}_x + \mathbf{B}\mathbf{Q}_y = \mathbf{C}, \tag{5.31}$$

where

$$\mathbf{A} = \begin{pmatrix} u & 0 & \boxed{g} \\ 0 & u & 0 \\ \boxed{H} & 0 & u \end{pmatrix} \qquad \mathbf{B} = \begin{pmatrix} v & 0 & 0 \\ 0 & v & \boxed{g} \\ 0 & \boxed{H} & v \end{pmatrix}$$

$$\mathbf{Q} = \begin{pmatrix} u \\ v \\ \eta \end{pmatrix} \qquad \mathbf{C} = \begin{pmatrix} 0 \\ 0 \\ -uh_x - vh_y \end{pmatrix}.$$

Equation (5.31) is a strictly hyperbolic system with real and distinct eigenvalues. The characteristic equation, given by

$$det(q\mathbf{I} + r\mathbf{A} + s\mathbf{B}) = 0\,, \tag{5.32}$$

can be simplified as

$$(q + ru + sv)\left[(q + ru + sv)^2 - gH(r^2 + s^2)\right] = 0\,, \tag{5.33}$$

where the solution (r, s, q) of equation (5.33) are the directions normal to a characteristic cone at the cone's vertex. When solving the two-dimensional shallow water equations, a very important feature arises in the so called characteristic cone, see Fig. 5.3. Characteristics ending at point (x_j, t^{n+1}), at the apex of the cone are not finite in number anymore but they belong to a cone around the speed of advection (u, v). Hence, propagation of the solution cannot be treated by following all characteristics - this also motivates why the semi-implicit technique remains a very good choice. We split equation (5.33), whereby we obtain

$$q + ru + sv = 0$$

and

$$(q + ru + sv)^2 - gH(r^2 + s^2) = 0, \tag{5.34}$$

with the characteristic curves $u = dx/dt$ and $v = dy/dt$. If the characteristic cone has a vertex at $(\overline{x}, \overline{y}, \overline{t})$, then this cone consist of the line passing through vertex $(\overline{x}, \overline{y}, \overline{t})$ and parallel to the vector $(u, v, 1)$, see Fig. 5.3, satisfying

$$((x - \overline{x}) - u(t - \overline{t}))^2 + ((y - \overline{y}) - v(t - \overline{t}))^2 - gH(t - \overline{t})^2 = 0. \tag{5.35}$$

In particular, the gradient of the left hand side of (5.35) satisfies (5.34) on the cone surface. After solving (5.32), the solution yields

$$\lambda_1 = \boldsymbol{v} - \sqrt{gH}, \qquad \lambda_2 = \boldsymbol{v}, \qquad \lambda_3 = \boldsymbol{v} + \sqrt{gH}.$$

When the particle velocity $\boldsymbol{v}$ is far smaller than the particle celerity $\sqrt{gH}$, i.e., $|\boldsymbol{v}| \ll \sqrt{gH}$, the particle flow is said to be strictly subcritical and thus the characteristic speeds λ_1 and λ_3 have opposite directions. The maximum wave speed is given as

$$\lambda_{\max} = \max(\sqrt{gH_i}, \sqrt{gH_j}).$$

In this case, $\sqrt{gH}$ represents the dominant term which originates from the off diagonal terms g and H in the matrices **A** and **B**.

We now have tracked back where the term $\sqrt{gH}$ originates from in the governing equations. We remark that the first part of the characteristic cone in (5.33) depends only on the particle velocity u and v. Equation (5.34), defining the second part of the characteristic cone, depends only on the celerity $\sqrt{gH}$. As we can see, gH in (5.33) comes from the off-diagonal terms g and H in the matrices **A** and **B**. The

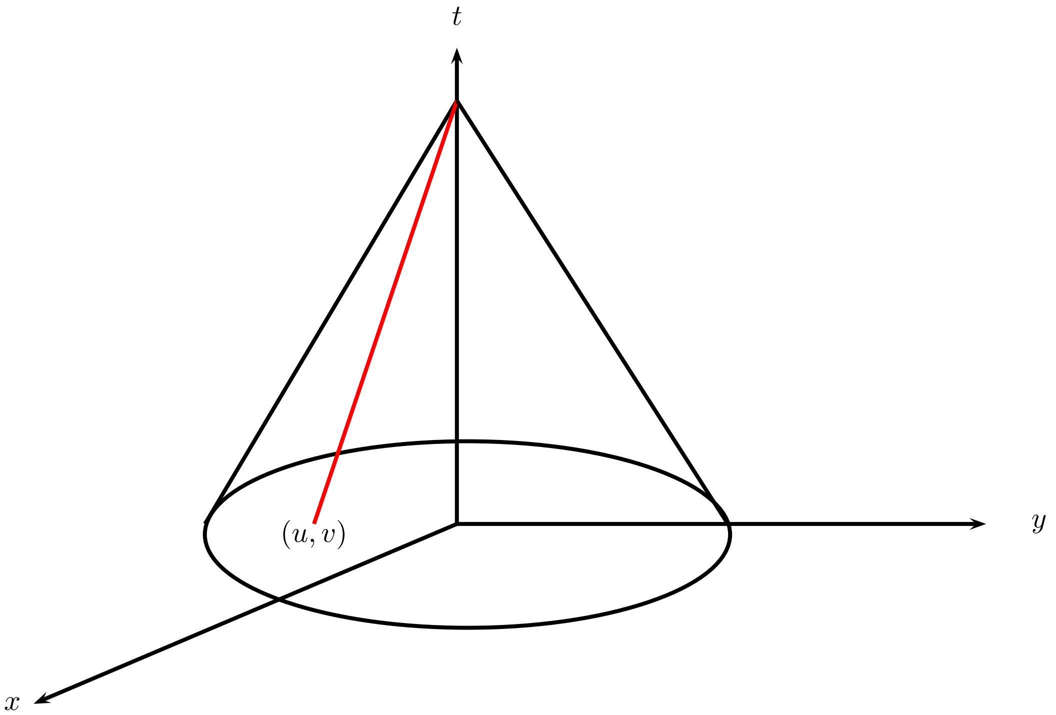


Figure 5.3: Characteristic cone for the two-dimensional shallow water equations.

terms g and H represent the coefficients of the derivative of the free surface elevation η_x in (5.28), the coefficient of the derivative η_y in (5.29) for the momentum equations, and the coefficient of velocity u_x and v_y in the volume conservation equation (5.30). We want to avoid the stability to depend on the celerity $\sqrt{gH}$, therefore we discretize the derivatives η_x, η_y and u_x, v_y implicitly.

Further along the lines of the above analysis, we now develop a semi-implicit SPH scheme for the two-dimensional shallow water equations. To this end, the derivatives of the free surface elevation η_x and η_y in the momentum equation and the derivative of the velocity in the continuity equation are discretized *implicitly*. The remaining terms, such as the nonlinear advective terms in the momentum equation, are discretized *explicitly*, so that the resulting equation system is linear.

The derivative of the velocity **v** and the free surface elevation η are discretized implicitly, whereas the total water depth H is discretized explicitly. In our following notation, for implicit and explicit discretization, we use $n+1$ and n for the superscript, respectively, i.e.,

$$\begin{aligned} \boldsymbol{v}_t^n + g \cdot \nabla \eta^{n+1} &= 0 \\ \eta_t^n + \nabla \cdot (H^n \boldsymbol{v}^{n+1}) &= 0. \end{aligned}$$

We discretize the particle velocities and free surface elevation in time by the Θ method, for the sake of time accuracy and computational efficiency, i.e., $n+1 = n+\Theta$, and so

$$\boldsymbol{v}_t^n + g \cdot \nabla \eta^{n+\Theta} = 0 \tag{5.36}$$

$$\eta_t^n + \nabla \cdot (H^n \boldsymbol{v}^{n+\Theta}) = 0 \tag{5.37}$$

where the Θ-method notation reads

$$\eta^{n+\Theta} = \Theta \eta^{n+1} + (1-\Theta)\eta^n$$

$$\boldsymbol{v}^{n+\Theta} = \Theta \boldsymbol{v}^{n+1} + (1-\Theta)\boldsymbol{v}^{n}.$$

The *implicitness factor* Θ should be in $[1/2, 1]$, according to Casulli & Cattani [28].

Theorem 5.2.1 (**Casulli and Cattani**)**.** *A semi-implicit finite difference scheme is stable in the von Neumann sense if* $\frac{1}{2} \leq \Theta \leq 1$ *and if the time time step* Δt *satisfies the following inequality*

$$\Delta t \leq \left[2D \left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right) \right]^{-1}. \tag{5.38}$$

The proof may be found in [28].

The general semi-implicit SPH discretization of (5.36)-(5.37) then takes the form

$$\frac{\boldsymbol{v}_{ij}^{n+1} - \boldsymbol{F}\boldsymbol{v}_{ij}^{n}}{\Delta t} + \frac{g}{\boldsymbol{\delta_{ij}}}\Theta(\eta_j^{n+1} - \eta_i^{n+1}) + \frac{g}{\boldsymbol{\delta}_{ij}}(1-\Theta)(\eta_j^{n} - \eta_i^{n}) = 0 \tag{5.39}$$

$$\begin{aligned} \frac{\eta_i^{n+1} - \eta_i^{n}}{\Delta t} &+ \Theta \sum_{j=1}^{N} \frac{m_j}{\rho_j} (2H_{ij}^{n} \boldsymbol{v}_{ij}^{n+1}) \nabla \boldsymbol{W}_{ij} \cdot \boldsymbol{n}_{ij} \\ &+ (1-\Theta) \sum_{j=1}^{N} \frac{m_j}{\rho_j} (2H_{ij}^{n} \boldsymbol{v}_{ij}^{n}) \nabla \boldsymbol{W}_{ij} \cdot \boldsymbol{n}_{ij} = 0 \end{aligned} \tag{5.40}$$

where

$$H_{ij}^{n} = \mathsf{max}(0, h_{ij}^{n} + \eta_i^{n}, h_{ij}^{n} + \eta_j^{n}).$$

In a Lagrangian formulation, the explicit operator $\boldsymbol{F}\boldsymbol{v}_{ij}^{n}$ in (5.39) has the form

$$\mathbf{F}\mathbf{v}_{ij}^{n} = \frac{1}{2}(\mathbf{v}_i^{n} + \mathbf{v}_j^{n}),$$

where $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ denote the velocity of particles i and j at time t^n. The velocity at time t^{n+1} is obtained by summation,

$$\mathbf{v}_i^{n+1} = \mathbf{v}_i^{n} + \sum_{j=1}^{N} \frac{m_j}{\rho_j} (\mathbf{v}_{ij}^{n+1} - \mathbf{v}_i^{n}) W_{ij}. \tag{5.41}$$

Note that in (5.39) we have *not* used the gradient of the kernel function for the discretization of the gradient of η. We rather used a finite difference discretization for the pressure gradient. This increases the accuracy, since $\boldsymbol{F}$ in (5.39) corresponds to an explicit spatial discretization of the advective terms. Since SPH is a Lagrangian scheme, the nonlinear convective term is discretized by the Lagrangian (material) derivative contained in the particle motion in (5.27). Equation (5.41) is used to interpolate the particle velocities from the particle location to the staggered velocity location. In equation (5.41), we have inherently exploited the advantages of the extended version of SPH suggested by Monaghan [87] where the particles are moved with a smoothed velocity. The smoothed velocity $\hat{\mathbf{v}}_i$ is defined by an average over the velocities of the neighboring particles according to

$$\hat{\mathbf{v}}_i = \mathbf{v}_i + \epsilon \sum_{j=1}^{N} \frac{m_j}{\rho_j} (\mathbf{v}_j - \mathbf{v}_i) W_{ij}, \tag{5.42}$$

where ϵ is a parameter that is typically chosen to be 0.5.

The Free Surface Equation

Let the particle volume ω_i in (5.40) be given as $\omega_i = m_i/\rho_i$. Irrespective of the form imposed on $\boldsymbol{F}$, equations (5.39)-(5.40) constitute a linear system of equations with unknowns $\boldsymbol{v}_i^{n+1}$ and η_i^{n+1} over the entire particle configuration. We solve this system at each time step for the particle variables from the prescribed initial and boundary conditions. To this end, the discrete momentum equation is substituted into the discrete continuity equation. This reduces the model to a smaller model, where η_i^{n+1} is the only unknown.

Multiplying (5.40) by ω_i and inserting (5.39) into (5.40), we obtain

$$\omega_i\eta_i^{n+1} - g\Theta^2\frac{\Delta t^2}{\boldsymbol{\delta_{ij}}}\sum_{j=1}^{N} 2\omega_i\omega_j \left[H_{ij}^n(\eta_j^{n+1} - \eta_i^{n+1})\nabla \boldsymbol{W}_{ij}\cdot \boldsymbol{n}_{ij}\right] = \boldsymbol{b}_i^n, \tag{5.43}$$

where the right hand side $\boldsymbol{b}_i^n$ represents the known values at time level t^n given as

$$\begin{aligned}\boldsymbol{b}_i^n &= \omega_i\eta_i^n - \Delta t\sum_{j=1}^{N} 2\omega_i\omega_j H_{ij}^n \boldsymbol{F}\boldsymbol{v}_{ij}^{n+\Theta}\nabla \boldsymbol{W}_{ij}\cdot \boldsymbol{n}_{ij}\\ &+ g\Theta(1-\Theta)\frac{\Delta t^2}{\boldsymbol{\delta_{ij}}}\sum_{j=1}^{N} 2\omega_i\omega_j\left[H_{ij}^n(\eta_j^n - \eta_i^n)\nabla \boldsymbol{W}_{ij}\cdot \boldsymbol{n}_{ij}\right],\end{aligned} \tag{5.44}$$

with $\boldsymbol{F}\boldsymbol{v}_{ij}^{n+\Theta} = \Theta\boldsymbol{F}\boldsymbol{v}_{ij}^n + (1-\Theta)\boldsymbol{v}_{ij}^n$. Since H_{ij}^n, ω_i, ω_j are non-negative numbers, equations (5.43)-(5.44) constitute a linear system of N equations for η_i^{n+1} unknowns.

The resulting system is symmetric and positive definite. Therefore, the system has a unique solution, which can be computed efficiently by an iterative method. We obtain the new free surface location by (5.40), and (5.39) yields the particle velocity $\boldsymbol{v}_i^{n+1}$.

5.3 Some Numerical Aspects

5.3.1 Boundary Conditions

In SPH, being a meshless particle method, the boundary of the computational domain is never well defined. The SPH method is plagued by the particle deficiency problem because the integral of the kernel function is truncated by the boundary as we can see earlier in Figure 4.4 in chapter 4, the SPH method needs a sufficient and necessary number of particles inside the support domain κh. In one, two and three dimensions the number of neighboring particles should be around 5, 21, and 57 respectively. Therefore, it is evident for particles near or on the boundary only particles inside the boundary contribute to the summation, so a one-sided contribution will lead to incorrect solutions. Over the years, several authors have proposed different solutions to the boundary treatment in SPH.

Ghost Particles

One of the first solution is the introduction of *ghost particles*. Libersky, Petscheck and Randles [79, 97] introduced ghost particles to reflect a symmetrical surface and Monaghan [89] introduced and used a line of virtual particles located right at the solid boundary to give a highly repulsive force to the particles near the boundary, this is done to avoid penetration - the Lennard-Jones potential. Particle position is mirrored from the flow to an external fictitious layer, particles nearby the wall are mirrored and a new ghost particle is created outside the domain that has the same properties of the SPH particle, Free-slip or no-slip conditions are easily realized with this technique. The *velocity slip condition*

$$\boldsymbol{v}_g \cdot \boldsymbol{n} = -\boldsymbol{v}_j \cdot \boldsymbol{n} \tag{5.45}$$

$$\boldsymbol{v}_g \cdot \boldsymbol{t} = \boldsymbol{v}_j \cdot \boldsymbol{t} \tag{5.46}$$

or the *no slip condition*

$$\boldsymbol{v}_g = \boldsymbol{v}_j \tag{5.47}$$

where $\boldsymbol{v}_g$ denotes the velocity to the adjacent fluid, $\boldsymbol{n}$ and $\boldsymbol{t}$ respectively denote the outer normal and tangential vector of the boundary. The Neumann condition on the free surface elevation η reads

$$\frac{\partial \eta}{\partial n} = 0, \tag{5.48}$$

$$\eta_g = \eta_j. \tag{5.49}$$

Ghost particle approach has some advantages such as, it is easy to implement and code for plane boundaries, it is computationally efficient, it prevents efficiently particle penetration and restores consistency at the boundaries. However, it has some disadvantages such as, in computational domains with corners there is the duplication of ghost particles and also for curved boundaries, there is the generation of coarser or finer particle distribution. In this thesis, we have employed the ghost particles approach to solve a discontinuous Riemann problem in Chapter 7.

5.3.2 Time Integration

Courant Condition

In order to integrate in time the particle positions $\boldsymbol{r}_i$ and velocities $\boldsymbol{v}_i$, there are approaches to mention a few, i.e., Euler, Runge-Kutta, leap-frog integration schemes. Explicit time integration schemes are subject to the CFL condition for stability reasons. Since, the CFL condition explains that the computational domain of dependence in the numerical simulation must include the physical domain of dependence, in another words the maximum speed of numerical propagation must exceed the maximum speed of physical propagation. Particularly in SPH, the CFL condition requires that the time step is proportional to the smallest spatial particle resolution which denotes the smallest smoothing length. The time step is determined by the Courant condition

$$\Delta t = C_{cfl} \text{min} \left(\frac{h}{c} \right) \tag{5.50}$$

where h is the smoothing length taken as $h = \text{min}(h_i, h_j)$, C_{cfl} is the Courant number and c is the maximum signal velocity between particle pairs.

Euler Scheme

The Euler scheme is the simplest time integration scheme. The particle positions are updated for particle i by

$$\boldsymbol{r}_i^{n+1} = \boldsymbol{r}_i^n + \Delta t \boldsymbol{v}_i^n \tag{5.51}$$

The Euler method is first order accurate.

Runge-Kutta Scheme

Let $L(\boldsymbol{r}) = D\boldsymbol{r}/Dt$ be an operator representing particle velocities. The optimal second order total variation diminishing (TVD) Runge-Kutta is given by

$$\boldsymbol{r}^{(1)} = \boldsymbol{r}^n + \Delta t L(\boldsymbol{r}^n), \tag{5.52}$$

$$\boldsymbol{r}^{n+1} = \frac{1}{2}\boldsymbol{r}^n + \frac{1}{2}\boldsymbol{r}^{(1)} + \frac{1}{2}\Delta t L(\boldsymbol{r}^{(1)}). \tag{5.53}$$

In the same vein, the optimal third order TVD Runge-Kutta is given by

$$\boldsymbol{r}^{(1)} = \boldsymbol{r}^n + \Delta t L(\boldsymbol{r}^n), \tag{5.54}$$

$$\boldsymbol{r}^{(2)} = \frac{3}{4}\boldsymbol{r}^n + \frac{1}{4}\boldsymbol{r}^{(1)} + \frac{1}{4}\Delta t L(\boldsymbol{r}^{(1)}), \tag{5.55}$$

$$\boldsymbol{r}^{n+1} = \frac{1}{3}\boldsymbol{r}^n + \frac{2}{3}\boldsymbol{r}^{(2)} + \frac{2}{3}\Delta t L(\boldsymbol{r}^{(2)}). \tag{5.56}$$

Leap-Frog Scheme

In the leap-frog as defined by Hernquist and Katz [53], the particle positions and velocities are updated for particle i by

$$\boldsymbol{r}_i^{n+1/2} = \boldsymbol{r}_i^{n-1/2} + \Delta t \boldsymbol{v}_i^n, \tag{5.57}$$

$$\boldsymbol{v}_i^{n+1} = \boldsymbol{v}_i^n + \Delta t \boldsymbol{a}_i^{n+1/2}. \tag{5.58}$$

The velocity is updated in two stages to maintain second order accuracy. Initially, a predicted estimate, $\boldsymbol{v}_i^{n+1/2}$ is obtained by

$$\boldsymbol{v}_i^{n+1/2} = \boldsymbol{v}_i^n + \frac{1}{2}\Delta t \boldsymbol{a}_i^{n-1/2}. \tag{5.59}$$

The value of $\boldsymbol{v}_i^{n+1/2}$ is then used to compute the time-centered acceleration, $\boldsymbol{a}_i^{n+1/2}$ which now allows the velocity to be updated via $\boldsymbol{v}_i^{n+1}$.

5.3.3 Nearest Neighbor Search

In this section, the neighboring search strategy used will be discussed. Since, SPH is a truly meshless method with a Lagrangian description. The search for surrounding particles j of the focal particle i at the position $\boldsymbol{x}_i$ is a major challenge that must be solved efficiently. This is because the neighboring search has to be done at every time step for each particle. In SPH, the neighboring particles are not known a priori, hence the SPH approximation of a field variable requires the search of neighboring particles j. Hence from a computational point of view, most of the time is spent searching for nearest neighbor particles. this is often computationally expensive. So to this effect, we will delve into neighboring search techniques which are less computationally demanding at least for the numerical examples considered in this thesis.

Pair-wise Search

The pair-wise search is a direct and simple search algorithm. As we illustrate in Fig. 5.4, for a given particle i given in red color, the pair-wise search calculates the distance $\boldsymbol{r}_{ij}$ from particle i to every neighboring particle j, for $j = 1, 2, \cdots, N$, where N is the total number of particles. If the relative distance $\boldsymbol{r}_{ij}$ is smaller than the dimension of the support domain of particle i, κh, then particle j is found to belong to the support domain of particle i. Hence, i and j are pairs of neighboring particles. This search is done for all the particles. This means, the pair-wise search is done for particles $i = 1, 2, \cdots, N$ and done for all neighbor particles $j = 1, 2, \cdots, N$. Clearly, the computational complexity of the pairwise search is of the order $O(N^2)$. This approach will not be practically useful for problems with large number of particles since the search will be done at all time steps.

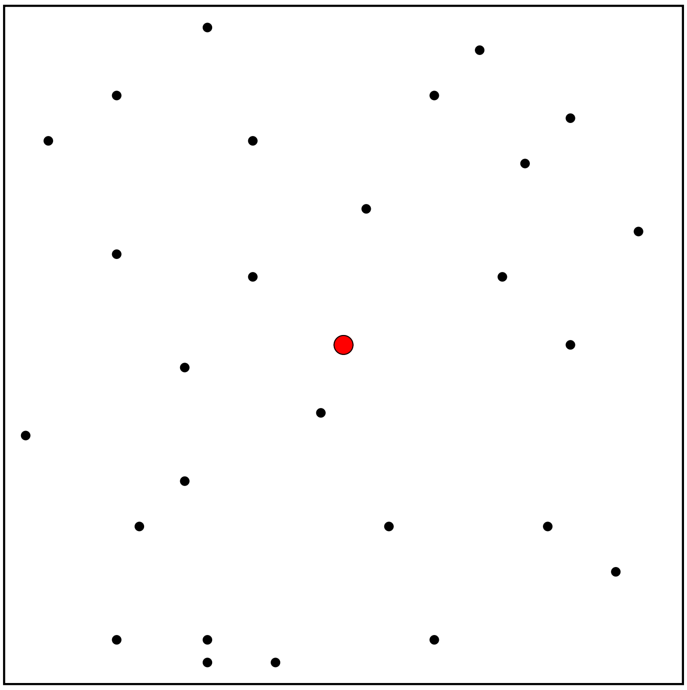


Figure 5.4: Pairwise Search Technique

Linked-list Nearest Neighbor Search

In this thesis, the following strategy called *linked-list* search algorithm is used, we define a background fictitious Cartesian grid as shown in Fig. 5.5. This grid contains the fluid domain with a mesh size of $2L$, the grid is kept fixed all through the simulation. Within the grid, comprises of macro cells which consists of particles as we can see details in [91]. The idea is analogous to the bookkeeping cells as used by Monaghan in [89], only particles in the neighboring cells can contribute to the value of a fluid variable in a given cell. To compute the free surface elevation η and the fluid velocity $\boldsymbol{v}$, only particles inside the same macro-cell or in the immediate surrounding macro-cells will contribute. Ferarri et al. [46] explained the efficiency of the neighboring search. The idea is building the list of particles in a given macro cell and also the indices pointing to macro-cell containing the particle. The coordinates of each particle in the macro-cell is stored in an integer format, the indices are computed from the particle position $\boldsymbol{x}_i$ by

$$\boldsymbol{x}_i^* = \text{floor}\left(\frac{\boldsymbol{x}_i}{2L}\right) \tag{5.60}$$

This strategy of storing coordinates of each particle reduces the time for accessing data in the neighbor search technique. So in our neighboring search, a particle can only interact with particles in its macro-

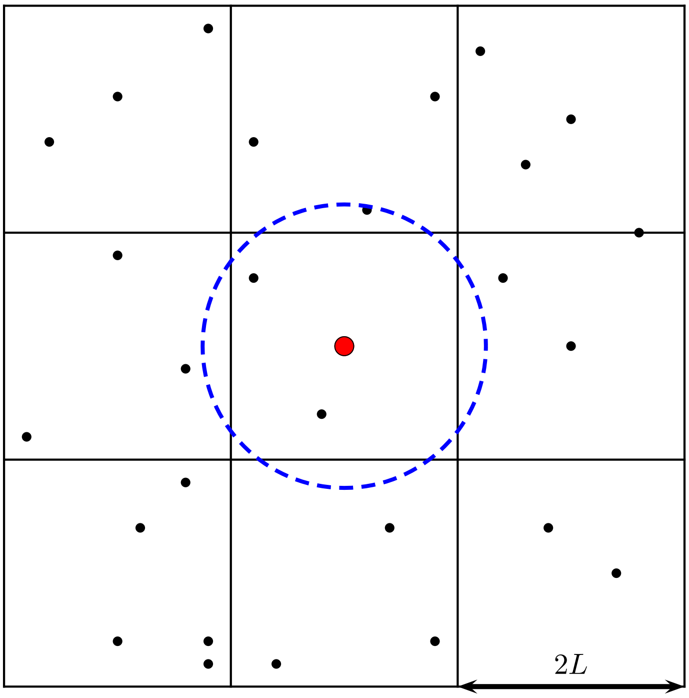


Figure 5.5: Fictitious Cartesian Grid: Neighboring search is done within the 9 cells in a two-dimensional space. The smoothing length is constant and the support domain for the particles is $2L$.

cell or in the neighboring macro-cells. As we can see in Fig. 5.5, we loop over the bounding box of 9 macro-cells. Once the neighbors of any particle is easily identified, we build the list of particles inside a given macro-cell and the list of indices pointing to the macro-cell containing a given particle. This operation is updated periodically, computationally cheap when compared to the pair-wise algorithm.

Tree Search

The tree search algorithm is very good for problems with variable smoothing lengths. The idea is creating trees that are ordered with respect to the particle positions. After the tree structure is created, it can be used to find the nearest neighboring particles. The tree technique recursively splits the global problem domain into octants which contains particles, until the leaves on the tree are individual particles. We refer to the book of Liu and Liu [80] for details.

5.4 The Conjugate Gradient Algorithm

The conjugate gradient algorithm (CG) discovered by Hestenes and Stiefel [54] is the original Krylov subspace iteration and one of the mainstays of scientific computing. The conjugate gradient algorithm is one of the best known iterative techniques for solving sparse symmetric positive definite linear systems.

Definition 5.4.1 (Krylov Subspaces). *Given a matrix $A \in \mathbb{R}^{N\times N}$ and a vector $b \in \mathbb{R}^N$, the sequence of Krylov subspaces is defined as*

$$\mathcal{K}_n(A,b) = \textit{span}\left\{b, Ab, \ldots, A^{n-1}b\right\} \subset \mathbb{R}^N, \qquad n = 0, 1, \ldots \tag{5.61}$$

Clearly, the Krylov subspaces are increasing, i.e., it holds that $\{0\} = \mathcal{K}_0(A, b) \subset \mathcal{K}_1(A, b) \subset \dots$.

Definition 5.4.2 (A-conjugate). *Given a matrix $A \in \mathbb{R}^{N \times N}$ that is symmetric, positive definite. The given vectors $d_0, d_1, \dots d_{m-1} \in \mathbb{R}^N \backslash \{0\}$ are called A-conjugate if the following holds:*

$$\langle Ad_k, d_j \rangle_2 = 0 \qquad k \neq j \tag{5.62}$$

Remark 5.4.3. *The A-conjugate property is therefore equivalent to pairwise orthogonality with respect to the scalar product $\langle \cdot, \cdot \rangle_A$.*

The CG method is a realization of the of an orthogonal projection technique onto the Krylov subspace $\mathcal{K}_n(r_0, A)$ where r_0 is the initial residual.

By assuming that A is not only real and symmetric but also positive definite, this means that the eigenvalues of A are all positive, or in other words, we have $x^T Ax > 0$ for every nonzero $x \in \mathbb{R}^N$. With this assumption, the function $\| \cdot \|_A$ defined by

$$\| \cdot \|_A = \sqrt{x^T Ax} \tag{5.63}$$

is a norm on $\mathbb{R}^N$ called the $A\text{–}norm$. Basically, the vector whose $A\text{–}norm$ we will consider is $e_n = x_n - x_*$, the error at step n. The conjugate gradient iteration can be paraphrased as follows. It is a system of recurrence formulas that generates the unique sequence of iterates $\{x_n \in \mathcal{K}_n\}$ with the property that at step n, $\|e_n\|_A$ is minimized.

The matrix based CG method is summarized in the following algorithm.

Algorithm 1 Calculate x

```
Input: A, b
Compute $r_0 := b - Ax_0$, $p_0 := r_0$.
For $j = 0, 1, \cdots$, until convergence Do:
    $\alpha_j := \frac{(r_j, r_j)}{(Ap_j, p_j)}$
    $x_{j+1} := x_j + \alpha_j p_j$
    $r_{j+1} := r_j - \alpha_j p_j$
    $\beta_j := \frac{(r_{j+1}, r_{j+1})}{(r_j, r_j)}$
    $p_{j+1} := r_{j+1} + \beta_j p_j$
End For
Output: $x$
```

In algorithm (1), together with the matrix $\boldsymbol{A}$, we need to store four vectors, x, p, Ap and r. The CG algorithm admits an existence and unique solution property. The following theorem explains the minimal property valid for orthogonal residual technique.

Theorem 5.4.4. *Given a symmetric, positive definite matrix $A \in \mathbb{R}^{N \times N}$ and, for $n = 1, 2, \dots,$ the vectors x_n determined by the orthogonal residual are uniquely determined, it holds that*

$$\|x_n - x_*\|_A = \mathit{min} \|x_n - x_*\|_A, \qquad n = 1, 2, \dots, \tag{5.64}$$

Proof. To prove uniqueness, we consider two vectors x_n, $\hat{x}_n$, with n fixed satisfying

$$\left.\begin{aligned} x_n &\in \mathcal{D}_n, \\ Ax_n - b &\in \mathcal{D}_n^{\perp} \end{aligned}\right\} \tag{5.65}$$

where $\mathcal{D}_n$ are finite or infinite linear subspaces. Then,

$$\langle A(x_n - \hat{x}_n), x_n - \hat{x}_n \rangle_2 = 0 \qquad \rightsquigarrow \qquad x_n = \hat{x}_n. \tag{5.66}$$

where $A(x_n - \hat{x}_n) \in \mathcal{D}_n^{\perp}$ and $x_n - \hat{x}_n \in \mathcal{D}_n$

To prove existence one proceeds as the following with an arbitrary basis $d_0, d_1, \dots, d_{m-1}$ of $\mathcal{D}_n$,

$$x_n = \sum_{j=0}^{m-1} \alpha_j d_j \tag{5.67}$$

from which x_n satisfies

$$Ax_n - b \in \mathcal{D}_n^{\perp} \tag{5.68}$$

where

$$\begin{aligned} \langle Ax_n - b, d_k \rangle_2 &= 0 \qquad \text{for} \qquad k = 0, \dots, m-1, & (5.69) \\ \sum_{j=0}^{m-1} \langle Ad_j, d_k \rangle_2 \alpha_j &= \langle b, d_k \rangle_2 \qquad \text{for} \qquad k = 0, \dots, m-1, & (5.70) \end{aligned}$$

which represents a system of m linear equations for m coefficients $\alpha_0, \dots, \alpha_{m-1}$. With the uniqueness already shown above, the system of equations is solvable.

To prove the minimal property, one calculates the following for an arbitrary vector $x \in \mathcal{D}_n$

$$\begin{aligned} \|x - x_*\|_A^2 &= \|x_n - x_* + x - x_n\|_A^2 & (5.71) \\ &= \|x_n - x_*\|_A^2 + 2\langle A(x_n - x_*), x - x_n \rangle_2 + \|x - x_n\|_A^2 & (5.72) \\ &\geq \|x_n - x_*\|_A^2 & (5.73) \end{aligned}$$

Since, $\langle A(x_n - x_*), x - x_n \rangle_2 = 0$, this completes the proof. □

Remark 5.4.5. *We wish to remark that in the earlier theorem, the spaces* $\mathcal{D}_n = \mathcal{K}_n(A, b)$ *are considered for symmetric, positive definite matrix* $A \in \mathbb{R}^{N \times N}$.

Theorem 5.4.6. *Let the CG iteration (Algorithm (1)) be applied to a symmetric positive definite matrix problem* $Ax = b$. *As long as the iteration has not yet converged that is,* ($r_j \neq 0$)*, the algorithm proceeds without divisions by zero, and we have the following identities of subspaces*

$$\begin{aligned} \mathcal{K}_n &= \langle x_1, x_2, \dots, x_n = \langle p_0, p_1, \dots, p_{n-1} \rangle & (5.74) \\ &= \langle r_0, r_1, \dots, r_{n-1} = \langle b, Ab, \dots, A^{n-1} b \rangle & (5.75) \end{aligned}$$

Moreover, the residuals are orthogonal,

$$r_n^T r_j = 0 \qquad (j < n), \tag{5.76}$$

and the search directions are A-conjugate,

$$p_n^T A p_j = 0 \qquad (j < n), \tag{5.77}$$

The proof can be found in [108].

The CG method will converge to the exact solution in at most N steps for N degrees of freedom. From the iterates step k, there is a projection of the exact solution into the solution space generated by the A-conjugate basis vectors. The CG method will converge to an acceptable error-tolerance in less than N iterations. The convergence and error of the CG algorithm is given in the following theorem.

Theorem 5.4.7. *Given matrix $A \in \mathbb{R}^{N\times N}$ that is symmetric, positive definite, then the following error estimate holds for the CG method:*

$$\|x_n - x_*\|_A \leq 2\gamma^n \|x_*\|_A \qquad n = 0, 1, \ldots, \tag{5.78}$$

$$\|x_n - x_*\|_2 \leq 2\sqrt{\kappa_A}\gamma^n \|x_*\|_2 \qquad n = 0, 1, \ldots, \tag{5.79}$$

with $\kappa_A := \textbf{cond}_2(A)$ and

$$\gamma := \frac{\sqrt{\kappa_A} - 1}{\sqrt{\kappa_A} + 1}. \tag{5.80}$$

where x_* is the exact solution, and x_n is the approximate solution after n steps

The proof can be found in [99]. The greatest strength of the CG method is the fast convergence rate.

5.4.1 The Matrix-free Conjugate Gradient Algorithm

Because of the computational cost that will be needed in storing matrices and vectors coupled with the fact that the SPH method is inherently computationally demanding, the matrix-based version of the CG algorithm will consume a lot of computational time. To this effect, we give the *matrix-free* version of the CG algorithm. The matrix-free version do not explicitly store the matrix coefficients however, it access the matrix by calculating matrix-vector products. Matrix-free methods are much better to be employed if the matrix is very large such that its storage and other operations on the matrix will consume a lot of computational time and computer memory.

The construction of our iterative procedure inside the SISPH scheme does not involve the actual matrix A. Therefore the matrix-vector product Aw_j in the conjugate gradient algorithm is done as follows: In this thesis, we develop a user-defined matrix-vector product which is called inside the CG algorithm (1). The user-defined matrix-vector product is done per particle as we see in the following algorithm (2). Let the matrix-vector product $y \mapsto Aw$.

Algorithm 2 Matrix-vector calculation of $y = Aw$

```
Input: A, w
Guess $y := 0$
For $i = 1, 2, \ldots$ N(# of Particles), Do:
    For $j = 1$, Neigh(# of Neighbors) Do:
        $y_i := y_i + A_{ij} w_j$
    End For
End For
Output: $y$
```

We therefore call our matrix-free function with the vector w as input and receive as output the vector y without ever storing or creating the system matrix A. By knowing the structure of matrix A, we create a function which calculates the matrix-free matrix-vector products between matrix A, and thus we program this without the need to store A, with this construction we save a valuable portion of memory. Having

Algorithm 3 Matrix-free matrix-vector to calculate x

```
Input: A, b
Call matrix-vector product - algorithm (2) $y := Ax$
Compute $r_0 := b - Ax_0$, $p_0 := r_0$.
For $j = 0, 1, \cdots$, until convergence Do:
    Call matrix-vector product - algorithm (2) $y := Ap$
    $\alpha_j := \frac{(r_j, r_j)}{(Ap_j, p_j)}$
    $x_{j+1} := x_j + \alpha_j p_j$
    $r_{j+1} := r_j - \alpha_j p_j$
        If ($\|r\| < \epsilon$) Then
            Exit
        End If
    $\beta_j := \frac{(r_{j+1}, r_{j+1})}{(r_j, r_j)}$
    $p_{j+1} := r_{j+1} + \beta_j p_j$
End For
Output: $x$
```

calculated the matrix-vector product, we now incorporate our user-defined matrix-vector product and hence we arrive at the matrix-free version of the CG algorithm used in this thesis in the algorithm(3).

6. Wetting and Drying Semi-implicit SPH Methodology for Shallow Water Flows

Following the SISPH scheme presented in the preceding chapter, we now consider a semi-implicit SPH wetting and drying technique in this chapter.

6.1 Introduction

In this section of the thesis, we propose a new *wetting and drying* semi-implicit SPH algorithm applied to the shallow water equations. We consider the inviscid hydrostatic free surface flows. Such flows are governed by the *shallow water equations* which we can derive by vertically or laterally averaging the fully three dimensional incompressible Navier-Stokes equations with the assumption of a hydrostatic pressure distribution (see [28, 29]).

Wetting and drying is a common phenomena in shallow water flows where water level rises, called *wetting*, and where water level recedes, called *drying*. This process can occur during events, such as inundation of coastal regions that are often due to storm surges and wave driven run-up on beaches, even more in biological processes i.e., during the drying phase on a tidal mud flat algal mats [52]. These processes occur on periodic time intervals. Since the shallow water equations are well defined in a fully wetted region in the domain, when water height recedes and goes to zero, this affects the numerical solution of the equations, where the arising problems may become ill-posed. Viable approaches to tackle such problems are essentially incorporating wetting and drying into the numerical scheme or a dynamic adaptivity in the computational domain as the water level moves. Pioneering work in wetting and drying on two-dimensional shallow water equations is due to Leendertse [72], whose approach makes use of an alternating direction implicit ADI method to discretize the governing equations. There is a considerable amount of work relying on finite volume and finite element schemes to treat wetting and drying, e.g., [6, 17, 24, 33, 59, 61, 102, 118], to mention but a few.

All these techniques make use of mesh adaptation (by deforming domains and meshs) and mesh reduction. The latter is by putting 'screens' at velocity points of the flow configuration when the water height drops below a certain *drying threshold* and removing the screens when the water height rises above a *wetting threshold*. This approach is problem-dependent and the threshold parameters must be tuned, where the thin water layer technique uses a fixed mesh to maintain a thin layer of water in nominally dry elements. Vater, Beisiegel and Behrens [114] propose a limiter-based approach in the velocity and water height to prevent instabilities.

The treatment of wetting and drying in shallow water equations using a truly meshfree numerical method is a new approach. In fact, to the best of our knowledge, only [110, 112] solved the Thacker's test case [106] and flooding problem with a shallow water SPH model using a dynamic particle coalescing and splitting method.

This thesis proposes a new wetting and drying semi-implicit *Smoothed Particle Hydrodynamics* (SPH) algorithm for the numerical solution of the shallow water equations, following the semi-implicit SPH scheme in [8]. The wetting and drying relies on the work of Casulli [26] for unstructured meshes, where the resulting numerical algorithm can directly be developed from the governing equations. In this way, a correct mass balance is assured in wet particle regions and in transition regions, i.e., particle regions, from wet to dry and from dry to wet regions, with maintaining a nonnegative water height. The approach taken boils down to solving a *mildly nonlinear* system. When wetting and drying occurs, more iterations are needed for the solution of the mildly nonlinear system.

6.2 Wetting and Drying Methodology

In this section we introduce the methodology employed towards the construction of our proposed wetting and drying semi-implicit SPH algorithm. Fig. 6.1 depicts a simple hydraulic wetting and drying pattern. When the water level drops below or above a threshold water depth, a null value is assigned to the particle velocity components at dry particles and set the free surface elevation to the bed elevation and at a wetted particle, since the boundary is moving, the moving boundary is evaluated by extrapolating the free surface elevation at surrounding wet particles. Below the free surface, the domain is fully wetted with a non vanishing velocity i.e., $\boldsymbol{v} \neq 0, H > 0$ and at the dry region both velocity and total water depth is zero, $\boldsymbol{v} = 0, H = 0$.

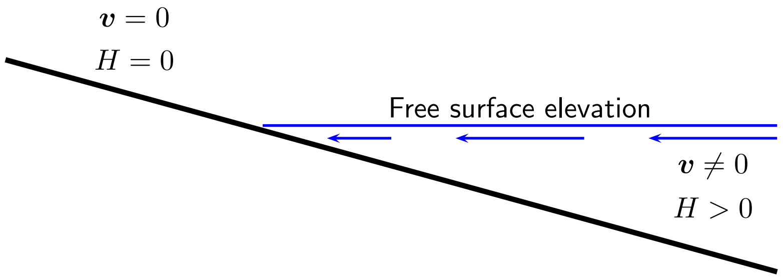


Figure 6.1: Wetting and drying hydraulic pattern

6.2.1 Subparticle Modeling

When wetting and drying processes are modeled and simulated, the shallow water equations are defined on a time dependent domain $\Omega(t)$ as

$$\Omega(t) = \{(x, y) : H(x, y, t) > 0\} \tag{6.1}$$

where $\Omega(t)$ is intrinsically one of unknowns to be determined numerically. Also, since the fluid boundary is also moving and one can not determine the position *a priori*. To circumvent this difficulty, Casulli [26] defined a piecewise constant function. For a specified bathymetry $h(x, y)$ we give a precise description of the flow by a *saturation* function $a(x, y, z)$ defined by

$$a(x, y, z) = \begin{cases} 1 & \text{for} \quad h(x, y) + z > 0 \\ 0 & \text{otherwise} \end{cases}$$

for $(x, y) \in \Omega$ and $-\infty < z < \infty$. At $z = \eta_i^n$, the horizontal integral for each particle i at time level $t = t^n$ given by

$$a_i(\eta_i^n) = \int\limits_{\Omega_i} a(x, y, \eta_i^n) dxdy \tag{6.2}$$

represents the free-surface area. We can state clearly that when $a_i(\eta_i^n) = 0$, the ith particle is dry, when $a_i(\eta_i^n) = V_i$, the ith particle is wet and when $0 < a_i(\eta_i^n) < V_i$, the ith is partially wet respectively. The piecewise constant function defined by $a(x, y, z)$ means that $a_i(\eta_i^n)$ is nonnegative, nondecreasing and bounded. For each particle i, the total water depth is given by

$$\begin{aligned} H(x, y, \eta_i^n) &= \int\limits_{-\infty}^{\eta_i^n} a(x, y, z) dz \\ &= \max\left[0, h(x, y) + \eta_i^n\right] \end{aligned} \tag{6.3}$$

so that $H(x, y, \eta_i^n) \geq 0$, and strict inequality identifies a wet particle. Hence, the wet region is given by

$$\Omega_i^n = \{(x, y) \in \Omega_i : H(x, y, \eta_i^n) > 0\} \tag{6.4}$$

The *water volume* for particle i is given by

$$V_i(\eta_i^n) = \int\limits_{-\infty}^{\eta_i^n} a_i(z) dz = \int\limits_{\Omega_i} H(x, y, \eta_i^n) dxdy \tag{6.5}$$

Because $a_i(\eta_i^n)$ is nonnegative, nondecreasing and bounded, we have $V_i(\eta_i^n) \geq 0$ and strict inequality necessarily implies $a_i(\eta_i^n) > 0$, in particular $0 \leq a_i(\eta_i^n) \leq V_i$. Clearly when

- $a_i(\eta_i^n) = 0$, the ith particle is dry
- $a_i(\eta_i^n) = V_i$, the ith particle is wet
- $0 < a_i(\eta_i^n) < V_i$, the ith particle is partially wet

6.2.2 The Free Surface Equation and Mass Conservation

Substituting the discrete momentum equation into the discrete continuity equation. The model is reduced into a smaller model in η_i^{n+1} as the only unknown. Multiplying (5.40) by ω_i and inserting (5.39) into (5.40) we obtain

$$V(\eta_i^{n+1}) - g\Theta^2 \frac{\Delta t^2}{\delta_{ij}} \sum_{j=1}^{N} 2\omega_i\omega_j \left[H_{ij}^n(\eta_j^{n+1} - \eta_i^{n+1}) \nabla \boldsymbol{W}_{ij} \cdot \boldsymbol{n}_{ij}\right] = \boldsymbol{b}_i^n, \tag{6.6}$$

where the right hand side $\boldsymbol{b}_i^n$ represents the known values at time level t^n given as

$$\begin{aligned} \boldsymbol{b}_i^n &= V(\eta_i^n) - \Delta t \sum_{j=1}^{N} 2\omega_i\omega_j H_{ij}^n \boldsymbol{F} \boldsymbol{v}_{ij}^{n+\Theta} \nabla \boldsymbol{W}_{ij} \cdot \boldsymbol{n}_{ij} \\ &+ g\Theta(1 - \Theta) \frac{\Delta t^2}{\delta_{ij}} \sum_{j=1}^{N} 2\omega_i\omega_j \left[H_{ij}^n(\eta_j^n - \eta_i^n) \nabla \boldsymbol{W}_{ij} \cdot \boldsymbol{n}_{ij}\right], \end{aligned} \tag{6.7}$$

where $V(\eta_i^{n+1})$ is the water volume where the *nonlinearity* resides as we have defined in the piecewise constant saturation function $a(x,y,z)$, $\boldsymbol{Fv}_{ij}^{n+\Theta} = \Theta\boldsymbol{Fv}_{ij}^n + (1-\Theta)\boldsymbol{v}_{ij}^n$. Since H_{ij}^n, ω_i, ω_j are non-negative numbers, equations (6.6) - (6.7) constitute a nonlinear system of N equations for η_i^{n+1} unknowns due to the piecewise constant water volumes.

Having computed the free surface and water velocity, the new total depth H_{ij}^{n+1} has to be updated. Since, the bathymetry h_{ij} are specified at the locations. A negative value for H is physically meaningless, then our discrete total depth H_{ij} at the next time are defined as

$$H_{ij}^{n+1} = \max(0, h_{ij}^{n+1} + \eta_i^{n+1}, h_{ij}^{n+1} + \eta_j^{n+1}) \tag{6.8}$$

where we note that $H_{ij} = H_{ji}$.

But a zero value for H simply means a particle is dry which can be later on wetted when the total water depth H becomes positive. So, if H is positive, the particle is wet and the vertical variation of the particle will be non zero whereas when H is zero, the particle is dry and the particle's vertical variation will be zero.

In this numerical model, considering Equation (6.6) we can inspect clearly that the resulting semi-implicit SPH equation for the free surface equation accurately accounts for the treatment of positive and zero values for the total water depth H. We can further see that the treatment of wetting and drying is naturally present in the present study without taking into account special treatment. And this formulation guarantees mass conservation while accounting for wetting and drying fronts. When the total water depth of a particle is zero, this implies a no mass flux or a zero velocity until at a later time when H becomes positive. In Equation (6.6), if we set H to be zero, the free surface equation becomes that the water volume at time level $n+1$ equals water level at time level n. This means there is no variation in the free surface elevation for a dry particle. On a dry particle the velocity equations are replaced by $\mathbf{v}_{ij}^{n+1} = 0$, so when wetting and drying of particles occurs, we still solve the same SPH equations having satisfied the condition of no mass flux.

In the entire particle configuration, when the total water depth is zero, $H_{ij}^n = 0$, the free surface equation (6.6) trivially implies

$$V(\eta_i^{n+1}) = V(\eta_i^n), \tag{6.9}$$

hence we can assume

$$\eta_i^{n+1} = \eta_i^n. \tag{6.10}$$

In this scenario, equation (6.6) does not form part of the system to be constructed. The remaining set of the free surface equation i.e., where there exist at least one H_{ij}^n that is nonzero the system is assembled into a *mildly nonlinear* sparse system for η_i^{n+1}. Brugnano and Casulli have presented convergent iterative schemes to solve this system even for piecewise polynomials for the definition of the water volume $V(\eta)$, (see [22, 23]) for details.

6.3 Solution Algorithm

6.3.1 Mildly Nonlinear System

We hereby write system (6.6) in a compact vector notation:

$$\mathbf{V}(\eta) + \mathbf{T}\eta = \mathbf{b} \tag{6.11}$$

where

$$\mathbf{V}(\eta) = \begin{pmatrix} V_1(\eta_1) \\ V_2(\eta_2) \\ \vdots \\ V_N(\eta_N) \end{pmatrix}, \qquad \eta = \begin{pmatrix} \eta_1^{n+1} \\ \eta_2^{n+1} \\ \vdots \\ \eta_N^{n+1} \end{pmatrix}, \qquad \boldsymbol{b} = \begin{pmatrix} b_1^n \\ b_2^n \\ \vdots \\ b_N^n \end{pmatrix},$$

where **T** is a sparse and symmetric $N_\eta \times N_\eta$ matrix which comes from the second and third term in the left hand side of equation (6.6), matrix $\boldsymbol{T}$ is positive definite, then its inverse is also positive definite, **b** is a vector of N_η components from the right hand side of equation (6.6). Let us assume that the matrix **T** is irreducible. From Equation (6.6) we write the coefficient of η_i^{n+1} the ith main diagonal element of the matrix **T** given as

$$t_{i,i} = g\Theta^2 \frac{\Delta t^2}{\delta_{ij}} \sum_{j=1}^{N} 2\omega_i\omega_j H_{ij}^n \nabla \mathbf{W}_{ij} \cdot \mathbf{n}_{ij} \tag{6.12}$$

In the same vein, if we consider the non-zero off diagonal elements in each row of the matrix **T** which represents the coefficients of η_j^{n+1} in Equation (6.6), we have

$$t_{i,j} = -g\Theta^2 \frac{\Delta t^2}{\delta_{ij}} \sum_{j=1}^{N} 2\omega_i\omega_j H_{ij}^n \nabla \mathbf{W}_{ij} \cdot \mathbf{n}_{ij} \tag{6.13}$$

From the assumption that **T** to be irreducible we have that $t_{i,j} \leq 0$ for all particles $i = 1, 2. \cdots N$, so we have atleast one of $t_{i,j}$ is nonzero. From the above, such that $t_{i,i} > 0$ for each particle i and $t_{i,j} \leq 0$ whenever particle i is different from particle j, $i \neq j$, then we conclude that the **T** is an irreducible symmetric and positive semidefinite matrix. We can say that $\sum_{j=1}^{N} t_{i,j} = 0$ for $i = 1, 2. \cdots N$. If we denote for any nonzero diagonal matrix by **P**, from the above considerations, we have $\mathbf{P} \geq 0$, then we have that $\mathbf{P} + \mathbf{T}$ is an irreducible symmetric M-matrix. Therefore, $\mathbf{P} + \mathbf{T}$ is positive definite and consequently $(\mathbf{P} + \mathbf{T})^{-1} > 0$. From a physical point of view, the contribution of matrix **T** denotes the mass fluxes between pair of interacting particles.

For clarity, we define the water volumes and its corresponding gradient as

$$\mathbf{V}(\eta) = \begin{cases} \eta + h & \text{if} \quad \eta + h > 0 \quad \text{for} \quad \text{wet case} \\ 0 & \text{if} \quad \eta + h \leq 0 \quad \text{for} \quad \text{dry case} \end{cases} \tag{6.14}$$

The gradient of the water volumes is given as

$$\frac{\partial \mathbf{V}}{\partial \eta} = \begin{cases} 1 & \text{if} \quad \eta + h > 0 \quad \text{for} \quad \text{wet case} \\ 0 & \text{if} \quad \eta + h \leq 0 \quad \text{for} \quad \text{dry case} \end{cases} \tag{6.15}$$

The matrix **P** evaluated at η_i corresponds to the diagonal entries

$$\mathbf{P} = \left(\frac{\partial \mathbf{V}}{\partial \eta} \right) \tag{6.16}$$

evaluated at the ith particle and **P** is given as

$$\mathbf{P} = \text{diag}(a_{11}, \ldots, a_{1n}) \tag{6.17}$$

where a_{ij} corresponds to the gradient of the water volume.

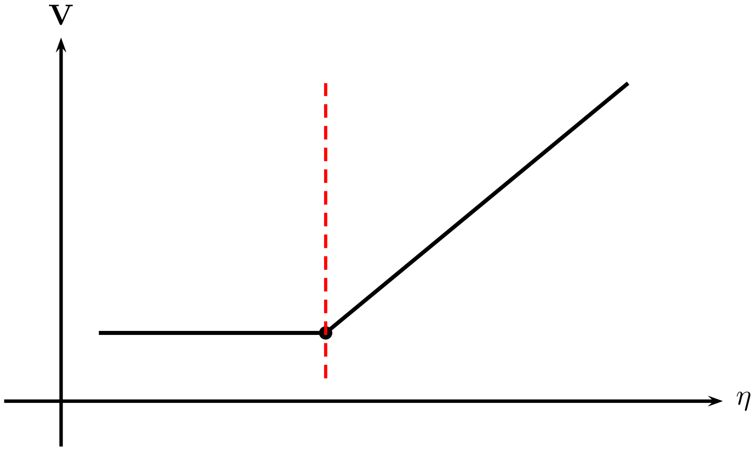


Figure 6.2: Non-differentiability of water volume

From the definition of the water volumes and in Fig. 6.2, we see that the function is not differentiable in zero i.e., at the black dot (red broken lines).

Before solving the free surface equation, we will state under the condition whereby the solution to (6.11) exists and is unique.

Theorem 6.3.1. *Let the vertical integral of the surface area for particle i be given as*

$$V_i(\eta_i) = \int_{-\infty}^{\eta_i} a_i(z)dz \tag{6.18}$$

with $a_i(z)$ nonnegative, nondecreasing and bounded $\forall\ i$. Also, let **T** *be an irreducible, symmetric and positive semidefinite matrix such that $t_{i,j} \leq 0$ whenever $i \neq j$ and $\sum_{j=1}^{N} t_{i,j} = 0$ for $i = 1, 2. \cdots N$. If $\sum_{j=1}^{N} b_j > 0$, then there is existence and uniqueness of solution to the system (6.11).*

Proof. Assume η^α and η^β solves system (6.11). Therefore, we have

$$\mathbf{V}(\eta^\alpha) + \mathbf{T}\eta^\alpha = \mathbf{b} \tag{6.19}$$

and

$$\mathbf{V}(\eta^\beta) + \mathbf{T}\eta^\beta = \mathbf{b} \tag{6.20}$$

Let the difference between particle volumes at different free surface elevation η_i^α and η_i^β be given as

$$V_i(\eta_i^\beta) - V_i(\eta_i^\alpha) = \int_{-\infty}^{\eta_i^\beta} a_i(z)dz - \int_{-\infty}^{\eta_i^\alpha} a_i(z)dz \tag{6.21}$$

$$= \int_{\eta_i^\alpha}^{\eta_i^\beta} a_i(z)dz \tag{6.22}$$

$$= \hat{a}_i(\eta_i^\alpha, \eta_i^\beta)(\eta_i^\beta - \eta_i^\alpha) \tag{6.23}$$

where $\hat{a}_i(\eta_i^\alpha, \eta_i^\beta)$ represents the average water volumes. Substracting (6.19) from (6.20), we obtain

$$\Big[\mathbf{V}(\eta^\beta) + \mathbf{T}\eta^\beta\Big] - [\mathbf{V}(\eta^\alpha) + \mathbf{T}\eta^\alpha] = \Big[\hat{\mathbf{P}}(\eta^\alpha, \eta^\beta) + \mathbf{T}\Big]\,(\eta^\beta - \eta^\alpha) = 0 \tag{6.24}$$

$\hat{\mathbf{P}}(\eta^\alpha, \eta^\beta)$ is a diagonal matrix whose values are the nonnegative values $\hat{a}_i(\eta_i^\alpha, \eta_i^\beta)$.
Since $\sum_{j=1}^N b_j > 0$, From (6.11), it implies

$$\sum_{i=1}^{N} V_i(\eta_i^\alpha) = \sum_{i=1}^{N} V_i(\eta_i^\beta) = \sum_{j=1}^{N} b_i > 0 \tag{6.25}$$

Therefore, some $V_i(\eta_i^\alpha)$ and $V_i(\eta_i^\beta)$ are strictly positive. But since, $a_i(z)$ is nonnegative, nondecreasing, then the values $\hat{a}_i(\eta^\alpha, \eta^\beta)$ are strictly positive. Hence, $\hat{\mathbf{P}}(\eta^\alpha, \eta^\beta)$ is nonzero and nonnegative. We can further infer that $\hat{\mathbf{P}}(\eta^\alpha, \eta^\beta) + \mathbf{T}$ is an M-matrix, therefore matrix $\hat{\mathbf{P}}(\eta^\alpha, \eta^\beta) + \mathbf{T}$ is nonsingular.

From Equation (6.24) we have that

$$\eta^\alpha = \eta^\beta \tag{6.26}$$

Hence uniqueness follows. ☐

6.3.2 A Newton Method

We arrive at the piecewise system which is strongly diagonally dominant, symmetric and positive definite. Hence, a unique solution can be efficiently obtained by a matrixfree version of the conjugate gradient method and solved exactly in a Newton-type iteration. A nested Newton-type method can be see in the work of Casulli and Zanolli (see [31, 32]).

Initializing the free surface elevation η, for all $k = 1, 2, \cdots$ where $\eta^{k,0} = \eta^{k-1}$ a sequence of iterates η^μ is obtained from Equ. (6.11). Linearizing $\boldsymbol{V}(\eta)$ as follows we have,

$$\Big[\mathbf{V}(\eta^{k,\mu-1}) + \mathbf{P}(\eta^{k,\mu-1})(\eta^{k,\mu} - \eta^{k,\mu-1})\Big] + \mathbf{T}\eta^{k,\mu} = \mathbf{b}^{k-1}, \tag{6.27}$$

we obtain the iterates from the linear systems

$$(\mathbf{P}^{k,\mu-1} + \mathbf{T})\eta^{k,\mu} = \mathbf{g}^{k,\mu-1}, \qquad \mu = 1, 2, \cdots \tag{6.28}$$

where

$$\mathbf{P}^{k,\mu-1} = \mathbf{P}(\eta^{k,\mu-1}) \tag{6.29}$$

and

$$\mathbf{g}^{k,\mu-1} = \mathbf{b}^{k-1} - \mathbf{V}^{k,\mu-1} + \mathbf{P}^{k,\mu-1}\eta^{k,\mu-1}$$

The (k, μ)th residual $\boldsymbol{r}$ is given as

$$\boldsymbol{r}^{k,\mu} = \mathbf{V}(\eta^{k,\mu}) + \mathbf{T}\eta^{k,\mu} - \mathbf{b}, \tag{6.30}$$

and a stopping criterion for the iterates is given as $\|\boldsymbol{r}^{k,\mu}\| < \epsilon$ where ϵ is a sufficiently small tolerance value. The nonlinear problem to solve reads:

$$\eta^{k+1} = \eta^k - \Big[\mathbf{P}(\eta^k) + \mathbf{T}\Big]^{-1}\Big[\mathbf{V}(\eta^k) + \mathbf{T}\eta^k - \mathbf{b}\Big],\ \ k = 0, 1, \cdots \tag{6.31}$$

Algorithm 4 Calculate η

```
Input: V, P, T, b, and $\epsilon$
Do $k = 1, 2, \cdots$
    Set $\boldsymbol{P}^{k,0} = \boldsymbol{I}$
    Do $\mu = 1, 2, \cdots$
        Solve $[\mathbf{P}(\eta^{k,\mu-1} + \mathbf{T})]\eta^{k,\mu} = \mathbf{b} - \mathbf{P}(\eta^{k,\mu-1})$
        If $\|\boldsymbol{r}^{k,\mu}\| < \epsilon$
            set $\eta^k = \eta^{k,\mu}$ and Exit
        End If
    End Do
End Do
Output $\eta$
```

where k denotes the iteration index, $\mathbf{P}(\eta^k)$ is a diagonal matrix. The iterative scheme in (6.31) is hereby summarized into Algorithm (4).

Now coupled together with the semi-implicit algorithm presented in the preceding chapter, the one time step of the algorithm is summarized as follows:

1. Initialize $\boldsymbol{v}_{ij}^{n+1,0}$ and $\eta_{ij}^{n+1,0}$;
2. Newton-type iteration over $k = 1 \ldots N$:
    - Compute $\boldsymbol{v}_{ij}^{n+1,k+1}$, i.e., convective terms are discretized explicitly; then set $\boldsymbol{F}\boldsymbol{v}_{ij}^{n+1,k+1} := \boldsymbol{v}_{ij}^{n+1,k+1}$,
    - Discretize the gradient of $\boldsymbol{v}$ and η implicitly,
    - Substitute discrete momentum into discrete continuity equation,
    - Compute $\eta_{ij}^{n+1,k+1}$ by solving the free surface equation (6.6),
    - Update $\boldsymbol{v}_{ij}^{n+1,k+1}$ explicitly from (5.39),
3. Set $\boldsymbol{v}_{ij}^{n+1} = \boldsymbol{v}_{ij}^{n+1,k+1}$.

Once the free surface location η_i is computed. Equation (5.39) constitute a linear system for $\boldsymbol{v}_i^{n+1}$, the systems are independent of each other and are symmetric and positive definite. This is conveniently solved to determine $\boldsymbol{v}_i^{n+1}$ throughout the particle configurations and the particle positions can be subsequently updated. Following our mildly nonlinear construction in equation (6.6), a correct mass balance is always achieved in all particle regions irrespective of the specified bottom bathymetry. Nonnegative water volumes and water heights are assured.

7. Numerical Examples

7.1 The 1D and 2D Shallow Water Equations

In this section, following the semi-implicit SPH scheme that has been derived in Chapter 5 and the wetting and drying algorithm presented in Chapter 6, the scheme will be validated on the one and two dimensional shallow water equations test problems. In this section, three numerical examples will be validated namely: smooth solution, discontinuous solution and an oscillating lake in a parabolic basin to validate wetting and drying phenomena. In the following section, we present two dimensional test case of a collapsing Gaussian bump. In the subsequent test problems, the acceleration due to gravity constant g is set to $g = 9.81$.

7.1.1 Smooth Surface Wave Solution

In this example, we consider a smooth free surface wave propagation. We consider the following initial value problem within the domain $\Omega = [-1, 1]$ with the data

$$\eta(x,0) = 1 + \frac{1}{2} e^{-\frac{1}{2}(x^2/\sigma^2)}, \tag{7.1}$$

together with the initial condition

$$v(x,0) = 0, \tag{7.2}$$

with flat bottom, where $\sigma = 0.1$,

$$h(x) = 0. \tag{7.3}$$

The computational domain Ω is discretized with $N = 200$ particles. We simulate till the final time $t = 0.15$, a fixed time step is chosen to be $\Delta t = 0.01$, an implicitness factor $\Theta = 1$ is used. The numerical solution is given in Fig. 7.1. The upper profile in Figure 7.1 depicts the free surface elevation with a flat bottom bathymetry and the lower profile depicts the particle velocity. In Figure 7.2, to demonstrate that the method permits larger timesteps, we simulated for different times $\Delta t = 0.0025s, 0.005s, 0.01s$, a very good agreement can be seen. The tolerance for solving the linear system was set to $tol = 10^{-14}$. The total mass during the simulation is conserved as we see in Figure 7.3, the total mass is given as 2.125331. We solve the example with a matrix-free implementation of the CG method, without making use of any preconditioner. We compare our SISPH solution with a reference solution obtained by solving the one-dimensional shallow water equation with the finite difference mesh based approach of Casulli [25] on a fine mesh of $10,000$ points. The comparison between our numerical results obtained with semi-implicit SPH scheme and the reference solution is shown. A very good agreement between the two solutions is observed in Figure 7.1.

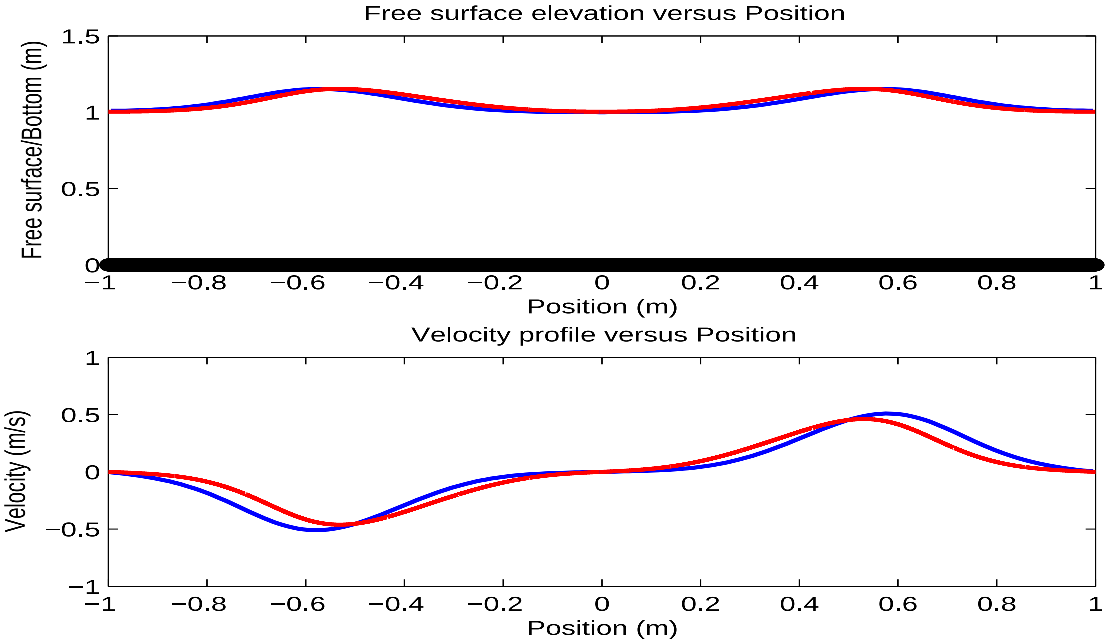


Figure 7.1: Smooth surface wave: Semi-implicit SPH scheme solution with 200 particles (solid line - blue) versus reference solution (solid line - red) - finite difference approach with a mesh of 10,000 points.

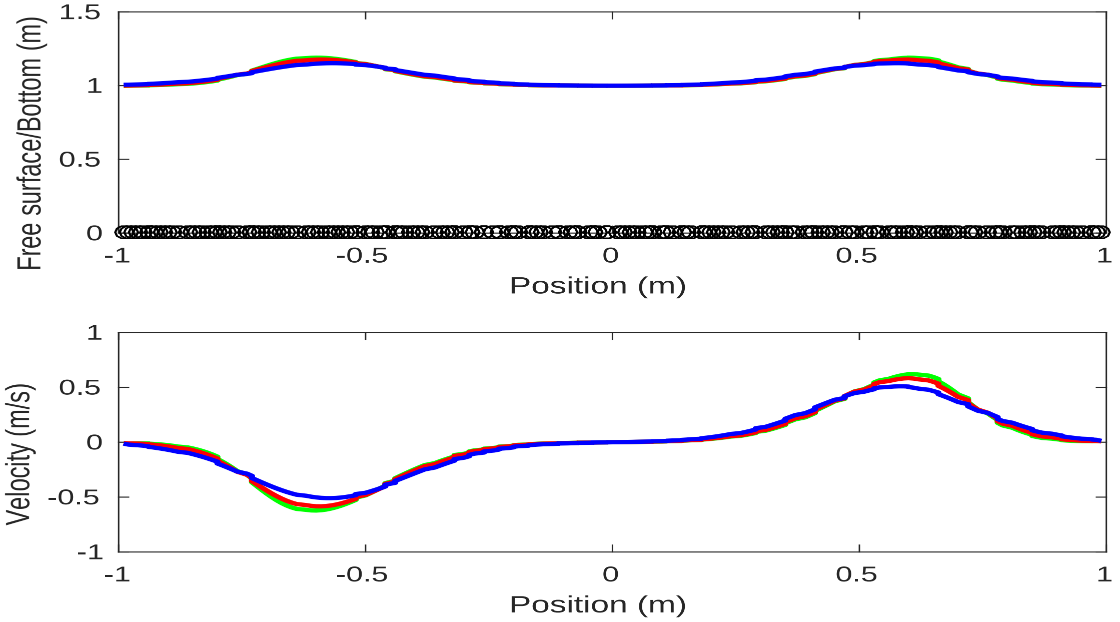


Figure 7.2: Smooth surface wave: Semi-implicit SPH solution, free surface elevation/bottom (upper profile), velocity (lower profile) computed at different time steps $\Delta t = 0.0025s$ (green), $\Delta t = 0.005s$ (red), $\Delta t = 0.01s$ (blue) at final time $t = 0.15s$.

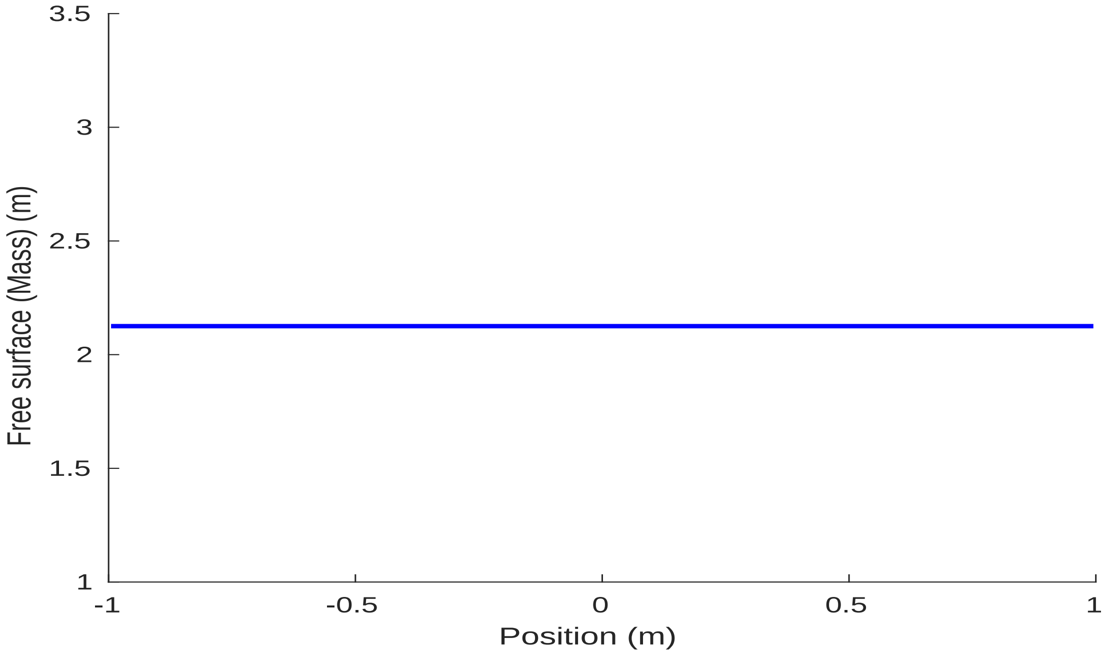


Figure 7.3: Smooth surface wave: Total mass conservation at time $t = 0.15s$.

7.1.2 Discontinuous Solution

In this example, we consider the following Riemann problem. Riemann problems are very important cases in initial value problem for PDE systems. The initial data is prescribed by two piecewise constant states often separated by a discontinuity:

$$\mathbf{q}(x, 0) = \begin{cases} \mathbf{q}_l & x < 0, \\ \mathbf{q}_r & x > 0 \end{cases} \tag{7.4}$$

where $\mathbf{q} = (v(x,0), \eta(x,0), h(x))$. The computational domain $\Omega = [x_l, x_r]$ given as $\Omega = [-1, 1]$ is discretized with the semi-implicit SPH scheme using 200 particles. In this example with flat bottom, the exact solution is given by the exact Riemann solver for the shallow water equations [107]. The left state $\mathbf{q}_l$ and the right state are given as $\mathbf{q}_r$

$$\mathbf{q}_l = \begin{pmatrix} -1 \\ 1 \\ 0 \end{pmatrix}, \qquad \mathbf{q}_r = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

In this present simulation, an implicitness factor $\Theta = 1$ is used, we simulated to the final time $t = 0.15$, with a fixed time step $\Delta t = 0.01$. We have used ghost particle boundary conditions in this example. The rarefaction solution of the one dimensional shallow water equation is presented in Figure 7.4, the solution consists of a left moving rarefaction fan and a right moving rarefaction fan solution both moving away from the discontinuity. To demonstrate that the method permits larger timesteps, we simulated for different times $\Delta t = 0.0025s, 0.005s, 0.01s$, a very good agreement can be seen in Figure 7.5. The tolerance for solving the linear system was set to $tol = 10^{-14}$. The total mass during this simulation is also conserved as we see in Figure 7.6, the total mass is obtained as 2.00000. We solve the example with a matrix-free implementation of the CG method, without making use of any preconditioner. We compare our semi-implicit SPH solution with the reference solution of the exact Riemann solver for the one dimensional shallow water equation. A very good agreement is observed in Figure 7.4. The upper profile in Figure 7.4 depicts the free surface elevation with a flat bottom bathymetry and the lower profile depicts a rarefaction particle velocity, respectively.

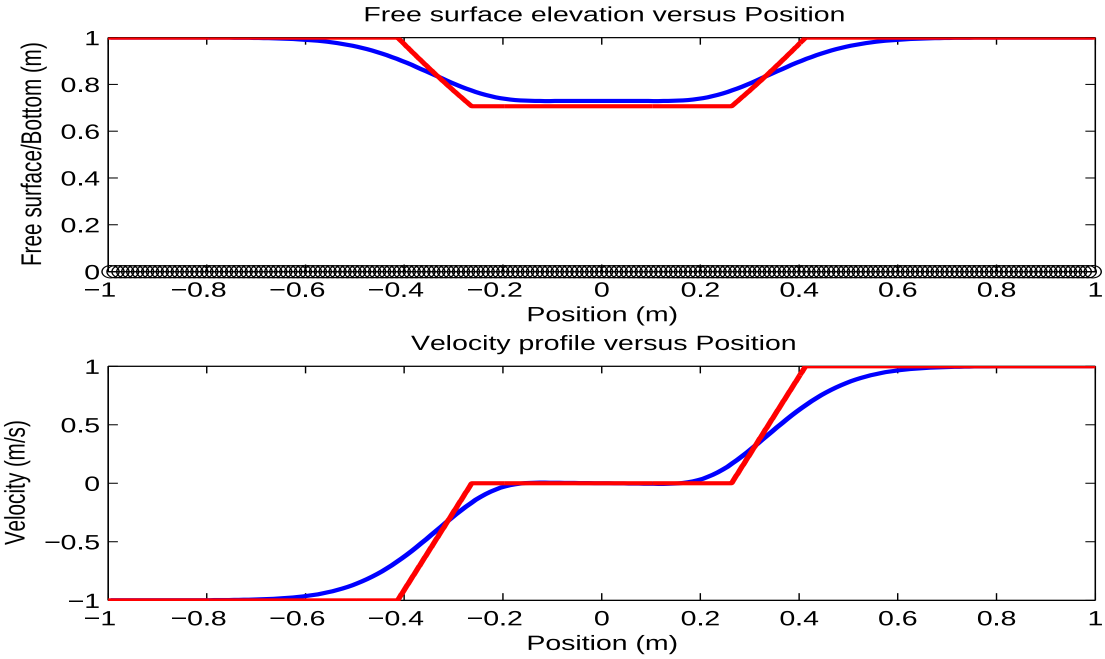


Figure 7.4: Discontinuous solution: Semi-implicit SPH scheme rarefaction solution (solid line - blue) versus exact solution (solid line - red). 200 particles is used in the numerical solution.

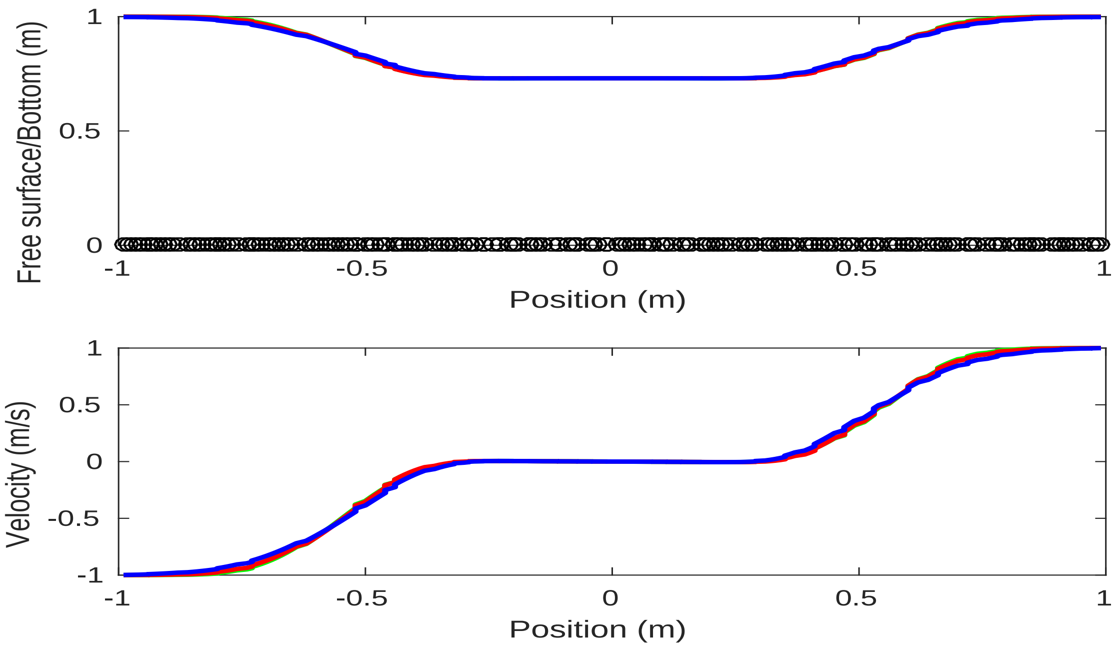


Figure 7.5: Discontinuous solution: Semi-implicit SPH solution, free surface elevation/bottom (upper profile), velocity (lower profile) computed at different time steps $\Delta t = 0.0025s$ (green), $\Delta t = 0.005s$ (red), $\Delta t = 0.01s$ (blue) at final time $t = 0.15s$.

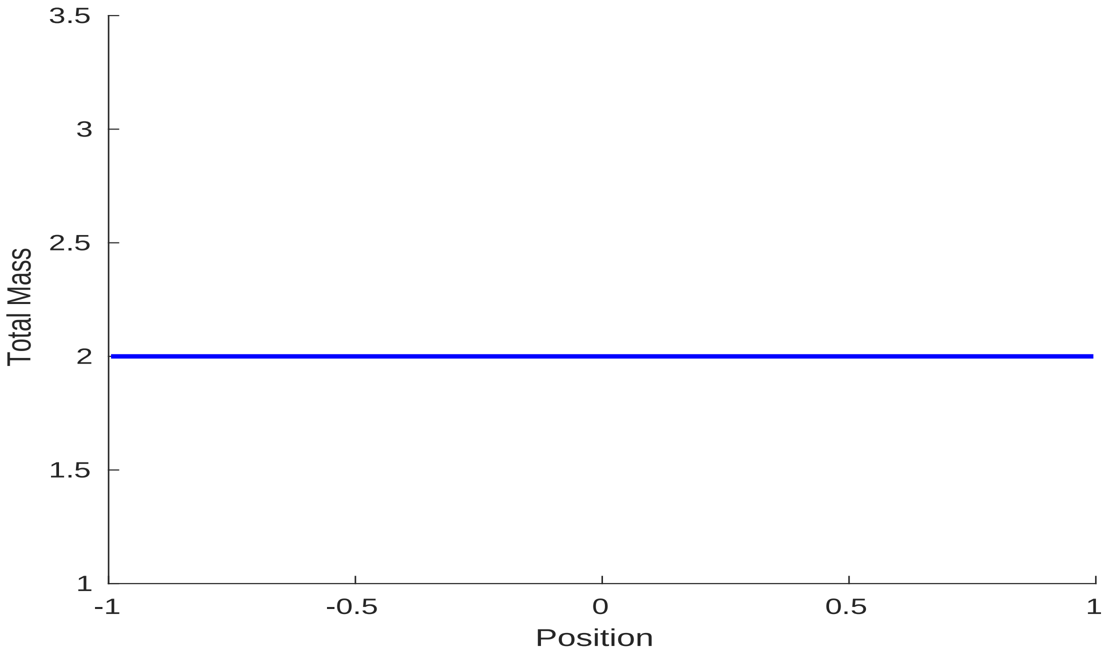


Figure 7.6: Discontinuous solution: Total mass conservation at time $t = 0.15s$.

7.1.3 An Oscillating Lake Problem

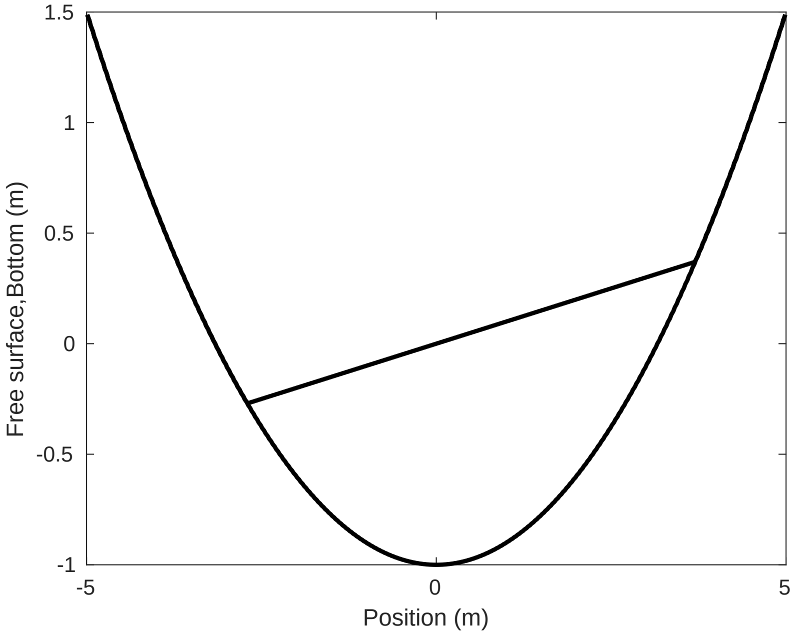


Figure 7.7: An oscillating lake

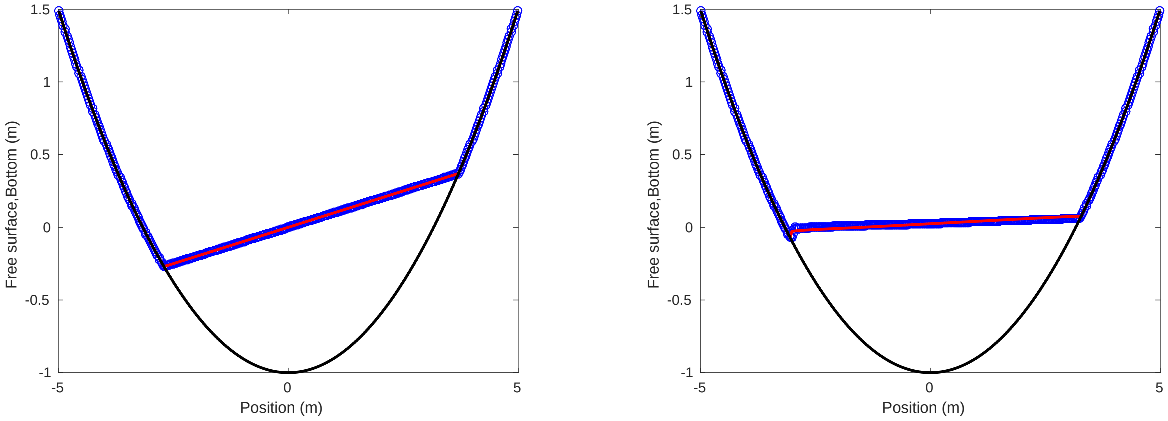
In this example, we consider fluid flow oscillating in a parabolic basin in Figure 7.7. The initial value problem in the domain $\Omega = [-5, 5]$ is given as:

$$\eta(x, 0) = 0.1x, \tag{7.5}$$

$$u(x, 0) = 0, \tag{7.6}$$

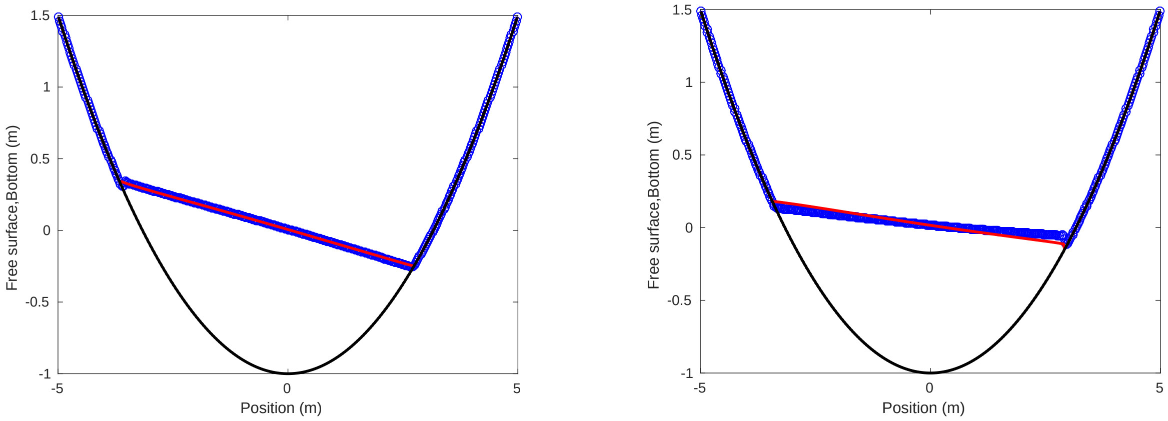
$$h(x) = 1 - 0.1x^2, \tag{7.7}$$

in a parabolic basin with bottom bathymetry h. In this present example, the flow is restricted below by a fixed bottom boundary h and bounded above by a moving free surface η, both free surface and bottom boundary are given by Equ. (7.5) and (7.7). The computational domain Ω is discretized with 400 particles. In our simulation, a variable smoothing length taken as $l_i = \alpha(\omega_i)^{\frac{1}{d}}$, where $\alpha \in [1.5, 2]$ and $d = 1$, implicitness factor $\Theta = 0.85$, a time step size of $\Delta t = 0.01$ and final time of simulation of $t = 7.2s$ is chosen. The plots of the evolution of the free surface inside the fixed parabolic basin is given at times $t = 0s$, $t = 1.8s$, $t = 3.6s$, $t = 5.4s$, and $t = 7.2s$ respectively in Figure 7.8. In this example, we solve a nonlinear system with the Newton method described in chapter 6, the tolerance for solving the nonlinear system was set to $tol = 10^{-14}$, the total mass is conserved as we see in Figure 7.9, the total mass is obtained to be 4.384533 at time $t = 7.2s$. We obtain a reference solution by solving the shallow water equation with a finite difference approach of Casulli and Cheng [29]. The comparison between our numerical results obtained with wetting and drying semi-implicit SPH and reference solution is shown. A very good agreement between the two solutions is observed in Figure 7.8 even at the transition regions.



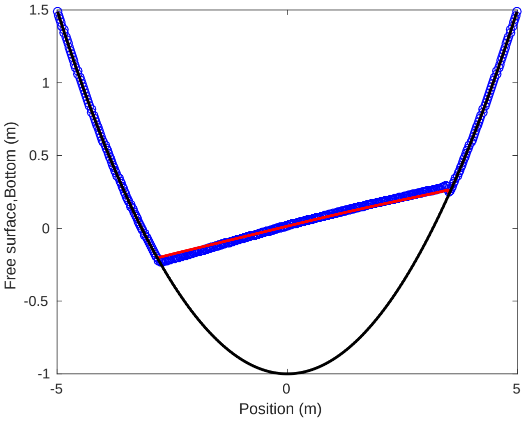
(a) Free surface at initial state t = 0s

(b) Free surface at time t = 1.8s



(c) Free surface at time t = 3.6s

(d) Free surface at time t = 5.4s



(e) Free surface at time t = 7.2s

Figure 7.8: Oscillating lake: Semi-implicit SPH wetting and drying solution for an oscillating lake problem at $t = 0.0s$, $t = 1.8s$, $t = 3.6s$, $t = 5.4s$, $t = 7.2s$ (blue dots) versus reference solution (red - solid line), the bottom bathymetry (black - solid line).

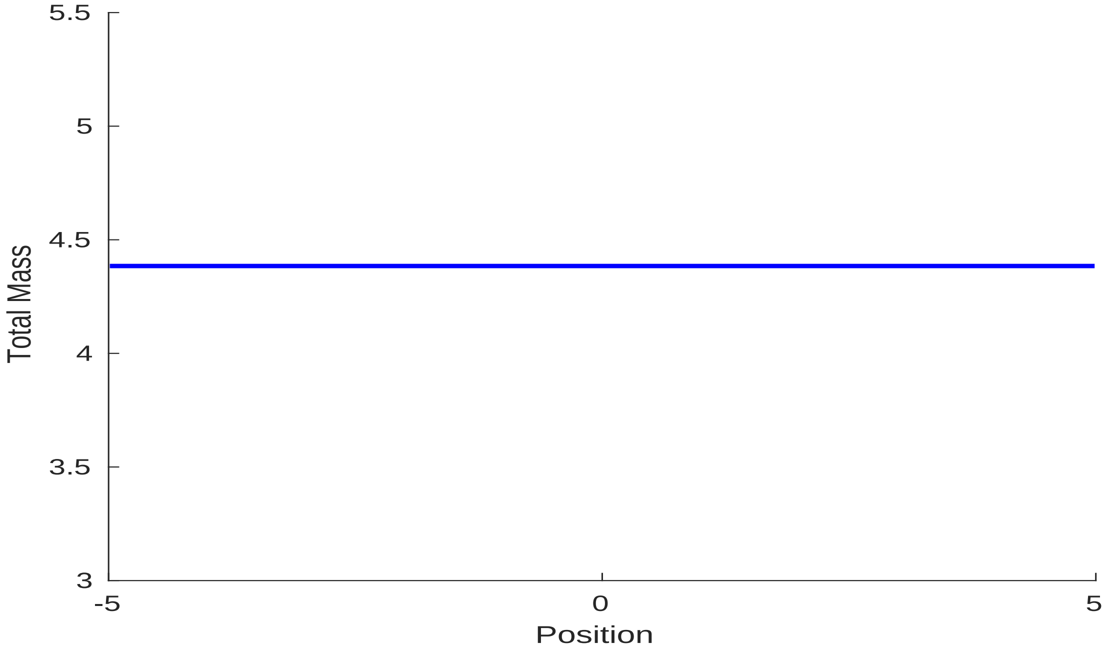


Figure 7.9: Oscillating lake example with wetting and drying: Total mass conservation at time $t = 7.2s$.

The 2D Shallow Water Equations

In this section, the semi-implicit SPH scheme will be validated on the two dimensional shallow water equation test problems. In this section, the following numerical examples will be validated namely: smooth surface wave propagation solution (collapsing Gaussian bump). In our subsequent test problems, the acceleration due to gravity constant g is also set to $g = 9.81$. In the numerical examples presented, we wish to mention that the particles are moved simply by $\frac{D\boldsymbol{r}_i}{Dt} = \boldsymbol{v}_i$. Our semi-implicit SPH solution will be benchmarked against reliable reference solutions.

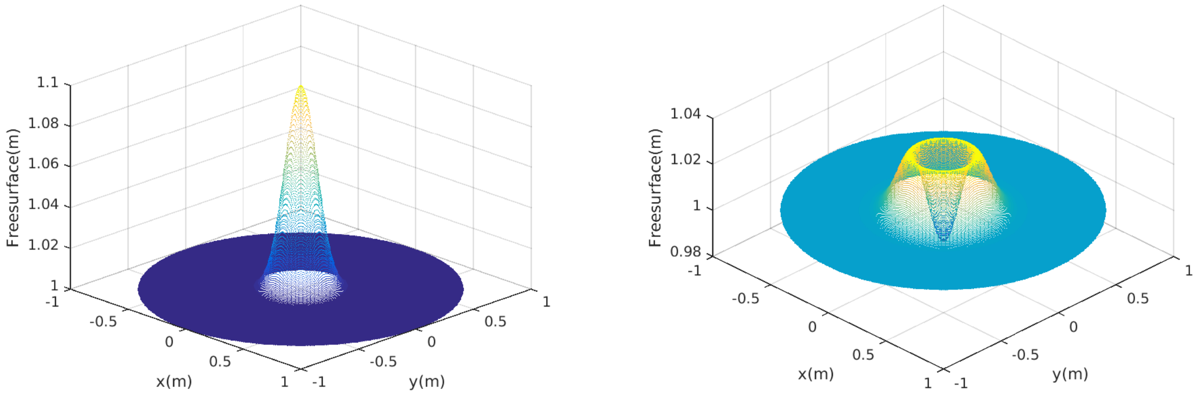
7.1.4 A Collapsing Gaussian Bump

In this example, we consider the following initial value problem in the domain $\Omega = [-1, 1] \times [-1, 1]$:

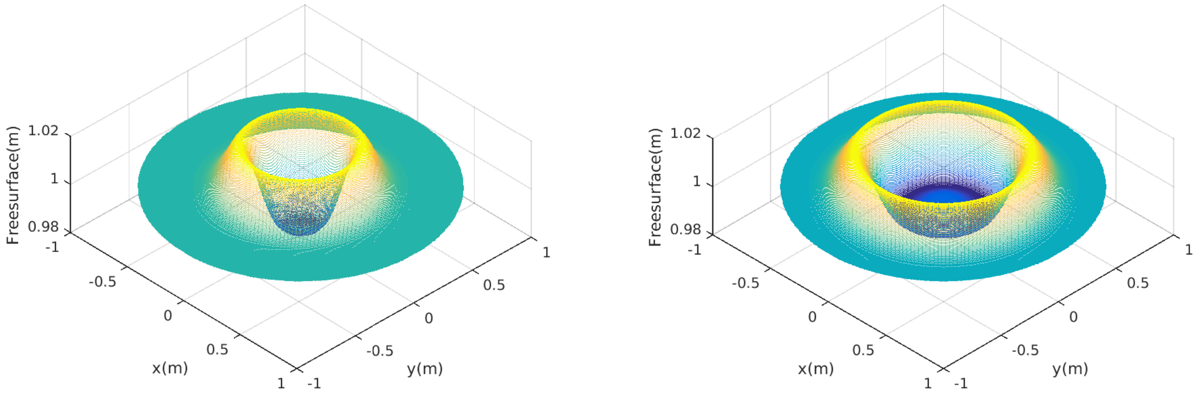
$$\eta(x, y, 0) = 1 + 0.1e^{-\frac{1}{2}\left(\frac{r^2}{\sigma^2}\right)},$$

$$u(x, y, 0) = v(x, y, 0) = h(x, y) = 0,$$

where $\sigma = 0.1$ and $r^2 = x^2 + y^2$ with flat bottom bathymetry, i.e., $h(x, y) = 0$, The computational domain Ω is discretized with $124, 980$ particles. The final simulation time is $t = 0.15$, and the time step is chosen to be $\Delta t = 0.0015$. We have used the implicitness factor $\Theta = 0.65$. The smoothing length is taken as $h_i = \alpha(\omega_i)^{1/d}$, where $\alpha = [1.5, 2]$ and $d = 2$. The obtained numerical solution is shown in Figure 7.12. The profiles in Figure 7.10 show the three dimensional surface plots of the free surface elevation at times $t = 0.0s, 0.05s, 0.10s, 0.15s$. Due to the radial symmetry of the problem, we obtain a reference solution by solving the one-dimensional shallow water equations with a geometric source term in radial direction: a method based on the high order classical shock capturing total variation diminishing (TVD) finite volume scheme is employed for computing the reference solution using $5,000$ points and the Osher-type flux for the Riemann solver, see [107] for details. The comparison between our numerical results obtained with semi-implicit SPH scheme and the reference solution is shown. The planar view and colour plots of the free surface elevation is given in Figure 7.11. A very good agreement between the two solutions is observed in Figure 7.12. We attribute the (rather small) differences in the plots to the fact that the SPH method has a larger effective stencil, which may increase the numerical viscosity. We have used a higher resolution of particle numbers of $195,496$, the cross section of the free surface elevation and the velocity at final time $t = 0.15s$ can be seen in Figure 7.13. We observe similar results compared to particle numbers 124,980.

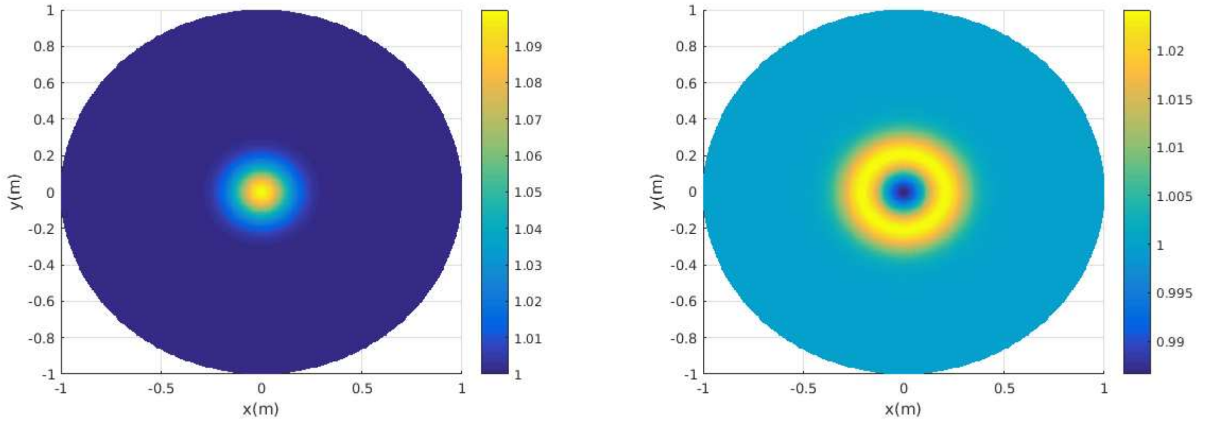


(a) Free surface at initial state t = 0s (b) Free surface at time t = 0.05s

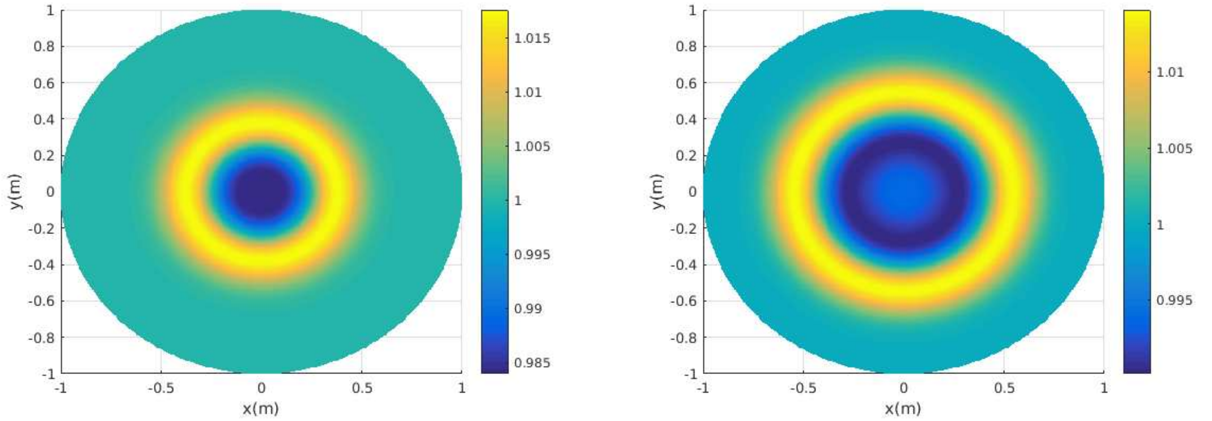


(c) Free surface at time t = 0.1s (d) Free surface at time t = 0.15s

Figure 7.10: Collapsing Gaussian bump: Three-dimensional view of the smooth surface wave propagation example: The free surface evolution Semi-implicit SPH scheme solution at times $t = 0s$, $t = 0.05s$, $t = 0.1s$ and $t = 0.15s$. A total of $124,980$ particles is used in the numerical solution.

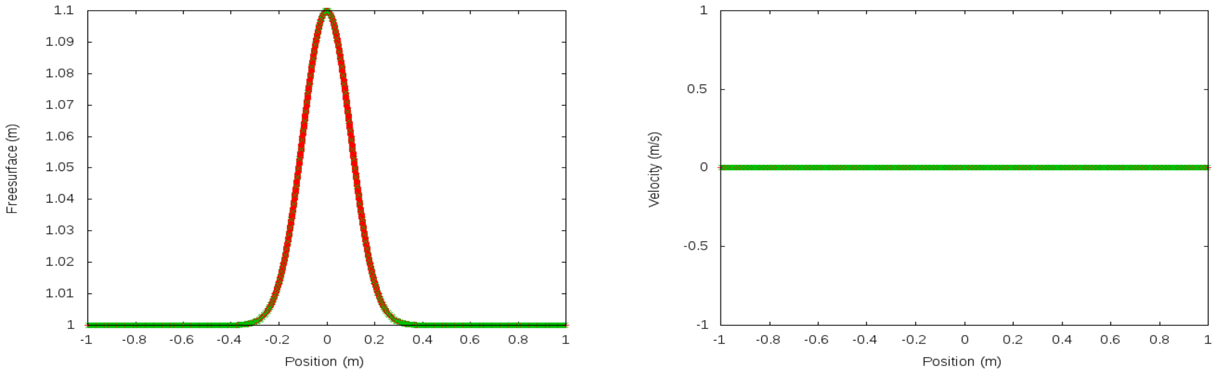


(a) Planar view at initial state t = 0s (b) Planar view at time t = 0.05s

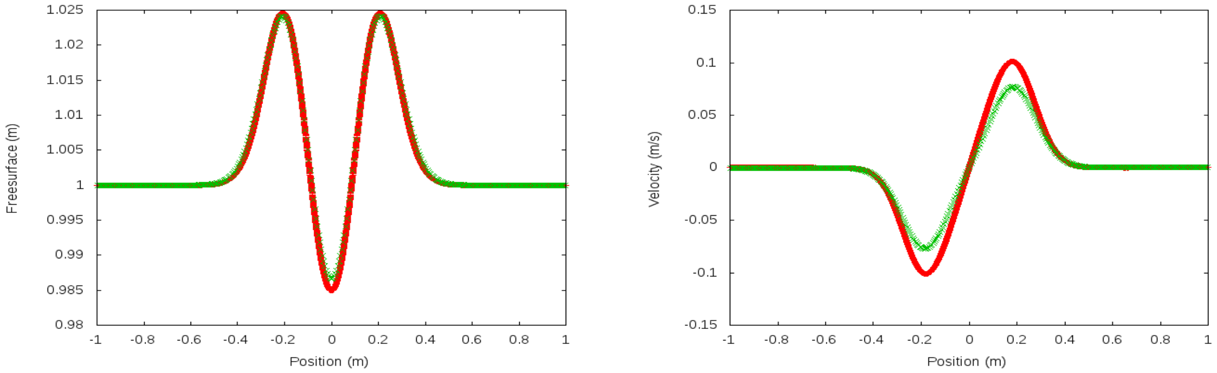


(c) Planar view at time t = 0.1s (d) Planar view at time t = 0.15s

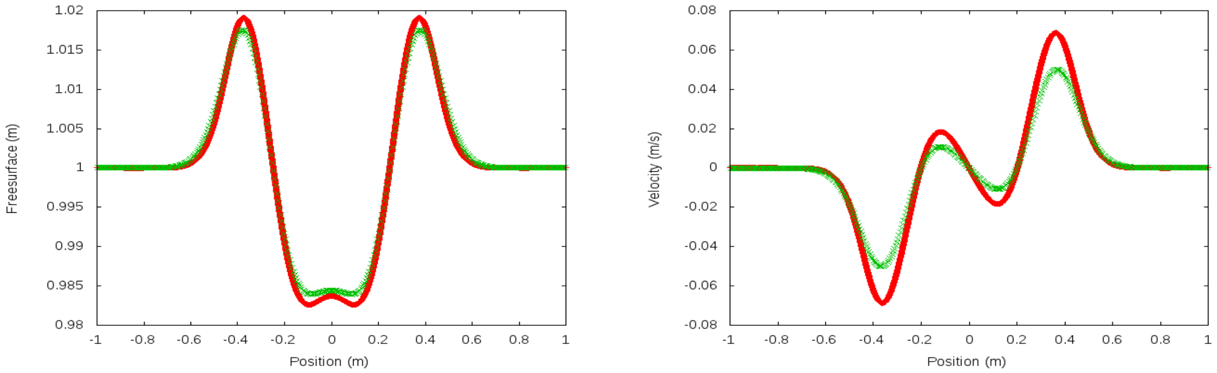
Figure 7.11: Collapsing Gaussian bump: Planar view of the smooth surface wave propagation example: The free surface evolution semi-implicit SPH scheme solution at times $t = 0s$, $t = 0.05s$, $t = 0.1s$ and $t = 0.15s$. A total of $124,980$ particles is used in the numerical solution.



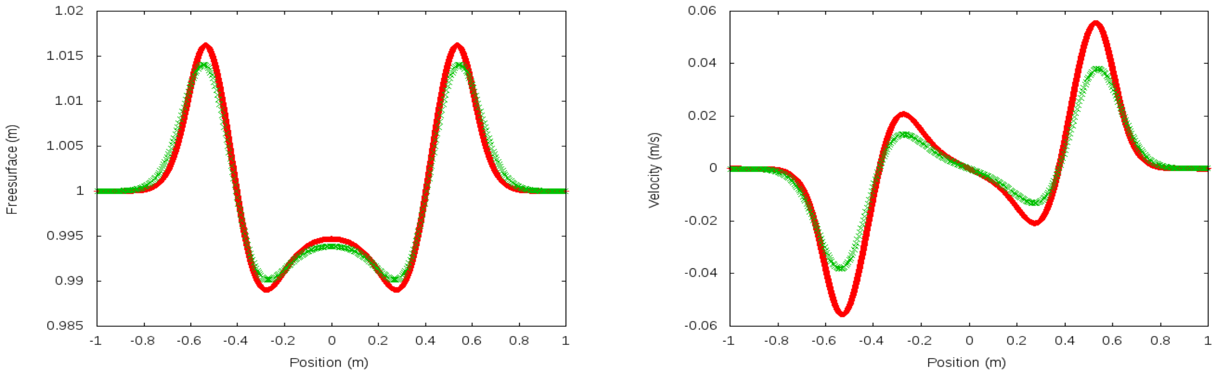
(a) Free surface at initial state t = 0s (b) Velocity at time t = 0s



(c) Free surface at time t = 0.05s (d) Velocity at time t = 0.05s



(e) Free surface at initial state t = 0.10s (f) Velocity at time t = 0.10s



(g) Free surface at time t = 0.15s (h) Velocity at time t = 0.15s

Figure 7.12: Collapsing Gaussian bump: Cross section of semi-implicit SPH solution (green) versus reference solution (red): Free-surface (left), velocity (right) in the $x-$ direction.

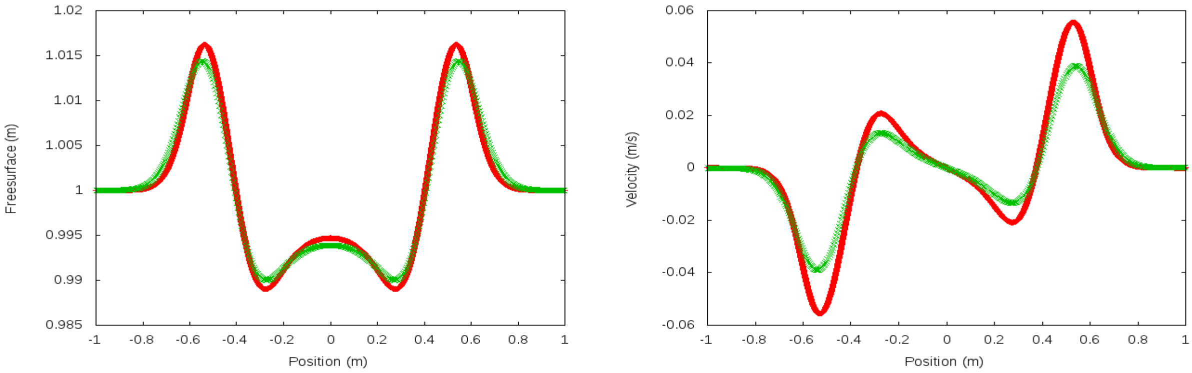


Figure 7.13: Collapsing Gaussian bump: Cross section of semi-implicit SPH solution (green) versus reference solution (red): Free-surface (left), velocity (right) in the x-direction at times $t = 0.15s$ with a higher resolution of 195,496 particles.

8. Conclusions and Outlook

8.1 Conclusion

This thesis deals with the design, development and analysis of a new semi-implicit Smoothed Particle Hydrodynamics method (SISPH), a meshfree method designed for the numerical treatment of fluid flows governed by conservation laws which in particular we deal with the shallow water equations (SWEs). The need to develop new numerical methods can not be overemphasized because one can refine existing numerical methods and improve them depending on the inherent drawbacks that accrue to them. Therefore, it is advantageous to develop a new SISPH method which is relatively simple, unconditionally stable, efficient, flexible and guarantees mass (volume) conservation. The summary of the thesis is given below.

In Chapter 2, we introduced basic concepts in the mathematical modeling of fluid flows, the conservation of mass and momentum, we presented the key ideas in the theory of hyperbolic conservation laws which the SWEs is a family of, and we give some matrix properties and definitions. In Chapter 3, we provided the theory of the two-dimensional SWEs in particular the depth-averaged version, we derived the SWEs from the Navier-Stokes in primitive variables. Because, the SISPH method developed in this thesis is formulated in primitive variables, we proceed by giving the characteristics and derived the eigenstructure of the SWEs in terms of the physical variables (primitive variables) in order to show that the eigenstructure do not differ from the case of using conservative variables. In Chapter 4, we gave the fundamental theory of the SPH method, we explained the consistency issues inherent in the SPH method. To this effect, we proceed by giving correction techniques that restore consistency of the SPH method at the particle level.

In Chapter 5, we have proposed a meshfree semi-implicit smoothed particle hydrodynamics (SISPH) method for the shallow water equations in one and two space dimensions, the SPH formulation is based on Ben Moussa and Vila. In our scheme, the momentum equation is discretized by a finite difference approximation for the gradient of the free surface and the SPH approximation for the mass conservation equation. The velocity and the free surface elevation have been defined at staggered positions, an artificial velocity was defined and a smoothing kernel interpolation has been used to interpolate between particle velocities and the staggered velocities: a technique analogous to the XSPH method of Monaghan. By substitution of the discrete momentum equations into the discrete mass conservation equations, we arrive at a single equation in one unknown this leads to a sparse linear system for the free surface elevation, a very unique feature that makes our method stands out from other semi-implicit approaches, this makes our method very flexible and robust. We solve this system efficiently by an iterative matrix-free version of the conjugate gradient (CG) algorithm.

The key features of the proposed semi-implicit SPH method are briefly as follows: The method is mass conservative; efficient; time steps are not restricted by a stability condition (coupled to the surface wave speed), thus large time steps are permitted. In Chapter 7, we presented standard model problems in one and two dimensions for the shallow water equations that validates the above mentioned features.

In Chapter 6, we propose a new wetting and drying semi-implicit SPH algorithm that is based on the novel semi-implicit SPH discretization. The semi-implicit SPH algorithm with wetting and drying applied to the shallow water equations has been derived and discussed, this involves solving a nonlinear system and in particular a problem with nonflat bottom bathymetry. The momentum equation is discretized by a finite difference approximation for the gradient of the free surface elevation and SPH approximation for the mass conservation equation. Because we substituted the discrete momentum equations into the discrete mass conservation equations and because we define the water particle volume as a piecewise constant function, we arrive at a mildly nonlinear sparse system for the free surface elevation. We thereby solve some Newton-type iterations when wetting and drying is encountered. We conveniently solve this mildly nonlinear system with the matrix-free version of the conjugate gradient (CG) algorithm. Our semi implicit algorithm automatically accommodates the treatment of wetting and drying. In all the test cases in this thesis, we have solved them with a matrix-free implementation of the CG method, without the usage of a preconditioner. A very good sparsity structure of the system for the free surface, leads to a computationally efficient scheme.

The key features of the proposed wetting and drying algorithm are as follows. The method does not involve putting screens at velocity points of the flow configuration when the water height drops below a certain drying threshold and removing the screens when the water height rises above a wetting threshold, the method achieves a correct mass balance in wet regions and in transition regions i.e., the regions from wet particles to dry particles and those from dry particles to wet particles. In our nonlinear wetting and drying algorithm, there is no thin water film or a small tolerance to deal with dry particles, the method is designed to perfectly treat dry particles with zero particle volume. The wetting and drying algorithm is simple and efficient, and it guarantees the production of non-negative water depths. Finally, the method's time step is not restricted by stability conditions that are dictated by the surface wave speed, thereby allowing large time steps. An oscillating lake example in Chapter 7 was presented that emphasizes and validates the above mentioned features for wetting and drying.

8.2 Outlook

Since the SISPH method in this thesis is still in its developmental stage, the proposed method has its weaknesses which can be improved. Therefore, there exists more research topics to be investigated:

Application to shock problems; the SISPH method presented in this thesis is done in primitive variables (nonconservative formulation), primitive variables are set of variables such that the mathematical and computational manipulation of the conservation laws become more easy and convenient. In most problems involving solutions to hyperbolic conservation laws whose solutions may develop discontinuities, shock solutions are mostly inevitable. In the past decades, it is an accepted practice to utilize conservative methods. Shock waves are the solution features that demand conservative methods. Computational experience proves that the use of a nonconservative method results in the wrong shock strength and consequently the wrong propagation speed. We note however that simulating problems involving shocks will give solutions converging to the wrong solution. Hence, there are very good reasons this aspect is worth looking into for consideration by recasting the formulation into a conservative form. Once the conservative formulation is done, we believe it will be possible to simulate the shallow water equations not only with fixed bed but with a mobile bed.

One area of further research could concern the extension of the scheme to high order of accuracy in time. Also since, the choice of discretization of the convective terms in the momentum equation affects the accuracy of the method. There is the need to improve the accuracy of the scheme by investigating on a high order velocity field reconstruction, this can be achieved by reconstructing high order velocity field in a semi-Lagrangian approach to discretize the nonlinear convective terms.

In the same spirit, as concerning solving the incompressible Navier-Stokes equations, further investigations is needed so as to see how our SISPH method behaves and with this we can compare solutions with the incompressible SPH (ISPH) technique.

Extension of the scheme to the fully three-dimensional case; the SISPH scheme developed in this thesis is flexible and robust so it can easily be extended to higher dimensional fluid flow configuration which includes application of the scheme to real life river flooding and drying scenarios, i.e., lakes, estuaries.

Bibliography

[1] R. Aris. *Vectors, Tensors, and the Basic Equations of Fluid Mechanics*. Dover Publications INC., New York, 1962.

[2] M. Arroyo and M. Ortiz. Local Maximum-entropy Approximation Schemes: A Seamless Bridge between Finite Element and Meshfree Methods. *International Journal for Numerical Methods in Engineering, 65: 2167–2202*, 2006.

[3] B. Ataie-Ashtiani and L. Farhadi. A Stable Moving-particle Semi-implicit Method for Free Surface Flows. *Fluid Dynamics Research, 38: 241–256*, 2006.

[4] B. Ataie-Ashtiani, G. Shobeyri, and L. Farhadi. Modified Incompressible SPH Method for Simulating Free Surface Problems. *Fluid Dynamics Research, 40: 637–661*, 2008.

[5] I. Babuska and J.M. Melenk. The Partition of Unity Finite Element Method. Technical report, Technical Report Technical Note BN-1185, Institute for Physical Science and Technology, University of Maryland, April, 1995.

[6] A. Balzano. Evaluation of Methods for Numerical Simulation of Wetting and Drying in Shallow Water Flow Models. *Coastal Engineering, Vol. 34, pages 83-107*, 1998.

[7] A.O. Bankole, M. Dumbser, A. Iske, and T. Rung. A Semi-implicit SPH Scheme for the Shallow Water Equations. *Proceedings of the 9th SPHERIC International Workshop, Paris, France, pages. 419–424*, 2014.

[8] A.O. Bankole, M. Dumbser, A. Iske, and T. Rung. A Semi-implicit SPH Scheme for the Two-dimensional Shallow Water Equations. *Proceedings of the 10th SPHERIC International Workshop, Parma, Italy, pages. 252–258*, 2015.

[9] A.O. Bankole, M. Dumbser, A. Iske, and T. Rung. A Wetting and Drying Semi-implicit SPH Algorithm for the Shallow Water Equations. *Proceedings of the 11th SPHERIC International Workshop, Munich, Germany, pages. 260–267*, 2016.

[10] A.O. Bankole, M. Dumbser, A. Iske, and T. Rung. A Meshfree Semi-implicit Smoothed Particle Hydrodynamics Method for Free Surface Flow. *To appear in Lecture Notes in Computational Science and Engineering: Meshfree Methods for Partial Differential Equations VIII, Springer-Verlag, Editors: M. Griebel and M.A. Schweitzer*, 2016.

[11] M. Basa, N. Quinlan, and M. Lastiwka. Robustness and Accuracy of SPH Formulations for Viscous Flow. *International Journal for Numerical Methods in Fluids, 60: 1127–1148*, 2009.

[12] T. Belytschko, Y. Krongauz, J. Dolbow, and C. Gerlach. On the Completeness of Meshfree Particle Methods. *Int. J. Numer. Methods Eng, 43:785-819*, 1998.

[13] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, and P. Krysl. Meshless Methods: an overview and recent developments. *Computer Methods in Applied Mechanics and Engineering Vol. 139, pp. 3-47*, 1996.

[14] T. Belytschko, Y.Y. Lu, and L. Gu. Element-free Galerkin Methods. *Int. J. Numer. Methods Eng, 37:229-256*, 1994b.

[15] T. Belytschko, Y.Y. Lu, and L. Gu. Crack Propagation by Element-free Galerkin Methods. *Engineering Fracture Mechanics, Vol 51, pp. 295-315*, 1995.

[16] W. Benz. Smoothed Particle Hydrodynamics: A Review. In *In Numerical Modeling of Non-linear Stellar Pulsation: Problems and Prospects, kluwer Academic, Boston.*, 1990.

[17] O. Bokhove. Flooding and Drying in Discontinuous Galerkin Finite-Element Discretizations of Shallow Water Equations. Part 1: One Dimension. *Journal of Scientific Computing, Vol. 23*, 2005.

[18] L. Bonaventura, A. Iske, and E. Miglio. Kernel-based Vector Field Reconstruction in Computational Fluid Dynamic Models. *International Journal for Numerical Methods in Fluids. Vol. 66, pages. 714-729*, 2011.

[19] J. Bonet and T. Lok. Variational and Momentum Preservation Aspects of Smoothed Particle Hydrodynamic Formulations. *Computer Methods in Applied Mechanics and Engineering Vol. 180, (1-2) pp. 97-115*, 1999.

[20] S.C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods*. Springer-Verlag, New York, 2002.

[21] L. Brookshaw. A Method of Calculating Radiative Heat Diffusion in Particle Simulations. In *Astronomical Society of Australia, Proceedings, Vol. 6, (ISSN: 0066-9997), pp. 207-210*, 1985.

[22] L. Brugnano and V. Casulli. Iterative Solution of Piecewise Linear Systems. *SIAM Journal on Scientific Computing, Vol. 30. pages 463-472*, 2008.

[23] L. Brugnano and V. Casulli. Iterative Solution of Piecewise Linear Systems and Applications to Flows in Porous Media. *SIAM Journal on Scientific Computing, Vol. 31, No. 3. pages 1858-1873*, 2009.

[24] S. Bunya, E.J. Kubatko, J.J. Westerink, and C. Dawson. A Wetting and Drying Treatment for the Runge-Kutta Discontinuous Galerkin Solution to the Shallow Water Equations. *Computer Methods in Applied Mechanics and Engineering. Vol. 198, pages 1548-1562*, 2009.

[25] V. Casulli. Semi-implicit Finite Difference Methods for the two-dimensional Shallow Water Equations. *Journal of Computational Physics*, 86:56–74, 1990.

[26] V. Casulli. A High-resolution Wetting and Drying Algorithm for Free Surface Hydrodynamics. *International Journal for Numerical Methods in Fluids, Vol. 60, pages 391-408*, 2009.

[27] V. Casulli. *Advanced Numerical Methods for Free-surface Hydrodynamics*. 2014.

[28] V. Casulli and E. Cattani. Stability, Accuracy and Efficiency of a Semi-implicit Method for Three-dimensional Shallow Water Flow. *Computers and Mathematics with Applications. Vol 27. pp. 99-112*, 1994.

[29] V. Casulli and R.T. Cheng. Semi-implicit Finite Difference Methods for the Three-dimensional Shallow Water Flows. *International Journal for Numerical Methods in Fluids. Vol. 15. pages: 629-648*, 1992.

[30] V. Casulli and R.A. Walters. An Unstructured Grid, Three-dimensional Model Based on the Shallow Water Equations. *International Journal for Numerical Methods in Fluids. Vol. 32 pages 331-348*, 2000.

[31] V. Casulli and P. Zanolli. A Nested Newton-type Algorithm for Finite Volume Methods Solving Richards' Equation in Mixed Form. *SIAM Journal on Scientific Computing, Vol. 32, No. 4, pages 2255-2273*, 2010.

[32] V. Casulli and P. Zanolli. Iterative Solutions of Mildly Nonlinear Systems. *Journal of Computational and Applied Mathematics. 236. pages 3937-3947*, 2012.

[33] L. Cea, J. Puertas, and M.E. Vazquez-Cendon. Depth Averaged Modelling of Turbulent Shallow Water Flow with Wet-dry Fronts. *Archives of Computational Methods in Engineering. Vol. 11, pages 1-50*, 2006.

[34] A. Chaniotis, D. Poulikakos, and P. Kououtsakos. Remeshed Smoothed Particle Hydrodynamics for the Simulation of Viscous and Heat Conducting Flows. *Journal of Computational Physics 182, pages 67-90*, 2002.

[35] J.K. Chen and J.E.Beraun. A Generalised Smoothed Particle Hydrodynamics Method for Nonlinear Dynamic Problems. *Computer Methods in Applied Mechanics and Engineering Vol. 190, pp. 225-239*, 2000.

[36] J.S. Chen, C. Pan, C.-T. Wu, and W.K. Liu. Reproducing Kernel Particle Methods for Large Deformation Analysis of Non-linear Structures. *Computer Methods in Applied Mechanics and Engineering Vol. 139, pp. 195-228*, 1996.

[37] A.J. Chorin and J.E. Marsden. *A Mathematical Introduction to Fluid Dynamics*. Springer-Verlag. ISBN 3-540-90406-9, 1979.

[38] P.G. Ciarlet and J.L. Lions (Eds.). *Finite Element Methods (Part 1). Handbook of Numerical Analysis*. Vol. 2, North-Holland, Amsterdam, 1991.

[39] W.S. Cleveland. Visualizing Data. Technical report, AT&T Bell Laboratories, Murray Hill, NJ, 1993.

[40] C. Dafermos. *Hyperbolic Conservation Laws in Continuum Physics*. Springer, Heidelberg, 2000.

[41] C.A. Duarte and J.T. Oden. An hp Adaptive Method using Clouds. *Computer Methods in Applied Mechanics and Engineering Vol. 139, pp. 237-262*, 1996.

[42] M. Dumbser and V. Casulli. A Staggered Semi-implicit Spectral Discontinuous Galerkin Scheme for the Shallow Water Equations. *Applied Mathematics and Computation, Vol. 219 pages 8057-8077*, 2013.

[43] R. Eymard, T. Gallouet, and R. Herbin. *Finite Volume Methods. Handbook of Numerical Analysis*. Vol. VII, 713-1018, 2000.

[44] R. Fatehi, M. Fayazbakhsh, and M.T. Manzari. On Discretization of Second-Order Derivatives in Smoothed Particle Hydrodynamics. *International Journal of Natural and Applied Sciences 3 (1) 50-53*, 2009.

[45] R. Fatehi and M.T. Manzari. Error Estimation in Smoothed Particle Hydrodynamics and a New Scheme for Second Derivatives. *Computers and Mathematics with Applications. Vol 61. pp. 482-498*, 2011.

[46] A. Ferarri, M. Dumbser, E.F. Toro, and A. Armanini. A New 3D Parallel SPH Scheme for Free Surface Flows. *Computers and Fluid 38(6), pages 1203-1217*, 2009.

[47] O. Flebbe, S. Munzel, H. Herold, H. Riffert, and H. Ruder. Smoothed Particle Hydrodynamics - Physical Viscosity and the Simulation of Accretion Disk. *The Astrophysical Journal (431) 214-226*, 1994.

[48] R.A. Gingold and J.J. Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society, 181:375-389*, 1977.

[49] E. Godlewski and P.A. Raviart. *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer-Verlag, 1996.

[50] G.H. Golub and C.F. Van Loan. *Matrix Computations*. The John Hopkins University Press, 2013.

[51] M. Griebel and M.A. Schweitzer. A Particle-Partition of Unity Method for the Solution of Elliptic, Parabolic, and Hyperbolic PDES. *SIAM Journal of Scientific Computing, Vol. 22, No. 3, pp. 853-890*, 2000.

[52] A. Gunatilaka. Some Aspects of the Biology and Sedimentology of Laminated Algalmats from Mannar Lagoon. *Northwest Ceylon. Sedimentary Geology 14(4) pages 275-300*, 1975.

[53] L. Hernquist and N. Katz. Treesph - A Unification of SPH with the Hierarchical Tree Method. *Astronomical Journal, 70:419-446*, 1989.

[54] M.R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *J. Res. Nat. Bur. Stand. 49, pages 409-436*, 1952.

[55] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 2013.

[56] A. Iske and M. Käser. Conservative Semi-Lagrangian Advection on Adaptive Unstructured Meshes. *Numerical Methods for Partial Differential Equations. (20)388-411*, 2004.

[57] E.T Jaynes. Information Theory and Statistical Mechanics. *Physical Review, 106, 620*, 1957.

[58] J. Jeong, M. Jhon, J.Halow, and J. van Osdol. Smoothed Particle Hydrodynamics: Applications to Heat Conduction. *Computer Physics Communications 153 (1) 71-84*, 2003.

[59] Y.W. Jiang and W.H. OnyxWai. Drying-wetting Approach for 3D Finite Element Sigma Coordinate Model for Estuaries with Large Tidal Flats. *Advances in Water Resources. Vol. 28, pages 779-792*, 2005.

[60] G.R. Johnson and S.R. Beissel. Normalized Smoothing Functions for SPH Impact Calculations. *International Journal for Numerical Methods in Engineering, 39: 2725–2741*, 1996.

[61] G. Kesserwani and Q. Liang. Well-balanced RKDG2 Solutions to the Shallow Water Equations over Irregular Domains with Wetting and Drying. *Computers and Fluids. Vol. 39. pages 2040-2050*, 2010.

[62] A. Khayyer and H. Gotoh. Modified Moving Particle Semi-implicit Methods for the Prediction of 2d Wave Impact Pressure. *Coastal Engineering, 56: 419–440*, 2009.

[63] A. Khayyer and H. Gotoh. Enhancement of Stability and Accuracy of the Moving Particle Semi-implicit Method. *Journal of Computational Physics, Vol 230, pages. 3093–311*, 2011.

[64] A. Khayyer and H. Gotoh. Enhancement of Performance and Stability of Mps Mesh-free Particle Method for Multiphase Flows Characterized by High Density Ratios. *Journal of Computational Physics, Vol 242, pages, 211–233*, 2013.

[65] A. Khayyer, H. Gotoh, and S.D. Shao. Corrected Incompressible SPH Method for Accurate Water-surface Tracking in Breaking Waves. *Coastal Engineering, 55: 236–250*, 2008.

[66] S. Koshizuka, A. Nobe, and Y. Oka. Numerical Analysis of Breaking Waves using the Moving Particle Semi-implicit Method. *Intenational Journal for Numerical Methods in Fluids, 26: 751–769*, 1998.

[67] S. Koshizuka and Y. Oka. Moving-Particle Semi-implicit Method for Fragmentation of Incompressible Fluid. *Nuclear Science and Engineering, 123: pages, 421–434*, 1996.

[68] Y. Krongauz and T. Belytschko. Consistent Pseudo-derivatives in Meshless Methods. *Computer Methods in Applied Mechanics and Engineering Vol. 146, pp. 371-386*, 1997a.

[69] D. Kröner. *Numerical Schemes for Conservation Laws*. Wiley-Teubner, 1997.

[70] P. Lancaster and K. Salkauskas. Surfaces Generated by Moving Least Squares Methods. *Math. Comput., 37, 141-158*, 1981.

[71] P.D. Lax. Weak Solutions of Nonlinear Hyperbolic Equations and Their Numerical Computations. *Comm. Pure Appl. Math. VII, 159-193*, 1954.

[72] J.J. Leendertse. A Water-quality Simulation Model for Well-mixed Estuaries and Coastal Seas. Principles of Computation. *Rand, Santa Monica. 1:RM-6230-RC*, 1970.

[73] M. Lentine, J.T. Gretarsson, and R. Fedkiw. An Unconditionally Stable Fully Conservative Semi-Lagrangian Method. *Journal of Computational Physics, Vol 230. pages. 2857-2879*, 2011.

[74] R.J. LeVeque. *Numerical Methods for Conservation Laws*. Birkhäuser Verlag, 1992.

[75] R.J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2002.

[76] S. Li, W. Hao, and W.K. Liu. Numerical Simulations of Large Deformations of Thin Shell Structures using Meshfree Methods. *Computational Mechanics, Vol. 25 No.2/3, pages 102-116*, 2000.

[77] S. Li and W.K. Liu. *Meshfree Particle Methods*. Springer-Verlag Berlin Heidelberg, 2007.

[78] S. Li, Q. Qian, W.K. Liu, and T. Belytschko. A Meshfree Contact Detection Algorithm. *Computer Methods in Applied Mechanics and Engineering Vol. 190, pp. 7185-7206*, 2000.

[79] L.D Libersky and A.G. Petscheck. Smoothed Particle Hydrodynamics with Strengths of Materials. In *in H. Trease, J. Fritts and W. Crowley (ed.): Proceedings of the Next Free Lagrange Conference, Springer-Verlag, NY, 395, pages 248-257*, 1991.

[80] G.R. Liu and M.B. Liu. *Smoothed Particle Hydrodynamics - a meshfree particle method*. World Scientific Publishing Co. Pte. Ltd., Singapore, 2003.

[81] M.B. Liu and G.R Liu. Restoring Particle Consistency in Smoothed Particle Hydrodynamics. *Applied Numerical Mathematics Vol. 56, pp 19-36*, 2006.

[82] W.K. Liu, S. Jun, and Y. Zhang. Reproducing Kernel Particle Methods. *International Journal for Numerical Methods in Fluids, 20: 1081–1106*, 1995.

[83] L.B. Lucy. A Numerical Approach to the Testing of the Fission Hypothesis. *Astronomical Journal, 82:1013-1024*, 1977.

[84] S. Manenti, S. Sibilla, M. Gallati, G. Agate, and R. Guandalini. SPH Simulation of Sediment Flushing Induced by a Rapid Water Flow. *Journal of Hydraulic Engineering, 138(3):272-284*, 2012.

[85] S. Mas-Gallic and R. Raviart. A Particle Method for First-order Symmetric Systems. *Numer. Math. 51 pp. 323-352*, 1987.

[86] J.J. Monaghan. An Introduction to SPH. *Computer Physics Communications Vol. 48, pages 89-96*, 1988.

[87] J.J Monaghan. On the Problem of Penetration in Particle Methods. *Journal of Computational Physics 82(1), pages 1-15*, 1989.

[88] J.J. Monaghan. Smoothed Particle Hydrodynamics. *Annual Review of Astronomical and Astrophysics, 30:543-574*, 1992.

[89] J.J. Monaghan. Simulating Free Surface Flows with SPH. *Journal of Computational Physics, 110:399-406*, 1994.

[90] J.J Monaghan. Smoothed Particle Hydrodynamics. *Reports on Progress in Physics Vol. 68, pages 1703-1759*, 2005.

[91] J.J. Monaghan and J.C. Lattanzio. A Refined Particle Method for Astrophysical Problems. *Astron. Astrophys. 149*, 1985.

[92] J. P. Morris. *Analysis of Smoothed Particle Hydrodynamics with Applications*. PhD thesis, Monash University, 1996.

[93] B. Ben Moussa and J.P. Vila. Convergence of Sph Method for Scalar Nonlinear Conservation Laws. *SIAM Journal of Numerical Analysis Vol. 37(3), pages 863-887*, 2000.

[94] B. Nayroles, G. Touzot, and P. Villon. Generalizing the Finite Element Method: Diffuse Approximation and Diffuse Elements. *Comput. Mech., 10, 307-318*, 1992.

[95] M. Pastor, B. Haddad, G. Sorbino, S. Cuome, and V. Drempetic. A Depth-integrated, Coupled SPH Model for Flow-like Landslides and related Phenomena. *Int. J. Numer. and Anal. Meth. in Geomech., 33:143-172*, 2009.

[96] L.C. Qui. Two-Dimensional SPH Simulations of Landslide-generated Water Waves. *Journal of Hydraulic Engineering, 134:668-671*, 2008.

[97] P. Randles and L. Libersky. Smoothed Particle Hydrodynamics: Some Recent Improvements and Applications. *Computer Methods in Applied Mechanics and Engineering Vol. 139(1-2), pp. 375-408*, 1996.

[98] R.D. Richtmyer and K.W. Morton. *Difference Methods for Initial Value Problems*. 2nd Edition, Wiley-Interscience, London, 1967.

[99] Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadephia, 2003.

[100] C.E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal, 27(3): 379-423*, 1948.

[101] D. Shepard. A Two Dimensional Function for Irregularly Spaced Data. In *ACM National Conference*, 1968.

[102] L. Song, J. Zhou, J. Guo, Q. Zou, and Y. Liu. A Robust Well-balanced Finite Volume Model for Shallow Water Flows with Wetting and Drying over Irregular Terrain. *Advances in Water Resources. Vol. 34. pages 915-932*, 2011.

[103] J.C Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Wadsworth and Brooks, Pacific Grove, CA, 1989.

[104] H. Takeda, S. Miyama, and M. Sekiya. Numerical Simulation of Viscous Flow by Smoothed Particle Hydrodynamics. *Progress of Theoretical Physics 92 (5) pp. 939-960*, 1994.

[105] M. Tavelli and M. Dumbser. A High Order Semi-implicit Discontinuous Galerkin Method for the Two Dimensional Shallow Water Equations on Staggered Unstructured Meshes. *Applied Mathematics and Computation, Vol. 234, pages 623-644*, 2014.

[106] W.C. Thacker. Some Exact Solutions to the Nonlinear Shallow-Water Wave Equations. *Journal of Fluid Mechanics. Vol. 107, pages 499-508*, 1981.

[107] E.F. Toro. *Shock-Capturing Methods for Free-Surface Shallow Flows*. John Wiley and Sons, Ltd, 2001.

[108] L. N. Trefethen and III D. Bau. *Numerical Linear Algebra*. SIAM, Society for Industrial and Applied Mathematics, Philadephia, 1997.

[109] C. Ulrich. *Smoothed-Particle-Hydrodynamics Simulation of Port Hydrodynamic Problems*. PhD thesis, Technical University Hamburg-Harburg, 2013.

[110] R. Vacondio. *Shallow Water and Navier-Stokes SPH-like Numerical Modelling of Rapidly Varying Free-surface Flows*. PhD thesis, University of Parma, Italy, 2010.

[111] R. Vacondio, B.D. Rogers, P.K. Stansby, and P. Mignosa. SPH Modeling of Shallow Flow with Open Boundaries for Practical Flood Simulation. *Journal of Hydraulic Engineering, 138:530-541*, 2012.

[112] R. Vacondio, B.D. Rogers, P.K. Stansby, and P. Mignosa. Shallow Water SPH for Flooding with Dynamic Particle Coalescing and Splitting. *Advances in Water Resources Vol. 58, pages 10-23*, 2013.

[113] R.S. Varga. *Matrix Iterative Analysis*. Springer-Verlag, 2000.

[114] S. Vater, N. Beisiegel, and J. Behrens. A Limiter-based Well-balanced Discontinuous Galerkin Method for Shallow Water Flows with Wetting and Drying: One-dimensional Case. *Advances in Water Resources Vol. 85, pages 1-13*, 2015.

[115] J. Vila. On Particle Weighted Methods and Smoothed Particle Hydrodynamics. *Mathematical Models and Methods in Applied Sciences 9 (2) pp. 161-210*, 1999.

[116] S. Watkins, A. Bhattal, N. Francis, J. Turner, and A. Whitworth. A New Prescription for Viscosity in Smoothed Particle Hydrodynamics. *Astronomy and Astrophysics 119 177-187*, 1996.

[117] H. Wendland. Piecewise Polynomial, Positive Definite and Compactly Supported Radial Functions of Minimal Degree. *Adv. Comput. Math., 4:389-396*, 1995.

[118] Y. Xing, X. Zhang, and C.-W. Shu. Positivity-preserving High Order Well-balanced Discontinuous Galerkin Methods for the Shallow Water Equations. *Advances in Water Resources Vol. 33, pages 1476-1493*, 2010.

[119] S. Zhang, K. Morita, K. Fukuda, and N. Shirakawa. An Improved MPS Method for Numerical Simulations of Convective Heat Transfer Problems. *International Journal for Numerical Methods in Fluids, 51: 31–47*, 2006.

[120] T. Zhu and S.N. Atluri. A Modified Collocation Method and a Penalty Formulation for Enforcing the Essential Boundary Conditions in the Element Free Galerkin Method. *Computational Mechanics, Vol. 21, pages 211-222*, 1998.

Curriculum Vitae

Adeleke Olusegun Bankole

Universität Hamburg
Department of Mathematics
Bundesstraße 55
20146 Hamburg
Germany

E-Mail: adeleke.bankole@gmail.com

Academic Qualifications

02/2017	Doctoral Dissertation with title *“A Semi-implicit Smoothed Particle Hydrodynamics (SPH) Method for the Numerical Simulation of Shallow Water Flows “* Supervisor: Prof. Dr. Armin Iske , Field: *Optimization and Approximation* and Prof. Dr. Thomas Rung, Field: *Fluid Dynamics and Ship Theory*.
05/2012 - 02/2017	Studies of Applied Mathematics at Universität Hamburg, Germany with degree Ph.D. Applied Mathematics.
09/2011	Master Thesis with title *“Mathematical Modelling and Numerical Simulation of Multiscale Phenomena in Three-Phase Flow“* Supervisor: Prof. Dr. Armin Iske.
09/2009 - 09/2011	Studies of Industrial Mathematics at Universität Hamburg, Germany with degree MSc. Industrial Mathematics.
09/2009 - 09/2011	Studies of Applied Mathematics at University of L'Aquila, Italy with degree MSc. Mathematical Engineering.
06/2009	Diploma Thesis with title *“Thermal Decomposition in 1-D Shear Flow of Generalized Newtonian Fluids“* Supervisor: Dr. Tirivanhu Chinyoka.
08/2008 - 06/2009	Studies of Mathematics at the University of the Western Cape, (African Institute for Mathematical Sciences (AIMS) at Stellenbosch University), Cape Town, South Africa with degree Postgraduate Diploma (PGD) Mathematical Sciences.
11/2007	Bachelor Thesis with title *“The Dynamic Response of Elastic Plates under Linearly Moving Masses or Loads“* Supervisor: Dr. Moses Sunday Dada.
01/2003 - 11/2007	Studies of Mathematics at the University of Ilorin, Nigeria with degree BSc. Mathematics.

Research Interests

Semi-implicit meshfree discretization techniques for free-surface hydrodynamic flows.

Meshfree particle method (Smoothed Particle Hydrodynamics).

Adaptive kernel-based particle simulation methods for large-scale high performance computations.

Incompressible Navier Stokes equations and the shallow water equations.

Hyperbolic Conservation Laws

Mathematical modelling and numerical simulation of multiphase flows in porous media.

Mathematical modelling and numerical simulation of generalized Newtonian/non-Newtonian fluids.

Publications

A.O. Bankole, M. Dumbser, A. Iske, T. Rung: *A wetting and drying semi-implicit SPH algorithm for the shallow water equations*, published in the 11th International SPHERIC Workshop Proceedings, Munich, Germany, (2016).

A.O. Bankole, M. Dumbser, A. Iske, T. Rung: *A meshfree semi-implicit SPH method for free surface flow*, Report in the HBAM series with Archive Number $2015-44$. To appear in Lecture Notes in Computational Science and Engineering: Meshfree Methods for Partial Differential Equations VIII, Springer-Verlag, Editors: M. Griebel and M.A. Schweitzer.

A.O. Bankole, M. Dumbser, A. Iske, T. Rung: *A semi-implicit SPH scheme for the two-dimensional shallow water equations*, published in the 10th International SPHERIC Workshop Proceedings, Parma, Italy, (2015).

A.O. Bankole, M. Dumbser, A. Iske, T. Rung: *A semi-implicit SPH scheme for the shallow water equations*, published in the 9th International SPHERIC Workshop Proceedings, Paris, France, (2014).

Teaching Experience

10/2016 - 03/2017	Teaching Assistant “*Department of Mathematics, University of Hamburg* “ (Germany). Responsibilities: “*Anchoring and Correcting of Exercise Classes: Computerized Tomography - The Mathematics of Medical Imaging. Lecture by Prof. Dr. Armin Iske.* “
04/2016 - 09/2016	Teaching Assistant “*Department of Mathematics, University of Hamburg* “ (Germany). Responsibilities: “*Anchoring and Correcting of Exercise Classes: Numerical Approximation of Partial Differential Equations by Finite Differences and Finite Volumes. Lecture by Prof. Dr. Jens Struckmeier.* “
10/2015 - 03/2016	Teaching Assistant “*Department of Mathematics, University of Hamburg* “ (Germany). Responsibilities: “*Anchoring and Correcting of Exercise Classes: Computerized Tomography - The Mathematics of Medical Imaging. Lecture by Prof. Dr. Armin Iske.* “
04/2015 - 09/2015	Teaching Assistant “*Department of Mathematics, University of Hamburg* “ (Germany). Responsibilities: “*Anchoring and Correcting of Exercise Classes: Numerical Approximation of Partial Differential Equations by Finite Differences and Finite Volumes. Lecture by Prof. Dr. Jens Struckmeier.* “
05/2012 - 03/2015	Research Scientist “*Department of Mathematics, University of Hamburg* “ (Germany). Responsibilities: “*Organization of Seminars, Workshops.* “
09/2008 - 06/2009	Mathematics Tutor “*African Institute for Mathematical Sciences Schools Enrichment Center AIMSSEC* “ (South Africa). Course taught: “*Creative Thinking Approach in Problem Solving.* “
01/2004 - 08/2007	Mathematics Tutor, Academic Director, Librarian, “*University of Ilorin Christian Union and Mathematics Student Christian Fellowship* “ (Nigeria). Courses taught: “*Linear Algebra, Analysis, Calculus, Mathematical Methods, Differential Equations, Introduction to Fluid Dynamics.* “

Conferences and Workshops

** = Presentation was given*

*Winter School on Numerical Analysis of Multiscale Problems, 9.-13. January 2017, Bonn (Germany).

*SPHERIC2016: 11th International SPHERIC SPH-Workshop, 13.-16. June 2016, Munich (Germany).

*8th International Workshop on Meshfree Methods for Partial Differential Equations, 7.-9. September 2015, Bonn (Germany).

*SPHERIC2015: 10th International SPHERIC SPH-Workshop, 16.-18. June 2015, Parma (Italy).

*HONOM2015: European Workshop on High Order Nonlinear Numerical Methods for Evolutionary Partial Differential Equations, 16.-20. March 2015, Trento (Italy).

*Young Researchers Meeting and CSE Workshop, 9.-13. March 2015, Ploen (Germany).

RRW2015: 25th Rhein-Ruhr Workshop on Applied Mathematics, 30.-31. January 2015, Bestwig - Sauerland (Germany).

High Performance Computing Summer School, Hartree Center, Science and Technology Facilities Council (STFC), 21.-25. July 2014, Daresbury, Warrington (England)

*SPHERIC2014: 9th International SPHERIC SPH-Workshop, 3.-5. June 2014, Paris (France).

*Young Researchers Meeting and CSE Workshop, 24.-28. March 2014, Ploen (Germany).

7th International Workshop on Meshfree Methods for Partial Differential Equations, 9.-11. September 2013, Bonn (Germany).

SPHERIC2013: 8th International SPHERIC SPH-Workshop, 4.-6. June 2013, Trondheim (Norway).

*Young Researchers Meeting and CSE Workshop, 22.-24. May 2013, Ploen (Germany).

*RRW2013: 23rd Rhein-Ruhr Workshop on Applied Mathematics, 1.-2. February 2013, Bestwig - Sauerland (Germany).

SeisMath Intensive Summer School 2012: Mathematical Models in Seismology, 27. August-7. September 2012, L'Aquila (Italy).

Workshop on Model Order Reduction MOA 25.-27. July 2012, Hamburg (Germany).

Summer School on Model Order Reduction MOA 23.-24. July 2012, Hamburg (Germany).

Scholarships and Awards

Hamburg Research Foundation PhD. Scholarship (05/2012 - 06/2015)

Erasmus Mundus MSc. Scholarship - Bursary by the European Commission (2009 - 2011)

African Institute for Mathematical Science (AIMS) Postgraduate Diploma (PGD) Scholarship (2008 - 2009)

Best Graduating Mathematics Student - University of Ilorin, Nigeria. (Class of 2007)

Best Graduating Faculty Student - University of Ilorin, Nigeria. (Faculty of Science 2007 generation)

Organizations

Society for Industrial and Applied Mathematics (SIAM) *"Member "*

Smoothed Particle Hydrodynamics European Research Interest Community (SPHERIC) *"Member "*

Languages

Yoruba *"Advanced and Proficient User "*

English *"Advanced and Proficient User "*

German *"Modest User "*

Computer Skills and Competences

Ubuntu Linux, Unix Applications

Microsoft Office Applications

Programming in MATLAB, Fortran, C, Python

Typesetting with $\LaTeX$