Two-dimensional shock capturing numerical simulation of shallow water flow applied to dam break analysis

This item was submitted to Loughborough University’s Institutional Repository by the/an author.

Additional Information:

- A Doctoral Thesis. Submitted in partial fulfillment of the requirements for the award of Doctor of Philosophy of Loughborough University.

Metadata Record: https://dspace.lboro.ac.uk/2134/7750

Publisher: © Fayaz A. Khan

Please cite the published version.
This item was submitted to Loughborough's Institutional Repository (https://dspace.lboro.ac.uk/) by the author and is made available under the following Creative Commons Licence conditions.

![Creative Commons Licence](https://creativecommons.org/licenses/by-nc-nd/2.5/)

**Attribution-NonCommercial-NoDerivs 2.5**

You are free:

- to copy, distribute, display, and perform the work

Under the following conditions:

**Attribution.** You must attribute the work in the manner specified by the author or licensor.

**Noncommercial.** You may not use this work for commercial purposes.

**No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

Your fair use and other rights are in no way affected by the above.

This is a human-readable summary of the Legal Code (the full license).

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/
Two-Dimensional Shock Capturing Numerical Simulation of Shallow Water Flow Applied to Dam Break Analysis.

Fayaz A Khan
Civil and Building & Engineering
Loughborough University

A Thesis submitted for the Degree of
Doctor of Philosophy
October 2010
©2010 - Fayaz A Khan

All rights reserved.
Dedicated to the flood victims of Pakistan...
Acknowledgement

I would like to take the opportunity to express my gratitude to my supervisor Dr. Cecil F Scott for his persistent motivation, support and farsighted guidance from the beginning of my research work. His broad knowledge and deep insights has been an important resource to me in writing this thesis. The discussions and critiques made by him were critical to the progress of this work. I would like to express my sincere thanks to my Directors of Research, Prof. Jim Chandler and Prof. Graham Sander, for providing me encouragement and support in various ways.

My deepest gratitude goes to my family for their unflagging love and support throughout my life, without which this dissertation is simply impossible.

I would like to thank all colleagues and friends in the Department of Civil & Building Engineering for all the help and encouragement, I have received. A special thanks to all my research hub buddies Abdul, Benoit, Maninder, Peter, Dan, Naag, Amey, Camilo, Omid, Salih, Nabeel, Milanei, Waseem and Sajjad who made the hub a convivial place to work. As a result, research life became smooth and rewarding for me.

Loughborough Fayaz A Khan
October, 2010
Abstract

With the advances in the computing world, computational fluid dynamics (CFD) is becoming more and more critical tool in the field of fluid dynamics. In the past few decades, a huge number of CFD models have been developed with ever improved performance.

In this research a robust CFD model, called Riemann2D, is extended to model flow over a mobile bed and applied to a full scale dam break problem. Riemann2D, an object oriented hyperbolic solver that solves shallow water equations with an unstructured triangular mesh and using high resolution shock capturing methods, provides a generic framework for the solution of hyperbolic problems. The object-oriented design of Riemann2D has the flexibility to apply the model to any type of hyperbolic problem with the addition of new information and inheriting the common components from the generic part of the model. In a part of this work, this feature of Riemann2D is exploited to enhance the model capabilities to compute flow over mobile beds. This is achieved by incorporating the two dimensional version of the one dimensional non-capacity model for erodible bed hydraulics by Cao et al. (2004).

A few novel and simple algorithms are included, to track the wet/dry and dry/wet fronts over abruptly varying topography and stabilize the solution while using high resolution shock capturing methods. The negative depths computed from the surface gradient by the limiters are algebraically adjusted to ensure depth positivity. The friction term contribution in the source term, that creates unphysical values near the wet/dry fronts, are resolved by the introduction of a limiting value for the friction term.

The model is validated using an extensive variety of tests both on fixed and mobile beds. The results are compared with the analytical, numerical and experimental results available in the literature. The model is also tested against the actual field data of 1957 Malpasset dam break. Finally, the model is applied to simulate dam break flow of Warsak Dam in Pakistan. Remotely sensed topographic data of Warsak dam is used to improve the accuracy of the solution.

The study reveals from the thorough testing and application of the model that the simulated results are in close agreement with the available analytical, numerical and experimental results. The high resolution shock capturing methods give far better results than the traditional numerical schemes. It is also concluded that the object
oriented CFD model is very easy to adapt and extend without changing the generic part of the model.

**Keywords:** Computational fluid dynamics, Object oriented, hyperbolic solver, shallow water equation, unstructured triangular mesh, high resolution, shock capturing, wet/dry, remote sensing.
# CONTENTS

## List of Tables

* List of Tables

## List of Figures

* List of Figures

## 1 Introduction

1.1 Introduction ................................................. 1
1.2 Historical Background ...................................... 2
1.3 Objectives of the Research ................................. 3
1.4 Research Methodology .................................. 3
1.5 Thesis Outline ............................................. 4

## 2 Literature Survey

2.1 Introduction ................................................. 6
2.2 Hyperbolic Problems ...................................... 6
2.2.1 Partial Differential Equations ......................... 6
2.2.2 Solutions of Partial Differential Equations .......... 8
2.2.3 Hyperbolic PDEs ...................................... 9
2.2.4 Hyperbolic Systems ................................ 10
2.2.5 Conservation Laws ................................ 11
2.2.6 Numerical Methods for Hyperbolic PDEs ............. 12
2.2.7 Finite Volume Method for Hyperbolic Problems .... 14
### 2.3 High Resolution Methods

- 2.3.1 High Resolution Methods for Structured Meshes
- 2.3.2 High Resolution Methods for Unstructured Meshes
- 2.3.3 Numerical Methods for High Resolution Shock Capturing

### 2.4 Shallow Water Equations

- 2.4.1 Riemann Problem
- 2.4.2 Shallow flow over Mobile Bed
- 2.4.3 Source Terms
- 2.4.4 Wetting and Drying

### 2.5 Object Oriented Modelling

### 3 Numerical Scheme

#### 3.1 Introduction

#### 3.2 Finite Volume Formulation

- 3.2.1 The One Dimensional Problem
- 3.2.2 The Two Dimensional Problem

#### 3.3 Riemann Problem

- 3.3.1 The Riemann Solver of Roe
  - 3.3.1.1 The Intercell Flux
- 3.3.2 The HLL (Harten-Lax-Van Leer) Solver
- 3.3.3 The HLLC Approximate Riemann Solver

#### 3.4 Numerical Methods

- 3.4.1 Traditional Numerical Methods
  - 3.4.1.1 An Unstable Flux
  - 3.4.1.2 The Lax-Friedrichs Method
  - 3.4.1.3 The Richtmyer Two-Step Lax-Wendroff Method
  - 3.4.1.4 The Lax-Wendroff Method
  - 3.4.1.5 The Beam-Warming method
- 3.4.2 Upwind Methods
- 3.4.3 Godunov’s Method
- 3.4.4 Godunov’s Method For Triangular Meshes
- 3.4.5 Total Variation Diminishing (TVD) Method
- 3.4.6 High Resolution Methods for Riemann2D

---

**Contents**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.3 High Resolution Methods</td>
<td>17</td>
</tr>
<tr>
<td>2.3.1 High Resolution Methods for Structured Meshes</td>
<td>18</td>
</tr>
<tr>
<td>2.3.2 High Resolution Methods for Unstructured Meshes</td>
<td>19</td>
</tr>
<tr>
<td>2.3.3 Numerical Methods for High Resolution Shock Capturing</td>
<td>21</td>
</tr>
<tr>
<td>2.4 Shallow Water Equations</td>
<td>22</td>
</tr>
<tr>
<td>2.4.1 Riemann Problem</td>
<td>23</td>
</tr>
<tr>
<td>2.4.2 Shallow flow over Mobile Bed</td>
<td>24</td>
</tr>
<tr>
<td>2.4.3 Source Terms</td>
<td>25</td>
</tr>
<tr>
<td>2.4.4 Wetting and Drying</td>
<td>27</td>
</tr>
<tr>
<td>2.5 Object Oriented Modelling</td>
<td>28</td>
</tr>
<tr>
<td>3 Numerical Scheme</td>
<td>32</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>32</td>
</tr>
<tr>
<td>3.2 Finite Volume Formulation</td>
<td>33</td>
</tr>
<tr>
<td>3.2.1 The One Dimensional Problem</td>
<td>33</td>
</tr>
<tr>
<td>3.2.2 The Two Dimensional Problem</td>
<td>36</td>
</tr>
<tr>
<td>3.3 Riemann Problem</td>
<td>37</td>
</tr>
<tr>
<td>3.3.1 The Riemann Solver of Roe</td>
<td>38</td>
</tr>
<tr>
<td>3.3.1.1 The Intercell Flux</td>
<td>40</td>
</tr>
<tr>
<td>3.3.2 The HLL (Harten-Lax-Van Leer) Solver</td>
<td>42</td>
</tr>
<tr>
<td>3.3.3 The HLLC Approximate Riemann Solver</td>
<td>44</td>
</tr>
<tr>
<td>3.4 Numerical Methods</td>
<td>44</td>
</tr>
<tr>
<td>3.4.1 Traditional Numerical Methods</td>
<td>44</td>
</tr>
<tr>
<td>3.4.1.1 An Unstable Flux</td>
<td>45</td>
</tr>
<tr>
<td>3.4.1.2 The Lax-Friedrichs Method</td>
<td>45</td>
</tr>
<tr>
<td>3.4.1.3 The Richtmyer Two-Step Lax-Wendroff Method</td>
<td>45</td>
</tr>
<tr>
<td>3.4.1.4 The Lax-Wendroff Method</td>
<td>46</td>
</tr>
<tr>
<td>3.4.1.5 The Beam-Warming method</td>
<td>47</td>
</tr>
<tr>
<td>3.4.2 Upwind Methods</td>
<td>47</td>
</tr>
<tr>
<td>3.4.3 Godunov’s Method</td>
<td>50</td>
</tr>
<tr>
<td>3.4.4 Godunov’s Method For Triangular Meshes</td>
<td>51</td>
</tr>
<tr>
<td>3.4.5 Total Variation Diminishing (TVD) Method</td>
<td>53</td>
</tr>
<tr>
<td>3.4.6 High Resolution Methods for Riemann2D</td>
<td>54</td>
</tr>
<tr>
<td>Section Count</td>
<td>Title</td>
</tr>
<tr>
<td>---------------</td>
<td>-------</td>
</tr>
<tr>
<td>3.4.7</td>
<td>Data Reconstruction</td>
</tr>
<tr>
<td>3.4.7.1</td>
<td>No Limiters</td>
</tr>
<tr>
<td>3.4.7.2</td>
<td>Superbee and Minmod Limiters</td>
</tr>
<tr>
<td>3.4.7.3</td>
<td>Limited Central Difference (LCD) Limiter</td>
</tr>
<tr>
<td>3.4.7.4</td>
<td>Extended vanLeer (EV) Limiter</td>
</tr>
<tr>
<td>3.4.7.5</td>
<td>Maximum Limited Gradient (MLG) Limiter</td>
</tr>
<tr>
<td>3.4.7.6</td>
<td>The MUSCL-Hancock Scheme</td>
</tr>
<tr>
<td>4</td>
<td>Shallow Water Flow</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>4.2</td>
<td>Free Surface Flows</td>
</tr>
<tr>
<td>4.3</td>
<td>Shallow Flow Equations</td>
</tr>
<tr>
<td>4.4</td>
<td>Derivation of the SWE</td>
</tr>
<tr>
<td>4.5</td>
<td>Hyperbolic Character of the Shallow Water Equations</td>
</tr>
<tr>
<td>4.6</td>
<td>Approximate Riemann Solvers</td>
</tr>
<tr>
<td>4.6.1</td>
<td>The Roe Solver</td>
</tr>
<tr>
<td>4.6.2</td>
<td>The HLL Solver</td>
</tr>
<tr>
<td>4.7</td>
<td>Source Terms</td>
</tr>
<tr>
<td>4.7.1</td>
<td>Coriolis Factor</td>
</tr>
<tr>
<td>4.7.2</td>
<td>Bed Resistance</td>
</tr>
<tr>
<td>4.8</td>
<td>Conclusion</td>
</tr>
<tr>
<td>5</td>
<td>Extension of Riemann2D</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>5.2</td>
<td>Cao Pender Erosion Model</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Governing Equations</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Model Closure</td>
</tr>
<tr>
<td>5.2.3</td>
<td>CaoPenderErosion Package</td>
</tr>
<tr>
<td>5.3</td>
<td>Wetting and Drying</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Wet and Dry Cells</td>
</tr>
<tr>
<td>5.3.2</td>
<td>No flow condition</td>
</tr>
<tr>
<td>5.3.3</td>
<td>Partially Submerged Cells</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Friction Term Limitation</td>
</tr>
<tr>
<td>5.4</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>
6 Validation of Riemann2D

6.1 Introduction ................................................. 94
6.2 Tests over fixed bed ........................................... 94
  6.2.1 Dam break problem ....................................... 95
  6.2.2 Dam-break with wetting and drying ....................... 97
  6.2.3 Dam-break on slope with wetting and drying ............. 99
  6.2.4 Sub-critical, Super-critical and Transitional Flow .... 101
  6.2.5 Flow over a bump ....................................... 106
  6.2.6 Dam-break in a converging-diverging channel .......... 108
6.3 Tests over erodible bed ..................................... 112
  6.3.1 Dam break over erodible bed ............................ 112
  6.3.2 Dam-break over erodible flat bed with wetting and drying . 118
  6.3.3 Flow in Channel with Spur Dykes ....................... 120

7 Application of Riemann2D .................................... 123

7.1 Introduction ................................................ 123
7.2 Malpasset Dam Break ........................................ 123
  7.2.1 Brief History .......................................... 123
  7.2.2 Topography ............................................ 125
  7.2.3 Field Data ............................................ 128
  7.2.4 Conditions for Simulation ................................ 129
  7.2.5 Results ............................................... 132
7.3 Warsak Dam-Pakistan ....................................... 136
  7.3.1 Brief History .......................................... 136
  7.3.2 Topography ............................................ 138
  7.3.3 Conditions for simulation ............................... 140
  7.3.4 Results ............................................... 142

8 Summary, Conclusions, Recommendations and Future Work 152

8.1 Summary .................................................. 152
8.2 Conclusion ................................................. 154
8.3 Future Work ................................................. 155

References ..................................................... 169
2.1 Classification of PDEs based on the coefficients of the second order terms in equation 2.1. .................................................. 7

6.1 Definition and type of boundary condition used for the sub-parts N.1.1 to N.1.5 in the sub-critical, super-critical and transitional flows test case. 101

6.2 Hydraulic condition for the experiment and simulation . . . . . . . . . . . . . . 120

7.1 Coordinates and flood wave arrival times of the electric transformers . . 128
7.2 Coordinates and peak water depths of the points surveyed by police . . 129
7.3 Coordinates and flood wave arrival times of the electric transformers . . 136
LIST OF FIGURES

2.1 Information propagation along characteristics of the wave equation (a) elliptic (b) parabolic (c) hyperbolic PDE (Recktenwald, 2004) ....... 8

3.1 Illustration of finite volume method for updating the cell average $U^n_i$ by fluxes at the cell edges. One-dimensional problem ............... 33

3.2 Illustration of the Riemann problem (a) initial data at $t = 0$, consists of two constant states separated by a discontinuity at $x = 0$. (b) Solution in the $x - t$ plane for the linear advection equation with positive characteristic speed $a$ ..................... 37

3.3 Approximate HLL Riemann solver. Solution in the star region consists of a single state $U^{hll}$ separated from data states by two waves of speeds $S_L$ and $S_R$. .................................................. 42

3.4 Approximate HLLC solver. In HLLC solver the contact and shear waves are restored. .................................................. 44

3.5 Two representations of the upwind method for advection. (a) If $U^n_i$ represents the value at a point where information is available, then we can trace the characteristic back and interpolate. (b) If $U^n_i$ represents the cell average, then the flux at the interface is determined by the cell value on the upwind side. ..................... 48
3.6 An illustration of reconstruct-evolve-average algorithm for the case of linear acoustics (adapted from LeVeque (2002)). The Riemann problem is solved at each cell interface, and the wave structure is used to determine the exact solution time $\Delta t$ later. This solution is updated over the elements to determine $U_{n+1}$.

3.7 (a) Constant reconstruction vs (b) piecewise linear reconstruction.

3.8 Naming convention for the Limiting Procedure defined for a triangular control volume (left) and a piecewise constant reconstruction of the solution (right).

4.1 River flow is one of the examples of free surface flows.

4.2 River flow is one of the examples of free surface flows.

4.3 River flow is one of the examples of free surface flows.

4.4 Failure of the Roe solver when the left and right states, $u_l$ and $u_r$, lie in particular regions of phase space. In this example the left and right states are shallow, with opposite velocities. The arrows are in the direction of the Roe eigenvectors, the intersection of which is the middle state $u^*$, corresponding to a negative depth $h$ George (2004).

4.5 Constructing a solution using conservation and estimated speeds. If the speeds $s^1$ and $s^2$ are predetermined estimates, an approximate solution can be uniquely determined by applying the conservation law to the region $\omega = [x_l, x_r]$ surrounding the interface George (2004).

5.1 Different levels of Riemann2D.

5.2 Relationship between various classes in the Generic Package, the Shallow Water Package and the CaoPenderErosion Package.

5.3 WET and DRY cells. Side AB and BC are allowed to calculate fluxes while the fluxes across CD are set equal to zero during computation.

5.4 Various scenarios of flow over variable topography. In case (a) and (b) the fluxes are set equal to zero, (c) and (e) fluxes in the left direction are allowed, (d) is allowed to permit fluxes in the right direction and for case (f) there is no need to modify the fluxes.

5.5 Data reconstruction for triangular mesh.

5.6 Adjustment of water depths at the sides of a cell in different cases.
6.1 Comparison between the numerical and exact solutions for a horizontal frictionless bed: (a) free surface and (b) velocity profiles. 96
6.2 Dam break after 24s on flat bed with wetting and drying: (a) free surface elevation (b) velocity vectors 98
6.3 Dam break on after 24s on sloping bed with wetting and drying: (a) free surface elevation (b) velocity vectors 100
6.4 Comparison of numerical and analytical results for (a) N.1.1 and (b) N.1.2 tests. The flow enters at the left and leaves on the right of the channel in both the tests. 103
6.5 Comparison of numerical and analytical results for (a) N.1.3 and (b) N.1.4 tests. The flow enters at the left and leaves at the right of the channel in N.1.3 test while for the N.1.4, the flow enters at the left and is reflected from the wall at the right of the channel. 104
6.6 Comparison of numerical and analytical results for N.1.5 test. The flow enters at the left and leaves at the right of the channel. 105
6.7 Flow over a bump. 107
6.8 Schematic illustration of the contraction and expansion test. 108
6.9 Comparison of the numerical and experimental results for flow in a converging and diverging channel at stations (a) S1 (b) S2. 110
6.10 Comparison of the numerical and experimental results for flow in a converging and diverging channel at stations (a) S3 (b) S4. 111
6.11 Evolution of free surface profile and bed after (a) 20s and (b) 30s. 114
6.12 Evolution of free surface profile and bed after (a) 40s and (b) 1min. 115
6.13 Evolution of free surface profile and bed after (a) 2min and (b) 8min. 116
6.14 Evolution of free surface profile and bed after (a) 14min and (b) 20min. 117
6.15 Dam break after 24s on flat erodible bed with wetting and drying: (a) free surface elevation (b) eroded bed (c) velocity vectors 119
6.16 Schematic illustration of the contraction and expansion test. 120
6.17 Eroded bed near the first two spur dykes, experimental results (top) and numerical results of the 3D model (bottom). 121
6.18 Eroded bed near the first two spur dykes, simulated by Riemann2D. 121
6.19 Velocity vectors in the channel near the first two spur dykes. 122
List of Figures

7.1 View of the Malpasset Dam break after 12 days of the failure. .......... 124
7.2 The old version of the topographic map of the Reyran river valley, that was digitized by EDF. .................................................. 126
7.3 Bed topography of the computational domain for Malpasset Dam. .... 127
7.4 Location of the seventeen points selected from the points surveyed by the police near the banks. ......................................... 128
7.5 Unstructured triangular mesh for Malpasset Dam, showing the local refinement of the mesh in the valley downstream of the dam where the bed variation is strong. ........................................... 131
7.6 Water surface elevation of Malpasset dam, 2800 seconds after the dam break. ............................................................. 132
7.7 Water depths of Malpasset dam, 2800 seconds after the dam break. ... 133
7.8 Velocity vector, representing the speed and direction of the flow, 2800 seconds after the dam break. .................................... 134
7.9 Comparison of the simulated results from Riemann2D and the observed points for the maximum water levels at the banks of the Reyran river after the dam break. .................................................. 135
7.10 Location map of Warsak Dam, Peshawar. ................................. 137
7.11 Bed topography of the computational domain for Warsak Dam. ....... 139
7.12 Unstructured triangular mesh for Warsak Dam, showing the local refinement of the mesh in the valley downstream of the dam where the bed variation is strong. ........................................... 141
7.13 Water surface elevation 30 minutes after the dam break. ................ 143
7.14 Water depths 30 minutes after the dam break. ........................... 144
7.15 Velocity vector, representing the speed of the flow, 30 minutes after the dam break. ..................................................... 145
7.16 Water surface elevation 60 minutes after the dam break. ................ 146
7.17 Water depths 60 minutes after the dam break. ........................... 147
7.18 Velocity vector, representing the speed of the flow, 60 minutes after the dam break. ..................................................... 148
7.19 Water surface elevation 2 hours after the dam break. ................... 149
7.20 Water depths 2 hours after the dam break. ............................... 150
7.21 Velocity vector, representing the speed of the flow, 2 hours after the dam break. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Chapter 1

Introduction

1.1 Introduction

Fluid dynamics is divided into theoretical and experimental branches. However, Computational Fluid Dynamics (CFD) is a third branch of fluid dynamics that uses numerical methods and algorithms to solve and analyze problems that involve fluid flows. In CFD, computers are used to perform the millions of calculations required to simulate the interaction of liquids and gases with surfaces defined by boundary conditions. CFD is a supplement rather than a replacement to the experiment or theory. It turns a computer into a virtual laboratory providing insight, foresight, return on investment and cost savings. In this work, some challenging issues of a CFD model, Riemann2D\(^1\), are discussed. Riemann2D is an object oriented hyperbolic solver, that solves shallow water equations using high resolution shock capturing methods on unstructured triangular mesh. In this work, the model is extended to flow over variable topography, wetting/drying problems, flow over erodible bed and finally, the model is applied to simulate the flood waves due to dam break.

\(^1\)Riemann2D, an object oriented hyperbolic solver applied to shallow water equations over fixed bed, initially developed by Cecil F Scott and Atmanand Jha, Department of Civil and Building Engineering, Loughborough University.
1.2 Historical Background

Flood is a temporary condition of partial or complete inundation of normally dry land areas from overflow of inland or from the unusual and rapid accumulation or runoff of surface waters from any source. Flood water claims thousands of lives every year and renders millions homeless in the world. Flood disasters account for about a third of all natural disasters throughout the world (by number and economic losses) and are responsible for more than half of the fatalities (Berz, 2000). Based on the source type, floods may be classified as rainfall floods, snow melt floods, sea surge or tidal floods and dam break floods. Although all these types of floods could be simulated using our proposed model, in this section we will discuss the floods caused by dam break events to expedite the importance of dam break analysis that could be vital in setting up emergency plans for the unlikely event of the dam break..

Dam breaks can cause significant loss of human life, especially when located in highly populated regions, such as is the case in Europe. The Malpasset Dam failure event in France occurred in December of 1959, releasing 50 million cubic meter of water that caused 384 deaths and 110 people missing (Zhang et al., 2007). The Banqiao Reservoir Dam and Shimantan Reservoir Dam are among 62 dams in Zhumadian Prefecture of China’s Henan Province that failed catastrophically in 1975 during Typhoon Nina. The dam failure caused the sudden loss of 18 GW of power. According to the Hydrology Department of Henan Province, approximately 26,000 people died from flooding and another 145,000 died during subsequent epidemics and famine. In addition, about 5,960,000 buildings collapsed, and 11 million residents were affected (Zhang et al., 2007).

The Teton dam breach in the United States in June of 1976 involved approximately 40,020 hectares of land inundated, 11 people dead, 25,000 homeless, and a total of 400 million US dollars in economic loss. On July 19, 1985, a fluorite tailings dam of Prealpi Mineraia failed at Stava, Trento, Italy. 200,000 $m^3$ of tailings flowed 4.2 $km$ downstream at a speed of up to 90 $km/h$, killing 268 people and destroying 62 buildings. The total surface area affected was 43.5 hectares (Zhang et al., 2007). On March 28, 2009, the Situ Gintung Dam burst that was overfilled by heavy rains in a southwestern suburb of Jakarta, Indonesia, killing at least 52 people and displacing a further 1,490. Officials, however, warn that the death toll will likely increase.
Chapter 1: Introduction

The above, few but not all, dam break events shows the importance of flood modeling that can be used in developing emergency plans in an unlikely event of a dam break. In recent years, it has become compulsory to set up emergency plans for dam breaks. Over the recent decades, there have been continuing efforts to enhance the understanding of flood modeling. Because real-time field measurements are prohibitively difficult to make, the majority of flood wave studies have been carried out in laboratories. It is noted, however, that physical experiments are largely constrained by the comparatively small spatial scales that can be realistically accommodated in laboratories, and thus may not be able to fully reveal the long-term mechanism of the flow. This shows the need of numerical modelling for the estimation of flood waves.

In recent years, various 1D and 2D computational hydrodynamic models have been developed that could be used for flood modelling. This work is a step forward in the development of a robust hydrodynamic model. The model is based on two-dimensional shallow water equations, solved on unstructured triangular mesh, using high resolution shock capturing techniques and applied to flow over fixed and mobile beds.

1.3 Objectives of the Research

The principal aim of this study is to develop a model based on 2D shallow water equations for simulating floods over fixed and mobile beds. The specific objectives to attain the aim of the study are:

- To show the importance of object oriented programming in the field of computational fluid dynamics.
- To develop a two dimensional hydrodynamic model based on shallow water equations that can simulate flood inundation over fixed and mobile beds.
- To study the applicability of the model with a case study of one of the basins in Pakistan.

1.4 Research Methodology

This research is based on four main components. Which are as follow:
Chapter 1: Introduction

1. **Extension of Riemann2D:** In this part of the work, the Riemann2D model is extended to two dimensional shallow water flow over mobile beds.

2. **Wetting and Drying Problem:** The wetting and drying issues like, flow over variable topography, wet/dry and dry/wet fronts, negative depths computed by the limiters and the friction term are addressed and some simple algorithms are introduced to cope with these issues.

3. **Validation of the Model:** A wide range of tests that cover the scenarios while dealing with real world problems, are carried out and the results are compared with numerical, analytical and experimental results.

4. **Application of the Model:** The model is applied to Malpasset dam France and Warsak dam Pakistan. Remotely sensed topographic data is used for the simulation of flood waves of Warsak dam.

1.5 **Thesis Outline**

The details of these chapters and appendices are as following:

- **Chapter 1** i.e., this chapter.

- **Chapter 2** presents literature review on the relevant methods of modelling hyperbolic and shallow water problems. Initially, this chapter describes the fundamental of PDEs and then selection of numerical method. The sections of this chapter provide brief background of the work done in the field of hyperbolic problems, high-resolution methods, shallow water equations, erodible bed hydraulics and object-oriented modelling.

- **Chapter 3** presents the finite volume formulation of the one- and two-dimensional hyperbolic problem, followed by discussion on Riemann problem and its solution. Later in the chapter, traditional numerical methods followed by the high-resolution methods used in Riemann2D are discussed.

- **Chapter 4** presents the theory of the shallow water equations. Initially, in this chapter the derivation of the two dimensional shallow water equations from Saint Venaint Equations is presented, followed by the discussion of hyperbolic
Chapter 1: Introduction

characteristics of the shallow water equations. At the end of the chapter the generic Riemann solvers discussed in Chapter 3 are extended to shallow water problem.

- **Chapter 5** presents the extension of Riemann2D. Initially the extension of Riemann2D to deal with flow over mobile bed problems is discussed. Later in the chapter some simple new algorithms are introduced to solve the wetting and drying problem and to make the model capable of handling real world problem.

- **Chapter 6** presents the results and discussion, when Riemann2D is applied to the shallow water equations over fixed and mobile bed problems. A series of test cases are presented to perform the model verification and validation. The tests results presented in this chapter are compared with the numerical, analytical and experimental results.

- **Chapter 7** In this chapter the application of Riemann2D to real world problem is discussed. First the model is applied to Malpasset Dam, France to see the performance of the model. For Malpasset Dam, that failed in December 1959, the local police recorded the maximum water levels at various points near the banks after the dam failure. The maximum water levels for the simulated flow at these locations are compared with the recorded data. In the next section the model is applied to Warsak dam located in the Khyber Pukhtoonkhwa province of Pakistan.

- **Chapter 8** presents summary of this work. Finally, future research possibilities are sketched.
Chapter 2

Literature Survey

2.1 Introduction

This chapter presents a brief review of the work done in the field of hyperbolic problems, numerical methods for hyperbolic PDEs, high resolution methods, shallow water flow over fixed and mobile beds and object oriented modelling in computational fluid dynamics. This chapter would help to provide a base for this study.

2.2 Hyperbolic Problems

In this section we will review the literature of Partial Differential Equations (PDEs), solution of PDEs, hyperbolic PDEs, hyperbolic systems and numerical methods for hyperbolic PDEs. Most of the text in this section is adopted from LeVeque (2002).

2.2.1 Partial Differential Equations

In mathematics, partial differential equations (PDE) are a type of differential equation, i.e., a relation involving an unknown function (or functions) of several independent variables and their partial derivatives with respect to those variables. Partial differential equations are used to formulate, and thus aid the solution of, problems involving functions of several variables; such as the propagation of sound or heat, electrostatics, electrodynamics, fluid flow, and elasticity.
The study of partial differential equations (PDEs) started in the 18th century with the work of Euler, d’Alembert, Lagrange and Laplace. PDEs are central tool in the description of mechanics of continua and more generally, as the principal mode of analytical study of models in the physical science (Brezis, 1988). The analysis of physical models has remained to the present day one of the fundamental concerns of the development of PDEs. Beginning in the middle of the 19th century, particularly with the work of Riemann, PDEs also became an essential tool in other branches of mathematics. Ames (1992) identified three basic types of physical problems that lead to PDEs, i.e., equilibrium, eigenvalue and propagation problems, which corresponds to elliptic, parabolic and hyperbolic type PDEs respectively (see figure 2.1 and table 2.1).

A second-order PDEs can be written as:

\[ Au_{xx} + 2Bu_{xy} + Cu_{yy} + Du_x + Eu_y + F = 0 \]  

(2.1)

Based on the matrix of coefficients of the second order terms in the above equation 2.1 i.e., \( Z \equiv \begin{bmatrix} A & B \\ B & C \end{bmatrix} \), PDEs are classified as shown in the table 2.1.

Table 2.1: Classification of PDEs based on the coefficients of the second order terms in equation 2.1.

<table>
<thead>
<tr>
<th>Type</th>
<th>Condition</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elliptic</td>
<td>( \det(Z) &gt; 0 )</td>
<td>Laplace’s Equation</td>
</tr>
<tr>
<td>Parabolic</td>
<td>( \det(Z) = 0 )</td>
<td>Diffusion/Schrödinger equation</td>
</tr>
<tr>
<td>Hyperbolic</td>
<td>( \det(Z) &lt; 0 )</td>
<td>Wave equation</td>
</tr>
</tbody>
</table>

In the Figure 2.1 three type of PDEs based on the propagation of information are shown, in which the domain is divided into:

- **Domain of Dependence:** For a point \( P \), it is defined as the region of the solution domain, upon which the solution at a point \( P, f(x_p, y_p) = f_p \) depends. In other words, \( f_p \) depends on everything that has happened in domain of dependence.

- **Domain of Influence:** For a point \( P \), it is defined as the region of the solution domain in which the solution at point \( f(x, y) \) depends on the solution at \( P, f_p \) In other words, \( f_p \) influences the solution at all points in the range of influence.
Chapter 2: Literature Survey

Figure 2.1: Information propagation along characteristics of the wave equation (a) elliptic (b) parabolic (c) hyperbolic PDE (Recktenwald, 2004).

Generally, elliptic equations have boundary conditions which are specified around a closed boundary (Figure 2.1(a)). This has domain of influence in all the directions and the domain of dependence is all of the boundary. Usually all the derivatives are with respect to spatial variables, such as in Laplace’s or Poisson’s Equation. Parabolic (Figure 2.1(b)) and hyperbolic equations (Figure 2.1(c)), by contrast, depend on time and have at least one open boundary. The boundary conditions for at least one variable, usually time, are specified initially and the system is integrated indefinitely. Thus, the diffusion equation and the wave equation contain a time variable and there is a set of initial conditions at a particular time. These properties are, of course, related to the fact that an ellipse is a closed object, whereas hyperbola and parabola are open.

2.2.2 Solutions of Partial Differential Equations

Generally the number of solutions satisfying a PDE are very large and the functions may be completely different from one another. However, when applying PDEs to model a physical system, other conditions are imposed, enabling a unique solution to be found. For example, the solution U may assume specific values on the boundary of the domain, and if, time dependent, have a condition specified at t = 0; these are the so called boundary conditions and initial conditions. This system of equations consisting of PDEs together with some initial and boundary values is properly posed (Harpham, 1997) if it has the following properties:
Chapter 2: Literature Survey

1. Existence: For any sufficiently smooth data functions (say from the initial conditions) there is a solution $U$ of the system of equations.

2. Uniqueness: There is at most one function $U$ satisfying the system of equations, for given data.

3. Continuity: The solutions $U$ corresponding to data which differ by small amounts in the appropriate norms, also differ by small amounts. (Continuity will normally imply uniqueness.)

2.2.3 Hyperbolic PDEs

Hyperbolic PDEs, as discussed in the above section, arise in a broad spectrum of disciplines where wave motion or advective transport is important: These areas include gas dynamics, acoustics, elastodynamics, optics, geophysics and biomechanics. Historically, many of the fundamental ideas were first developed for the special case of compressible gas dynamics (the Euler equations), for applications in aerodynamics, astrophysics, detonation waves, and related fields, where shock wave arise. The study of simpler equations such as the advective equations, Burgers’ equations and the shallow water equations have played an important role in the development of these methods. Often the model problem for testing methods has been the application to the Euler equation. This is also reflected in many of the texts on these methods. Nordstro & Gong (2006) states that these hyperbolic equations involved in modeling still remain a computational challenge both for academia and industry.

For a one-dimensional (in space) wave equation (hyperbolic PDE)

$$u_{tt} = cu_{xx} \quad 0 \leq x \leq L \quad t \leq 0,$$

with boundary and initial conditions

$$u(t,0) = u_0 \quad u(t,L) = u_L,$$

$$u(0,x) = f(x) \quad \frac{\partial u}{\partial t} |_{t=0} = g(x),$$

where $u = u(t,x)$ and $\sqrt{c}$ is the wave speed, the solution can be represented as a field in the semi-infinite strip, as shown in Figure 2.1. The state of the solution at time $t$ and position $x$ influences the solution at all other points in the forward wedge of space time labeled the domain of influence. This wedge is defined by two characteristics with slope $\pm c$.  

9
2.2.4 Hyperbolic Systems

In the hyperbolic systems of PDEs, the problems we consider are generally time-dependent, so that the solution depends on time as well as one or more spatial variables. In one space dimension, a homogeneous first order system of PDEs in \(x\) and \(t\) has the form

\[
 u_t(x,t) + A u_x(x,t) = 0
\]

in the simplest constant-coefficient linear case. Here \(u : \mathbb{R} \times \mathbb{R} = \mathbb{R}^m\) is a vector with \(m\) components representing the unknown functions (pressure, velocity, etc.) that are to be determined, and \(A\) is the constant \(m \times m\) real matrix. In order for this type of problem to be hyperbolic, the matrix \(A\) must satisfy certain properties.

“System 2.4 is said to be hyperbolic at a point \((x,t)\), if matrix \(A\) has \(m\) real eigenvalues \(\lambda_1, \lambda_2, ..., \lambda_m\) and a corresponding set of \(m\) linearly independent right eigenvectors \(K^{(1)}, K^{(2)}, ..., K^{(m)}\). The system is said to be strictly hyperbolic if the eigenvalues \(\lambda_i\) are all distinct (Toro, 1999).”

The eigenvalues \(\lambda_i\) of a matrix \(A\) are the solution of the characteristic polynomial

\[
 |A - \lambda_i I| = \text{det}(A - \lambda I) = 0
\]

where \(I\) is the identity matrix. The eigenvalues of the coefficient matrix \(A\) of a system of this form, equation 2.5, are also called as eigenvalues of the system. Physically these eigenvalues represent speeds of propagation of information. Whereas, the eigenvector of a matrix \(A\) corresponding to an eigenvalue \(\lambda_i\) of \(A\) is a vector, \(K^{(i)} = [K_1^{(i)}, K_2^{(i)}, ..., K_m^{(i)}]\) satisfying \(AK^{(i)} = \lambda_i K^{(i)}\).

The simplest case is the constant-coefficient scalar problem, in which \(m = 1\) and the matrix \(A\) reduces to a scalar value. This problem is hyperbolic provided the scalar \(A\) is real. This simple equation can model either advective transport or wave motion, depending on the context. For more detail refer LeVeque (2002).

Since the PDEs continue to hold away from discontinuities, one possible approach is to combine a standard finite volume method in smooth regions with some explicit procedure for tracking the location of discontinuities. This is the numerical analogue of the mathematical approach in which the PDEs are supplemented by jump conditions across discontinuities. This approach is often called as shock tracking or front tracking. In two-dimensions the discontinuities typically lie along curves or the surface of the
cell/element/grid and this makes it more complicated. Moreover in realistic problems there may be many such surfaces that interact in complicated ways as time evolves. This approach can be found in the work by Glimm et al. (1981), Davis (1992), Grove (1994), LeVeque (1992), Shyue (1995) and Mao (2000).

However, in this work a different approach, known as shock capturing, is adopted, where the goal is to capture discontinuities in the solution automatically, without explicitly tracking them. The success of this method lies on implicitly incorporating the correct jump condition, reducing the smearing to a minimum, and no introduction of the non-physical oscillation near the discontinuities. The development of high-resolution shock-capturing schemes has a long history (see, the classical references [Godunov (1959), Harten & Hyman (1983), vanLeer (1979) and Yee (1987)] or the recent textbooks [Godlewski & Raviart (1996), Kroner (1997), LeVeque (1992) and LeVeque (2002)]). The main reason for using shock-capturing methods is that the high-resolution finite volume methods based on Riemann solutions often perform well and are much simpler to implement than shock-tracking methods.

2.2.5 Conservation Laws

This work is concerned with an important class of homogeneous hyperbolic equations called conservation laws. In physics, a conservation law states that a particular measurable property of an isolated physical system does not change as the system evolves. The simplest form of a conservation law is a one-dimension PDE

$$u_t(x,t) + f(u(x,t))_x = 0; \quad (2.6)$$

where $f(u)$ is the flux function. This form (equation 2.6) is also know as the differential form of the conservation laws. Rewriting 2.6 in the quasi-linear form

$$u_t + f'(u)u_x = 0 \quad (2.7)$$

suggests that the equation is hyperbolic if the flux Jacobian matrix $f'(u)$ satisfies the conditions previously given for the matrix $A$ (see equation 2.4). In fact the linear problem in equation 2.4 is a conservation law with the linear flux problem $f(u) = Au$, but many physical problems give rise to nonlinear conservation laws in which $f(u)$ is a non-linear function of $u$. 
The principal feature of nonlinear hyperbolic conservation laws, demonstrated in the physical phenomenon of the breaking of waves, is the breakdown of classical solutions and the development of discontinuities that propagate as shock waves (e.g., propagating phase boundaries, fluid interfaces, gravitational waves, etc.), which play a dominant role in multiple areas of physics: astrophysics, cosmology, dynamics of (solid-solid) material interfaces, multi-phase (liquid-vapour) flows, combustion theory, etc. In recent years, major progress has been made in both the theoretical and numerical aspects of the field, while the number of applications has highly increased. Dafermos et al. (2003) states that the challenge for mathematicians is to comprehend the properties of these non-linear waves and their relationships with the dynamics of many physical phenomena.

2.2.6 Numerical Methods for Hyperbolic PDEs

There are many methods in practice to solve PDEs, of which the four commonly used methods, applied to hyperbolic PDEs, are:

- method of characteristics (MOC),
- finite element method (FEM),
- finite difference method (FDM),
- finite volume method (FVM).

Method Of Characteristics (MOC): In this method the PDEs are transformed into ordinary differential equations (ODEs) along the characteristics. The transformed ODEs are then solved numerically using explicit or implicit schemes. This method is an analysis based method. Classical work using these methods can be seen in the literature by Abott (1966), Simpson (1967), Abbott & Verwey (1970) and Edenhofer & Schmitz (1981).

Consequently, the governing equations, which are in continuous forms, are transformed into discrete forms, which then result in series of algebraic equations that can be solved with the computer. The solution of these discrete algebraic equations represents an approximation of the continuous problem, and several methods have been developed to find the most appropriate discrete representation of the actual continuous equation, like FEM, FDM and FVM.
The application of MOC is limited to general cases, since spatial variability, slope, surface roughness, and infiltration pattern cannot be adequately characterized. As most of the practical applications do not have analytical solutions, therefore, analysis-based methods, like the MOC are not very useful for such problems.

The Finite Element Method (FEM): This method involves the discretization of the system into a series of sub-domains, called finite elements, connected at a discrete number of nodal points. Thereafter, each of the dependent state variables is approximated in terms of the unknown values and the known shape function at the nodal points. Some of the work using these methods can be seen in the literature by Lee et al. (1987), Cockburn et al. (1989), Schwab (1998), Berzins (2001) and Sheu et al. (2003). The most attractive feature of the FEM is its ability to handle complex geometries (and boundaries) with relative ease.

The limitation of the method is that every improvement in the predictive ability implies a corresponding increase in number of nodal points to be handled, which consequently increases the number of points for solution. This always results in increased time for efficient computations, particularly when the variable surface properties and time-varying rainfall intended in this study are used.

Finite difference method (FDM): FDM is a numerical method for approximating the solutions to differential equations using finite difference equations to approximate derivatives. This of the most frequently used methods to solve PDEs. Some work using this method can be seen in the literature by Courant et al. (1952), Roe (1981), Smith (1985), Carpenter et al. (1993) and Rasulov et al. (2004). There are three basic steps in the application of this method to differential equations (Singh, 1996):

1. The continuous solution domain is discretized and replaced by a grid point called the finite-difference mesh.

2. The continuous derivatives of the differential equation are replaced by finite differences on the grid points, thus the solution equations, their variables and coefficients are established at all grid points (nodes).

3. In the final step, for each node, all the equations are solved using the values of the dependent variable given by the initial and boundary conditions. This set of solutions is then used as initial and boundary conditions when the solution at the next time step is desired.
The most attractive feature of FDM is that it is very easy to implement. However, the main problem with the FDM is that it is not flexible enough to deal with the irregular boundaries.

**Finite volume method (FVM):** In this method continuous equations are discretized into a number of finite volumes. In each of these finite volumes, the integral equations are applied to obtain the exact conservation within each element (also known as cell or grid). It is particularly useful in cases like hydrodynamic modelling, where the equation is solved based on the principle of conservation of momentum. By the discretization of the integral form of the conservation equation, the mass and momentum remain conserved. The resulting expression in the FVM solution appears similar to FDM depending on the techniques applied. As such, it is often considered as a FDM applied to the differential conservative form of the conservation law in arbitrary coordinates Ajaye (2004). The method can be applied also using an unstructured grid system as FEM, but will generally require less computational effort than FEM.

The main advantage of FVM is that it combines the simplicity of FDM with the geometric flexibility of FEM (Mingham & Causon, 1998). Lomax *et al.* (1999) stated that FVM have become popular in CFD as a result, primarily, of two advantages:

1. They ensure that the discretization is conservative, i.e., mass, momentum, and energy are conserved in a discrete sense. While this property can usually be obtained using a finite difference formulation, it is obtained naturally from a finite volume formulation as well; and

2. Finite volume methods do not require a coordinate transformation in order to be applied on irregular meshes. As a result, they can be applied on unstructured meshes consisting of arbitrary polyhedra in three dimensions or arbitrary polygons in two dimensions. This increased capability can be used to great advantage in generating grids about arbitrary geometries.

In this work, FVM is used to solve the hyperbolic PDEs. For details, why this method is adopted, see Jha (2006). In the following section this method is discussed in detail for hyperbolic PDEs.
2.2.7 Finite Volume Method for Hyperbolic Problems

The main purpose of FVM is to estimate the normal flux through each cell interface. Several algorithms are available to estimate these fluxes. Valiani et al. (1999) found that hyperbolic problems have an inherent directional property of signal propagation, therefore algorithms to estimate these fluxes must handle this property appropriately. In FVM the mass and momentum are conserved in each cell, even in the presence of flow discontinuities. The fluxes are evaluated at these cell faces by solving a Riemann problem, which accurately captures wave propagation and shocks generated in the system. Also, numerical oscillations due to sudden jumps (shocks) in the state parameter may be suppressed with the use of a slope limiter (Bradford & Sanders, 2002). Two principal ingredients in a FVM are reconstruction and flux evaluation.

- Reconstruction is the process of computing the spatial gradients of the flow variables.

- In structured methods, this is done using finite difference formulas and the \((i, j, k)\) grid indices. Whereas, in unstructured methods, other techniques have been devised to compute the gradient of the flow variables.

One successful technique is the least-square reconstruction algorithm proposed by Barth (1993), where the flow gradients in a cell are computed through a least-square fitting procedure. Although this technique has been used quite successfully for the Euler and Reynolds-averaged Navier-Stokes (RANS) simulations, it has not yet been applied for Large Eddy Simulation (LES) or Detached Eddy Simulation (DNS) of turbulent flows (Trong, 2000).

Generally, in FVM a flux formula is needed to compute a single flux at a cell boundary given the two different flow states on the left and right sides of the cell boundary. If the physical propagation of information of the inviscid flow is taken into account in computing the inviscid flux, then this results in a family of numerical methods known as upwinding (Trong, 2000). In recent years, finite volume methods have attracted wide attention and achieved a series of successes in the numerical simulation of hyperbolic problems. Alcrudo & Garcia-Navarro (1993) developed a high-resolution Godunov-type MUSCL1-FVM using Roe’s Riemann solver and reported impressive results for rapidly varying inviscid flow. Zhao et al. (1994) used FVM on unstructured meshes based on
Osher’s scheme. Anastasiou & Chan (1997) presented a Roe-type second-order accurate upwind FVM on unstructured triangular meshes. Following Harten-Lax-van Leer (HLL) Riemann solver, Hu et al. (1998) developed a HLL-type MUSCL-FVM. Tseng (1999) proposed an explicit FVM, which takes Roe, total variation diminishing (TVD) and essentially non-oscillatory (ENO) method as the special case. A composite FVM on unstructured triangular meshes was advanced by Ji-Wen & Ru-Xun (2000).

There has been a huge development in the field of FVM and hyperbolic problems in the past few decades. It is a very difficult to present the whole work. Therefore, some of the literature is presented below to show the variety of work done in this area.

Sonar (1997) presented essentially non-oscillatory (ENO) finite volume methods on conforming triangulations for the numerical solution of hyperbolic conservation laws. Besides theoretical results concerning the recovery of data from cell averages, they gave a description of practicable algorithms for the stencil selection to recover polynomials of arbitrary degree. Through extensive numerical tests, the accuracy of the methods was confirmed.

Zhang et al. (2000) suggested a posteriori error estimation technique for hyperbolic conservation laws. The error distributions were obtained by solving a system of equations for the errors which was derived from the linearized hyperbolic conservation laws. The error source term was estimated using the modified equation analysis. Numerical tests for one-dimensional linear and non-linear scalar equations and systems of equations were also presented. The results demonstrated that the error estimation technique can correctly predict the location and magnitude of the errors.

Ghidaglia et al. (2001) proposed a method for the discretization of non-linear systems of PDEs occurring in the numerical simulation of two phase flows. This method was based on a cell centered finite volume discretization on unstructured meshes. They were able to consider conservative and non-conservative systems of equations and the method belonged to the class of shock-capturing upwind ones.

Souadnia et al. (2005) used the general multi-material formulation of Kashiwa & Rauenzahn (1994) to derive an hydrodynamic model for a trickle-bed reactor operating under trickling flow conditions. This kind of trickle-bed reactor has applications in many fields e.g., bio-industry, electrochemical industry and remediation of underground

\[1\text{MUSCL stands for Monotone Upstream-centred Scheme for Conservation Laws}\]
Chapter 2: Literature Survey

water resources other than the traditional chemical, petrochemical and petroleum industries. For the model they used the FVM and the second order Godunov's method combined with the solution of Riemann problem.

Kroger & MLukacova-Medvid'ova (2005) proposed a finite volume evolution Galerkin (FVEG) scheme for the shallow water magnetohydrodynamic equations. This was one of the first attempts to apply multidimensional Evolution Galerkin (EG) techniques to a magnetohydrodynamic model.

Calle et al. (2005) proposed a stabilization technique and studied for the discontinuous Galerkin method applied to hyperbolic equations. In order to avoid the use of slope limiters, a streamline diffusion-like term was added to control oscillations for arbitrary element orders. The scheme combines ideas from both the RungeKutta discontinuous Galerkin method (Cockburn & Shu, 2001) and the streamline diffusion method (Brooks & Hughes, 1982).

Rossmanith (2006) presented an explicit FVM for solving general hyperbolic systems on the surface of a sphere. Applications where such systems arise include passive tracer advection in the atmosphere, shallow water models of the ocean and atmosphere, and shallow water magnetohydrodynamic models of the solar tachocline. This approach used TVD wave limiters, which allowed the method to be accurate for both smooth solutions and solutions in which large gradients or discontinuities can occur in the form of material interfaces or shock waves.

Nordstro & Gong (2006) proposed a stable hybrid method for hyperbolic problems that combines the unstructured finite volume method with high-order finite difference methods. The coupling procedure was based on energy estimates and stability was guaranteed.

2.3 High Resolution Methods

In fluid dynamics problems, discontinuities usually do not develop from smooth initial conditions; except in cases of breaking waves, such as the formation of hydraulic jumps that evolve in the shallow-water flows from smooth initial data. For instance in mid-latitudes, fronts can be formed in low-pressure systems, yet these fronts are not entirely discontinuities. Atmospheric fronts (also substances such as chemical pollutants) are transported from one location to another, described very well by a tracer
advection model. Due to the deformation (stretching and shearing) of the velocity field that advects the front, discontinuous can be formed on the resolution scale of the (computational) model (Durran, 1999). Therefore, from a purely computational standpoint, there is a need to apply numerical schemes devised for numerical solutions of conservation laws which support discontinuous solutions.

In recent years, a tremendous amount of research has been done in developing and implementing modern high-resolution methods for approximating solutions of hyperbolic systems of conservation laws. A review of such numerical methods can be found in the books by Godlewski & Raviart (1996) and LeVeque (2002).

Rood (1987) provided a detailed analysis and comparison of various advection schemes on a simple linear atmospheric transport model. Lin et al. (1994) have analyzed the effect of varying the slope limiters using an atmospheric general circulation model. Lin & Rood (1996) compared the first order upwind, central difference, PPM (modified monotonic and positive definite) and monotonic vanLeer schemes, and concluded that their monotonic version (vanLeer, 1977b) of PPM yields the most accurate results. Towards the development of a fully operational atmospheric general circulation model based on FV discretization, Lin & Rood (1997) implemented slope limited vanLeer schemes and the PPM scheme on a shallow water equation model using a semi-Lagrangian semi-implicit time integration scheme. For a discussion and applications of other popular schemes such as MPDATA of Smolarkiewicz (1984) and QUICK of Leonard [Leonard (1979) and Leonard (1991)], see Vukicevic et al. (2001).

Some of the most popular methods in the finite volume context are Lax-Wendrof, Lax-Friedrichs, Roes, flux corrected transport methods of Boris-Book and Zalesak, slope limited methods of van Leer, piecewise parabolic method (PPM) of Colella and Woodward essentially non-oscillatory schemes of Harten-Shu-Osher to name a few [LeVeque (2002), Laney (1998), Durran (1999)].

There are two ways of implementing the high resolution methods i.e., using structured and unstructured meshes. In the case of irregular boundaries the structured and unstructured mesh use different techniques:

- Structured mesh algorithms employed generally involve complex iterative smoothing techniques that attempt to align elements with irregular boundaries or physical domains. Where non-trivial boundaries are required, block-structured tech-
niques can be employed which allow the user to break the domain up into topological blocks.

- Unstructured mesh generation, on the other hand, relaxes the node valence requirement, allowing any number of elements to meet at a single node. Triangle meshes are most commonly thought of when referring to unstructured meshing, although quadrilateral and hexahedral meshes can also be unstructured.

2.3.1 High Resolution Methods for Structured Meshes

In the last two decades the CFD community has done a huge effort to develop robust high resolution methods for the simulation of advection-dominated flows (Darwish & Moukalled, 2003). Many of these methods have been implemented on structured mesh/grid within the framework of finite volume methods. The main ingredients common to all these methods are a high order profile for the reconstruction of cell face values from cell averages, combined with a monotonicity criterion.

The high order reconstruction is usually based on an upwind biased, sometimes symmetric, high order interpolation profile [Leonard (1979), Leonard (1991) and Holnicki (1996)]. To satisfy monotonicity, a number of concepts have been proposed over the years (LeVeque, 2002), most of them are within a structured mesh/grid framework. In the Flux Corrected Transport (FCT) approach of Book et al. (1981), a first order accurate monotone scheme was converted to a high resolution scheme by adding limited amounts of anti-diffusive flux. In the monotonic upstream-centered scheme for conservation laws (MUSCL) of vanLeer (1979), monotonicity is enforced through a limiter function applied to a piecewise polynomial flux reconstruction procedure. Harten & Hyman (1983) expressed monotonicity as a measure of discrete variation in the solution fields, hence the name Total Variational Diminishing (TVD). This criterion was then expressed as a flux limiter by Sweby (1984) using the $r - \psi$ diagram. Leonard (1991) presented his monotonicity criterion using a relation between a normalized face value and a normalized upwind value. While on the conceptual level the above-mentioned monotonicity criteria can be shown to be related and sometimes equivalent, but implementation-wise they are very different. However within the framework of structured grids these differences have not translated into increased difficulties in implementation.
While there is certainly some overlap between structured and unstructured mesh generation technologies, the main feature which distinguish the unstructured grids is the advantage of generality in that they can be made to conform to nearly any desired geometry.

2.3.2 High Resolution Methods for Unstructured Meshes

For unstructured meshes the situation is more complicated and high resolution methods are not as advanced as for structured meshes [Venkatakrishnan (1996) and Fursenko et al. (1993)]. This is specifically due to the difficulty in implementing and enforcing a monotonicity criterion that relies on logical or directional next-neighbor information, which is readily available in structured meshes but missing in unstructured meshes. To overcome this difficulty a number of approaches have evolved, with varying degrees of success, based on different monotonicity criteria, such as the FCT [LeVeque (2002), Zalesak (1979), Boris (1973) and Boris & Book (1976]), the flux difference splitting concepts [Lohner et al. (1987) and Ferzoui & Stoufflet (1989)], or the MUSCL approach [Venkatakrishnan & Barth (1989) and Frink (1992)].

The MUSCL-based technique developed by Barth and Jespersen (BJ) (Venkatakrishnan, 1994), by modifying the Spekreijse (1987) definition of monotonicity to bound the cell face values rather than the cell nodal value, was one of the most popular and successful approaches for the implementation of high resolution methods in unstructured meshes [Venkatakrishnan (1993), Anderson (1994) and Swanson et al. (1998)], partly because of its simplicity. Unfortunately, most of the limiters developed for structured meshes cannot be implemented using the BJ technique as it is restricted to schemes where the base high order profile uses a cell based gradient, which is basically equivalent to the FROMM scheme (vanLeer, 1979), whose bounded version is equivalent to the MUSCL scheme. In one-dimension the BJ method can be shown to be equivalent to the TVD-MUSCL scheme (Bruner, 1996). Bruner (1996) suggested a more general approach to bound convective schemes. In this approach he used the Sweby $r - \psi$ diagram with a modified $r$ factor defined for unstructured grids. Unfortunately his modification did not recover the exact $r$ factor on structured grids.

Many limiters have been developed and used for the hyperbolic problems with different level of success. A few of the well known limiters that are used for two-dimensional unstructured meshes are:
• Minmod Limiter: The minmod flux limiter applies the maximum possible limiting allowed within the second order TVD region. Anastasiou & Chan (1997) stated the Minmod limiter produces satisfactory results, but numerical diffusion was evident, as is always the case with any dissipative algorithm with a limiter. It is also used in several pieces of literature, few of them are by Sweby (1984), Hirsch (1990), Roe (1992), Wang et al. (2000), Caleffi et al. (2003) and Lipnikov and Shashkov (2006).

• Superbee Limiter: The superbee limiter applies the minimum limiting and maximum steepening possible to remain TVD. It is known to suffer from excessive sharpening of slopes as a result. It is also used in several literature, few of them are by Roe and Sidilkover (1992), Arora and Roe (1997) and Szpilka and Kolar (2003).

• LCD Limiter: The LCD (Limited Central Difference) limiter is one of the earliest limiters in the context of MUSCL-schemes. This limiter’s advantages lie in its simplicity and speed. Batten et al. (1996) and Vollmer (2003) have used the LCD limiter.

• MLG Limiter: It stands for Maximum Limited Gradient, developed by Batten et al. (1996). Author has shown that it reduces to Roe’s superbee limiter in one dimension. Vollmer (2003) found that it is less diffusive than the LCD limiter, but Wang and Liu (2002) states that this advantage of minimum numerical dissipation is expensive to compute.

2.3.3 Numerical Methods for High Resolution Shock Capturing

Modern numerical ideas of shock capturing for computational fluid dynamics can date back as early as 1944 when von Neumann proposed a new numerical method, a centered difference scheme, to treat the hydrodynamical shock problems, for which numerical calculations showed oscillations on mesh scale (Lax, 1986). von Neumann’s dream of capturing shocks was first realized when von Neumann & Richtmyer (1950) had the ingenious idea of adding to the hydrodynamic equations a numerical viscous term of the same size as the truncation error. Their numerical viscosity guarantees that the scheme is consistent with the Clausius inequality, the entropy inequality. The shock
jump conditions i.e., the Rankine-Hugoniot jump conditions, are satisfied provided that
the equations of fluid dynamics are discretized in conservation form. Then oscillations
were eliminated by the judicious use of the artificial viscosity; solutions constructed by
this method converge uniformly except in a neighborhood of shocks, where they remain
bounded and are spread out over a few mesh intervals (Chen, 2000).

The analytical idea of shock capturing i.e., vanishing viscosity methods, is quite old. Chen (2000) states that good numerical schemes should be numerically simple,
robust, fast, and low cost, and have sharp oscillation-free resolutions and high accuracy
in domains where the solution is smooth. It is also desirable that the schemes capture
contact discontinuities, and are coordinate invariant, among other things. For the one-
dimensional case, examples of success include the Lax-Friedrichs scheme (Lax, 1954),
the Glimm scheme (Glimm, 1965), the Godunov scheme (Godunov, 1959) and related
high order schemes e.g., vanLeer’s MUSCL (vanLeer, 1979), Colella-Woodward’s PPM
(Colella & Woodward, 1984), Harten-Engquist-Osher-Chakravarthy’s ENO (Harten
et al., 1987), and the Lax-Wendroff scheme (Lax & Wendorff, 1960) and its two-step ver-
sion, the Richtmyer scheme (Richtmyer & Morton, 1967) and the MacCormick scheme
(MacCormack, 1964).

The most common approach to shock capturing is to first develop a one-dimensional,
total-variation-diminishing (TVD) upwind scheme for a scalar conservation law and
then apply it to systems using one-dimensional characteristic decompositions or ap-
proximate Riemann solvers. Upwind schemes have been used very successfully for
gas dynamical calculations, where the Riemann problem can be solved exactly and
many approximate Riemann solvers are available (Tai et al., 2002). Tracking shocks,
especially when and where new shocks arise and interact in the motion of fluids, is
scientifically extremely important but numerically burdensome. The main motivation
in developing numerical shock capturing algorithms is to treat the shock problem in
fluids (Chen, 2000).

Some of the classical work on development of high-resolution shock-capturing schemes
can be found in Glaz & Liu (1984), Glimm et al. (1985), Chen et al. (1993) and Canuto
et al. (1998), or in the textbooks, Glass (1974), Chainais (1999), Toro (1999) and
LeVeque (2002).
Chapter 2: Literature Survey

2.4 Shallow Water Equations

In recent years free-surface flow models have been increasingly developed using explicit schemes, vis-a-vis implicit schemes (see Fennema & Chaudhry (1990), Nujic (1995) and Chan & Anastasiou (1999)). One of the main reasons for using the explicit scheme is that the resulting numerical models behave better when used to simulate flows with sharp gradient free surfaces, such as dam break flow. Several techniques have been published in the literature concerning the use of the finite volume method to solve the two-dimensional shallow water equations to model free surface flows. Zhao et al. (1994) and Zhao et al. (1996) used three-types of Riemann solvers, including the flux vector splitting, the flux difference splitting and the Riemann solver of Osher, to simulate shock wave problems. The model was based on the finite volume method and used 4-sided grids. Numerical tests showed that all of these three schemes were capable of predicting both gradually and rapidly varying flows, including those with sharp surface gradients. However, since these schemes were first order accurate, the effect of numerical damping was observed when studying model predicted surface profiles. Anastasiou & Chan (1997) and Chan & Anastasiou (1999) developed a finite volume scheme based on a Godunov-type second-order upwind formulation to solve incompressible flows, both with and without a free surface and using an unstructured triangular mesh. Mingham & Causon (1998) developed a high resolution finite volume scheme using a MUSCL reconstruction and with a slope limiter to capture surface discontinuities. Their model was applied to simulate bore wave diffraction in both internal and external hydraulic flows.

2.4.1 Riemann Problem

Prediction of flows with discontinuities, such as hydraulic jumps, has been a great challenge to numerical schemes for hydrodynamics Benedict et al. (2003). Extensive research has been performed in the area of shallow-water equations, and several upwind methods originally designed to solve the Euler equations have been extended to the shallow-water system (Mohamadian et al., 2005). These schemes are introduced for the solution of the hyperbolic sets of equations in order to take into account the information about the direction of signal propagation, enclosed in this class of equations. One method is based on the solution of the local Riemann problem at cell interfaces.
This approach was proposed by Godunov (1959), so the derived schemes are called Godunov-type. In the well known work of Godunov, the exact solution of the Riemann problem was used. Today the exact solution of the Riemann problem is replaced with an approximate solution in order to reduce computational time; this type of scheme is called Flux Difference Splitting (FDS) (Caleffi et al., 2003). In the last twenty years, many efforts have been made by several researchers in the field of approximate Riemann solvers. The more remarkable results are obtained in aerodynamics, but are also easily applicable to the shallow water equations. In fact, a strong analogy between compressible flow and free surface shallow water flow exists (Liggett, 1994).

Many approximate Riemann solvers now exist to evaluate numerically convective fluxes. Those are for example, vanLeer (1977b), Roe (1981), Osher & Solomon (1987), Harten & Hyman (1983), and later on extended to free surface hydraulics in several papers, including those by Glaister (1988), Alcrudo & Garcia-Navarro (1992), Alcrudo & Garcia-Navarro (1993), essentially non oscillatory (ENO) schemes (Nujic, 1995), the Harten, Lax and van Leer (HLL) solver (Mingham & Causon, 1998) and Valiani et al. (2002). Most of these methods have the capability of shock capturing with a high level of accuracy in few computational cells, and the flux vector is determined based on the wave propagation structure.

Some of these methods perform well for particular types of flow like discontinuous or transcritical flows over flat topographies, but in the case of flows over variable topography there is room for considerable improvement. The advantages of using Riemann solvers to describe rapidly varying shallow water flows became apparent in the early 1990s. The shallow-water equations describe the conservation of mass and momentum in shallow water bodies, and are particularly amenable to solution by finite volume Godunov-type approaches where Roe’s approximate Riemann solver can be used to evaluate inviscid fluxes [Alcrudo & Garcia-Navarro (1993), Anastasiou & Chan (1997) and Fujihara & Borthwick (2000)].

Although Roe’s approximate method is robust, difficulties arise in solving the Riemann problem when source terms are included in the analysis (Gascon & Coberan, 2001). Essentially, a numerical imbalance is created by the artificial splitting of physically meaningful terms in the governing equations between flux gradients and source terms in order to generate a mathematically hyperbolic formulation. These terms are
then evaluated by different methods at different locations within the computational grid creating the numerical imbalance.

2.4.2 Shallow flow over Mobile Bed

Flood waves not only involves the clear water flow but also the flow induced sediment transport and morphological changes, which in turn effects the flow hydraulics. Most of the shallow flow studies are carried out over fixed bed with the exception of few over mobile bed. Below are few of the studies carried out over mobile bed cases.

Fraccarollo & Armanini (1998) presented a theoretical analysis of the Riemann problem for a free-surface flow over a movable bed. In their study, they devoted the attention to the reciprocal interactions between the current and the bed in highly transient conditions. A fully coupled model, where mass and momentum exchanges due to the sediment fluxes were taken into account, was employed. The equilibrium concept was assumed for the evaluation of the sediment transport in the flow.

Fraccarollo & Capart (2002) presented a Riemann wave description of erosional dam break flows. In this theory, they separated the flow into an upper clear water sub-layer and a lower mixed water-sediment sub-layer (sediment concentration is fixed, 0.5 in volume), both with the same velocity. The shortcoming of this approach is that the sediment size will have no effect on the on the flow and the bed morphology.

Pritchard & Hogg (2002) presented analytical solution for suspended sediment transport due to dam-break. They explicitly calculated the suspended sediment concentration, including erosion and deposition, and investigated the effect of varying the erosional and depositional models employed. In their model, the velocity field derived from the very idealized clear water flow over frictionless and rigid bed is assumed. The strong interaction between flow, sediment and morphological evolution is ignored.

Cao et al. (2002) presented the development of a fully coupled model and identified the impacts of simplifications in the water-sediment mixture and global bed material continuity equations as well as of the asynchronous solution procedure for aggradation processes. They found in their study that simplifications in the continuity equations for the water-sediment mixture and bed material are found to have negligible effects on degradation. This is, however, in contrast to aggradation processes, in which the errors purely due to simplified continuity equations can be significant transiently.
Cao et al. (2004) presented a non-capacity model for dam-break flow, sediment transport and morphological evolution. The model is based on one-dimensional shallow flow equations together with existing formulations for the bed friction and sediment exchange between the water column and the bed. This model is restricted to one dimensional flow in a channel with rectangular cross sections of constant width and mobile bed composed of uniform and non-cohesive sediment.

2.4.3 Source Terms

The imbalance problem is particularly acute for the shallow water equations where the surface gradient term within the momentum equations is conventionally split into an artificial flux gradient and a source term that includes the effect of the bed slope. Thus, many numerical solvers of the shallow water equations based on the conventional formulation give nonphysical results for flows over physically realistic variable bathymetries, solely because of this mathematically convenient splitting.

Bermudez & Vazquez (1994) extended the vanLeer’s Q-scheme for variable topographies by using an upwind discretization of the source terms and they introduced the C property, which states that the scheme should preserve the stagnant conditions.

Nujic (1995) used the water level variable instead of the depth and extracted the gravitational terms from flux functions. He proposed a revised mathematical formulation of the shallow water equations, by reallocating all bed-slope related flux gradients to the source terms. Then, using the Shu and Osher (SO) scheme (Shu & Osher, 1988), computed the flux vector and obtained good results for dam break problems over variable topographies.

Ambrosi (1995) noted that the effectiveness of using Roe’s approximate Riemann solver was lost when the bottom slope varied, giving a quantitative estimation of the error of the scheme as first-order, but accepted that the quiescent still water solution was not computed, in favor of preserving the formal accuracy of the scheme.

LeVeque (1998) introduced a Riemann problem inside a cell for balancing the source terms and the flux gradients, and proposed a wave propagation algorithm by artificially introducing another discontinuity within each computational cell to account for the propagation of source terms. The resulting method was found to preserve both stagnant and quasi-steady state conditions. Although suitable for quasi-steady conditions, LeVeque’s method is reportedly less robust when predicting steady transcritical flows.
that contain shocks (Benedict et al., 2003). Also, LeVeque’s scheme is not directly transportable to unstructured grids.

Vazquez-Cendon (1999) used numerical upwinding of the source terms to achieve equilibrium between flux gradient and source terms in the shallow water equations. Hubbard & Garcia-Navarro (2000) and Garcia-Navarro & Vazquez-Cendon (2000) have since extended this numerical treatment to higher order total variation diminishing (TVD) schemes.

Zhou et al. (2001) introduced the surface gradient method and showed that interpolating the depth without considering the bed variations may lead to erroneous results. They used the interpolated water surface elevation to calculate the depth at the interface and showed that this approach combined with the HLL flux function (Harten & Hyman, 1983) satisfies the $C$ property. Their scheme performs very well for variable topographies without any extra efforts for balancing the source terms and the flux gradients. However, the $C$ property does not hold for unstructured grids and moreover, the HLL flux induces a high level of numerical viscosity in recirculating flows.

Burguete & Garcia-Navarro (2001) investigated different explicit schemes and presented conservative schemes in a non-conservative formulation of the equations with flux-adjusted source terms discretized using either a semi-implicit or upwinding technique. Gascon & Coberan (2001) present another approach to deal with the balancing difficulty by transforming non-homogeneous conservation laws into homogeneous ones by introducing a new flux generated by the addition to the physical flux of the primitive of the source term.

Alcrudo & Benkhaldoun (2001) defined the topography such that a sudden change in the topography occurs at the interface of two cells. They also developed a Riemann solver at the interface with a sudden change in the bed elevation. However, their approach leads to several cases of Riemann fan and it is numerically too expensive.

Vukovic & Sopta (2002) extended the ENO and WENO schemes to shallow-water equations with the source terms in 1D channels.

Xu (2002) proposed a second order gas kinetic scheme for shallow-water flows over variable topographies. The gas kinetic schemes are basically different from characteristics based schemes.

Mohamadian et al. (2005) has presented an efficient numerical method for recirculating flows over variable topographies on unstructured grids. In order to fulfill
Chapter 2: Literature Survey

this goal, Nujic’s method (Nujic, 1995) is combined with the Roe’s scheme, and the surface gradient method (Zhou et al., 2001) is used for calculating the water depth at the interface.

Murillo et al. (2007) presented a revised CFL condition that takes into account the presence of source terms in the equations for both scalar case and for system of equations. They found that the time step can be increased or decreased depending on the contribution of the source terms. In case, the time step needs a reduction, it must be reduced to ensure stability.

Burguete et al. (2008) presented the numerical limitation of the friction term. They introduced the maximum allowable value of friction force, that can be defined as the maximum force that could prevent the water movement. The concept of maximum allowable friction force was introduced at discrete level making sure that the scheme remains conservative and stable.

2.4.4 Wetting and Drying

Wetting and drying is the most problematic issue in the field of shallow water flow these days. Below some of the work done in this area is outlined.

Falconer & Owens (1987) used a quadrilateral grid. For the wetting and drying fronts he applied a check on four sides of each cell. The normal components of velocities are set to zero in those sides in which the water depth has become negative. If this is the case for all four sides, then the cell is removed from computations. Furthermore, in this approach a grid cell is removed from computations when the water level within it falls below a critical value.

Toro (2001) proposed the artificial wetting of the dry bed by setting the water depth over the dry bed by some small positive tolerance. By doing this, the solution contains a relatively weak propagating shock of speed in the direction of the dry front. Also this technique is not conservative.

Bradford & Sanders (2002) suggested the averaging of data in partially filled cells at a wet/dry boundary, which occurs implicitly as a result of solving the integral form of the governing equations. A partially filled cell defined as having at least one but not more than three dry corner nodes. The cell-average depth in such a cell may be very small since it is averaged over the entire cell area and not just the wet portion. The averaging process also wets all faces of that cell, which allows the artificial spreading of
water into adjacent dry cells, diffuses the wet/dry boundary, and further reduces the depth near the boundary.

Brufau et al. (2004) adapted an algorithm that modified the bed slope by enforcing the mass balance in the mass conservation equation. They used only the first order accurate scheme for both the bed slope and free surface.

Jiang & Wai (2005) presented an approach to represent drying and wetting processes in a three-dimensional finite element sigma coordinate model is described. In this approach he introduced a size factor coefficient and water level diffusion term in the conservation equation. Their approach results in a null momentum computation at the dry areas, which can guarantee numerical stability and satisfy the mass and momentum conservation.

Begnudelli & Sanders (2006) presented an algorithm for the wet/dry fronts over variable topography that tracks fluid volume and the free surface elevation in partially submerged cells. In this algorithm, for partially submerged cells, where the free surface elevation is below the centroid of the cell, a new term call VFR (Volume free surface relationship) is introduced that calculates the new depth of water at the center of the cell using the fluid volume and free surface elevation of the cell.

2.5 Object Oriented Modelling

The idea behind object-oriented modelling is that a computer program may be seen as composed of a collection of individual units, or objects, that act on each other, as opposed to a traditional view in which a program may be seen as a collection of functions or procedures, or simply as a list of instructions to the computer. Each object is capable of receiving messages, processing data, and sending messages to other objects. It is very broad subject and each model is different. To develop an object-oriented model, knowledge of object-oriented technologies and self-visualization of the problem is necessary. Therefore, this section is just an overview of some of the work done related to the use of object-oriented modelling.

Crutchfield & Welcome (1993) have built an adaptive mesh refinement algorithm model using a mixture of C++ classes for high organizational levels and Fortran for low level numerical routines. Effectively, the C++ classes encapsulate Fortran routines and
provide a hybrid solution that allows better memory management than pure Fortran coding.

Larsen & Gavranovic (1994) reviewed the state of object-orientation within hydroinformatics and concluded that the Dinosaur effect was taking place where even the average word processor application was getting so large that not only users failed to understand the complexity but that development may well have also stalled due to the ever-increasing code sizes. Deckers (1994) produced a geohydrological information based system using an object-oriented programming approach and both spatial and non-spatial information. The system used a proprietary Geographic Information Systems (GIS) with its own object-oriented language. This system allowed the coupling of the user interface, GIS, tools and databases into a seamless application. Ruland & Rouve (1994) described the integration of hydraulic models into object-oriented GIS using relational databases. GIS data classes were defined for points, chains, area, polygons and topographical objects, to which 1D backwater and water quality models and 2D finite element models were coupled, also written in objects. They found that this enabled easy operation of simulation models at reduced cost and allowed better integration of technologies.

Perla & Ponnambalam (1994) discuss the use of object-oriented programming within a multi-reservoir system simulation framework using Microsoft Foundation Classes. The Microsoft Foundation Classes provide the front end and Windows implementation and C++ classes provide the numerical algorithms. Solomatine (1994) addresses the need for object-orientation within the hydroinformatics community and shows an experimental hydraulic modelling system, HIS (water distribution modelling system). He argues that object-orientation will eventually replace procedural code as code toolboxes arrive, much like programming environments. These toolboxes will not limit object-oriented techniques to just interfaces, GIS and databases, but will spread into the numerical engines and artificial intelligence components and code reuse will then be possible. Finally, the author concludes that the large amount of Fortran code is holding back the use of object-oriented programming techniques.

Solomatine (1996) takes the original Hydraulic Modeling System (HIS) model and adapts and broadens it. In doing so, he has built on his previous work and used the same base classes, extending the hydraulic features and user interfaces within an event
framework. The entire system has been implemented using Pascal with objects within a Windows, icons, menus and pointer interface.

Shane et al. (1996) have produced an object-oriented water resources management system model (PRSYM), written in C++. The software is capable of modelling reservoir storage, different types of routing, power generation, control rules, water quality and linear programming optimization to optimise policy and economic constraints. Kutija (1998) describes the development of an object-oriented model for the solution of free surface flows in dendritic channel networks. The model uses the de Saint Venant equations, solved by the Abbott-Ionescu (Abott & Basco, 1989) implicit finite difference method. This gives alternating $H$ (elevation) and $Q$ (flow) calculating points with the nodes always being $H$ points. The model was designed to handle rectangular cross sections and flow and elevation boundary conditions, solved by a classical dendritic solver (Cunge et al., 1980).

Tomicic & Yde (1998) demonstrated a typical example of legacy procedural software integration across a wide range of models. Using six models: STORMPAC (Rainfall generation); MOUSE (Urban Sewer Modelling); STOAT (Waste water treatment); MIKE 11 (River modelling); MIKE 21 (Coastal Modelling); and SIMPOL (Simplified urban drainage), they progressively integrated the system of models, starting with input/output files, to produce integrated procedures and a common user interface on a multi-processor machine. This was carried out using procedural techniques to couple the various codes together and allow information to be passed between the various models that became fully integrated into the whole. Thus, the modelling system became fixed into a monolith of code highly suited for its task but incapable of being used in other configurations. Alternatively, Cate et al. (1998) took a more flexible approach to model coupling within a framework so that any collection of models reaching a common standard could be coupled in a multitude of common ways. CORBA technology was tried but was found to be not ready for the connection of these models. Whilst they found shared memory and common files adequate for loosely coupled systems, it was not found to be suitable for more closely coupled systems where data and information needed to flow in many directions.

Harvey et al. (2002) promoted the use of models being broken down into smaller components that could be arranged by a model developer into new models without the need for constant rewriting, much like user interfaces are now designed by dragging
buttons onto a form. This would create a component framework rather than a model framework. Alfredsen (2000) shows a good example of code wrapping. Components were divided into river reaches, lakes, reservoirs, etc. Each component was fitted with a standard interface that allowed flow to be taken from the upstream component and then routed to the next component. Using these basic types of component, further sub types were derived for different types of routing and storage. The system does not offer any real interaction as information flow is one way and each component is fired off one after another. Unlike other frameworks, the components can still be used individually.

Tachikawa et al. (2000) and Ichikawa et al. (2000) describe the development of the OhyMoS object-oriented hydrological modelling system which has been applied to Chao Phraya River basin in Thailand. The model was built around a grid system in which channels and links (ports) to neighbouring grid boxes could be made to form a channel network. Flow could therefore be routed through the grid boxes using a routing equation and rainfall added using a rainfall runoff (Xinanjiang) model. The entire model is built around the grid structure that allows different model representations to be selected for different grid boxes, such as HEC-HMS (Hydrologic Engineering Center - Hydrologic Modelling System, U.S. Army Corps of Engineers) where each model has a standardized form, inputs and outputs and can be wrapped in a class.
Chapter 3  

Numerical Scheme

3.1 Introduction

In Riemann2D, finite volume methods are used to solve the hyperbolic system of equations using high resolution shock capturing techniques. In this chapter, first the formulation of finite volume methods for hyperbolic systems are developed which are only first order accurate in space and time. Later in the chapter, these methods are extended to high order methods, accurate both in space and time. The objectives of this chapter are to present:

- the fundamental concepts of finite volume methods for hyperbolic systems; one-dimensional problem followed by two-dimensional problem.
- the problem due to discontinuity, the Riemann problem, and its solution using a variety of different solvers.
- traditional numerical methods and their deficiencies, such as spurious numeric oscillations in the vicinity of shocks, yielding poor accuracy and smeared results.
- methods to overcome the problems mentioned above.
- various slope limiters for unstructured triangular elements.
3.2 Finite Volume Formulation

Finite Volume Method (FVM) is a discretization technique that involves the discretization of continuous equations into a number of finite volumes, known as control volumes. Some of the important features of the FVM as below:

- It can be used on arbitrary geometries, on unstructured triangular meshes.
- The numerical fluxes remain conserved locally from one discretization cell to its neighboring cells, thus ensuring the overall conservation of the system.

In the following sections, finite volume formulation, adapted from LeVeque (2002), for one-dimensional and two-dimensional problems are presented.

3.2.1 The One Dimensional Problem

Let us consider a quadrilateral control volume of dimensions \([x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [t^n, t^{n+1}]\) in the \(x - t\) plane as shown in the figure. The average value of the \(u(x, t)\), the state variable, at the \(i^{th}\) interval is given by

\[
U^n_i \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x, t_n) dx
\] (3.1)

Figure 3.1: Illustration of finite volume method for updating the cell average \(U^n_i\) by fluxes at the cell edges. One-dimensional problem

The integral form of the conservation laws can be written as
Chapter 3: Numerical Scheme

\[
\frac{d}{dt} \int_{\Omega_i} u(x,t) dx = f(u(x_{i-1/2},t)) - f(u(x_{i+1/2},t)) \tag{3.2}
\]

\[
\int_{x_{i-1/2}}^{x_{i+1/2}} u(x,t_{n+1}) dx - \int_{x_{i-1/2}}^{x_{i+1/2}} u(x,t_n) dx = \int_{t_n}^{t_{n+1}} f(u(x_{i-1/2},t)) dt - \int_{t_n}^{t_{n+1}} f(u(x_{i+1/2},t)) dt \tag{3.3}
\]

Rearranging and dividing by \(\Delta x\) gives

\[
\frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x,t_{n+1}) dx = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x,t_n) dx - \frac{1}{\Delta x} \left[ \int_{t_n}^{t_{n+1}} f(u(x_{i-1/2},t)) dt - \int_{t_n}^{t_{n+1}} f(u(x_{i+1/2},t)) dt \right] \tag{3.4}
\]

This equation shows how the value of \(U\) should be updated in each time step. However, the problem with this equation is that the time integral on the right hand side cannot be evaluated as \(u(x_{i \pm 1/2}, t)\) varies with time along the cell interface. That is why we don’t have to work the exact solution and thus numerical method of the form

\[
U_{i}^{n+1} = U_{i}^{n} - \frac{\Delta t}{\Delta x} \left( F_{i+1/2}^{n} - F_{i-1/2}^{n} \right) \tag{3.5}
\]

is used. Where \(F_{i-1/2}\) is some approximation to the average flux along the cell interface at \(x = x_{i-1/2}\)

\[
F_{i-1/2}^{n} \approx \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(u(x_{i-1/2},t)) dt \tag{3.6}
\]

If we can approximate this flux based on the value of \(U^n\), then a fully discrete time marching algorithm can be achieved for the solution of this problem.

In hyperbolic problems the information propagates with finite speed, so it is reasonable to obtain \(F_{i-1/2}\) based only on the values of \(U_{i-1}^n\) and \(U_i^n\), the cell averages on either side of this cell interface. The we might use the formula

\[
F_{i-1/2}^{n} = F(U_{i-1}^n, U_i^n) \tag{3.7}
\]

where \(F\) is some numerical flux function. The method 3.5 can be written as

\[
U_{i}^{n+1} = U_{i}^{n} - \frac{\Delta t}{\Delta x} \left[ F(U_{i}^{n}, U_{i+1}^{n}) - F(U_{i-1}^{n}, U_{i}^{n}) \right] \tag{3.8}
\]
The specific method obtained depends on how we choose the formula $Ϝ$, but in general any method of this type is an explicit method with a three point stencil, meaning that the value of $U_{i}^{n+1}$ will depend on $U_{i}^{n+1}$, $U_{i}^{n+1}$ and $U_{i}^{n+1}$ at the previous time interval. Moreover, it is said to be in conservation form, since it mimics the property 3.4 of the exact solution.

Telescopic property of conservative method says that the sum of the flux differences cancels out except at the extreme edges. Therefore, the summation applied to 3.4, pre-multiplied by $\Delta x$, gives

$$\Delta x \sum_{i=1}^{J} U_{i}^{n+1} = \Delta x \sum_{i=1}^{J} U_{i}^{n} - \frac{\Delta t}{\Delta x} \left( F_{j+1/2}^{n} - F_{j-1/2}^{n} \right)$$

(3.9)

Rearranging equation 3.8 gives

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

(3.10)

The above equation shows that method 3.8 can be viewed as a direct finite difference approximation to the conservation law $u_t + f(u)x$.

Where flux $F$ is a function of the state variable $U$. Therefore equation 3.10 can be written as:

$$U_t + F(U)_x = 0$$

(3.11)

where

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, F(U) = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix}$$

(3.12)

In the above equation $U$ is the vector of the conserved or state variables and $F$ is the vector of fluxes and each of its component $f_i$ is a function of the component $u_i$ of $U$ (Toro, 1999). The Jacobian of the flux function $F(U)$ in the equation 3.11 is given by

$$A(U) = \frac{\partial F}{\partial U} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \ldots & \frac{\partial f_1}{\partial u_m} \\ \frac{\partial f_2}{\partial u_1} & \ldots & \frac{\partial f_2}{\partial u_m} \\ \vdots & \vdots & \vdots \\ \frac{\partial f_m}{\partial u_1} & \ldots & \frac{\partial f_m}{\partial u_m} \end{bmatrix}$$

(3.13)
3.2.2 The Two Dimensional Problem

The one dimensional conservation equation 3.11 can be extended to two dimensional problem by adding the additional flux term in the $y$-direction. So, the two dimensional conservation equation can be written as:

$$ U_t + F(U)_x + G(y)_y = 0 \quad (3.14) $$

The integral form of this equation can be written as:

$$ \frac{d}{dt} \int_V UdV = -\int_{\Omega} H.n d\Omega \quad (3.15) $$

where $V$ is the control volume, $\Omega$ is the boundary of the control volume, $H = (F,G)$ is the tensor of the flux, $n = [n_1, n_2]$ is the outward unit vector normal to the surface $\Omega$, $d\Omega$ is the area of the element and $H.n d\Omega$ is the flux component normal to the boundary $\Omega$.

The cell average of this control volume can be defined as:

$$ \hat{U} = \frac{1}{|V|} \int_V UdV \quad (3.16) $$

where $|V|$ is the volume of the control volume.

$$ \frac{d}{dt} \hat{U} = -\frac{1}{|V|} \sum_{s=1}^{N} F_s \quad (3.17) $$

with fluxes

$$ F_s = \int_{A_s}^{A_{s+1}} [n_1 F(U) + n_2 G(U)] d\Omega \quad (3.18) $$

and $N$ is the number of sides of the control volume.

$$ U^{n+1} = U^n - \Delta t \sum_{s=1}^{N} = 3 \frac{n_1 F(U) + n_2 G(U)] d\Omega}{A} \quad (3.19) $$

$$ F_{i-1/2}^n = F(U^n_{i-1}, U^n_i) = \frac{1}{2} [f(U^n_{i-1}), f(U^n_i)] \quad (3.20) $$

Expression 3.17 forms the basis of semidiscrete numerical methods, in that the right hand side is assumed to be discrete in space, leaving the left hand side continuous in time. Replacing the time derivative by a forward in time approximation, a fully discrete scheme is obtained (Toro, 1999).
Chapter 3: Numerical Scheme

For triangular mesh the fully discrete scheme can be written as:

\[ U_{i}^{n+1} = U_{i}^{n} - \Delta t \frac{A}{A} \sum_{s=1}^{N=3} [n_{1}F(U) + n_{2}G(U)] \] (3.21)

\[ U_{i}^{n+1} = U_{i}^{n} - \Delta t \frac{2}{2\Delta x} [f(U_{i+1}^{n}) - f(U_{i-1}^{n})] \] (3.22)

3.3 Riemann Problem

The numerical flux is calculated at each side of the control volume with two states separated by a discontinuity. The Riemann problem consists of computing the breakup of this discontinuity, which initially separates two arbitrary constant states, i.e., \( L \) (left) and \( R \) (right). The solution to this problem depends on these two constant states. A simplified case of one spatial dimension in which the Riemann problem is the simplest possible initial-value-problem (IVP) of discontinuity for hyperbolic systems is

\[
PDE \quad u_{t} + au_{x} = 0
\]

\[
IC \quad u(x, 0) = u_{0}(x) = \begin{cases} 
  u_{L}, & \text{if } x < 0 \\
  u_{R}, & \text{if } x > 0 
\end{cases}
\] (3.23)

where \( u_{L} \) and \( u_{R} \) are two constant values (see Figure 3.2). Solving this problem helps to obtain information that is used to compute a numerical flux that is used to update the cell average state over a time step. For linear hyperbolic systems, the Riemann

\[ i
\]

Figure 3.2: Illustration of the Riemann problem (a) initial data at \( t = 0 \), consists of two constant states separated by a discontinuity at \( x = 0 \). (b) Solution in the \( x - t \) plane for the linear advection equation with positive characteristic speed \( a \).
Chapter 3: Numerical Scheme

problem is easily solved in terms of the eigenvalues and eigenvectors. In numerical method, Riemann problems are solved using Riemann solvers, discussed in the following sections.

3.3.1 The Riemann Solver of Roe

Roe solver is perhaps the widely used solver, first presented by Roe (1981). The method was then refined by Roe & Pike (1984), Harten & Hyman (1983), Einfeldt et al. (1991) and Dubois & Mehlman (1993). In this section Roe solver is described for a general system of \( m \) hyperbolic conservation laws (adapted from Toro (1999)).

Roe solved the Riemann problem approximately. By introducing the Jacobian matrix

\[ A(U) = \frac{\partial F}{\partial U} \]  

(3.24)

and using the conservation laws

\[ U_t + FU_x = 0 \]  

(3.25)

in the equation 3.23 may be written as

\[ U_t + A(U)U_x = 0 \]  

(3.26)

Roe’s approach replaces the Jacobian matrix \( A(U) \) in the equation 3.26 by a constant Jacobian matrix

\[ \tilde{A} = \tilde{A}(U_L, U_R) \]  

(3.27)

which is a function of the data states \( U_L, U_R \). In this way the original PDEs in equation 3.23 are replaced by

\[ U_t + \tilde{A}U_x = 0 \]  

(3.28)

This is a linear system with constant coefficients. The original Riemann problem 3.23 is replaced by the approximate Riemann problem:

\[ Ut + \tilde{A}Ux = 0 \]

\[ Ux = \begin{cases} U_L, & x < 0 \\ U_R, & x > 0 \end{cases} \]  

(3.29)
which is then solved exactly. The approximate problem results from replacing the
original non-linear conservation laws by a linearized system with constant coefficients
but the initial data of the exact problem is retained.

For a general hyperbolic system of $m$ conservation laws, the Roe Jacobian matrix
is required to satisfy the following properties:

Property (A): Hyperbolicity of the system. $\tilde{A}$ is required to have real eigenvalues
$\tilde{\lambda} = \tilde{\lambda}_i(U_L, U_R)$, which are normally ordered as

$$\tilde{\lambda}_1 \leq \tilde{\lambda}_2 \leq \ldots \leq \tilde{\lambda}_m$$

(3.30)

and a complete set of linearly independent right eigenvectors

$$\tilde{K}^{(1)}, \tilde{K}^{(2)}, \ldots, \tilde{K}^{(m)}$$

(3.31)

Property (B): Consistency with the exact Jacobian

$$\tilde{A}(U, U) = A(U)$$

(3.32)

Property (C): Conservation across discontinuities

$$F(U_R) - F(U_L) = \tilde{A}(U_R - U_L).$$

(3.33)

Property (A) on hyperbolicity is an obvious requirement; the approximate problem
should at the very least preserve the mathematical character of the original non-linear
system. Property (B) ensures consistency with the conservation laws. Property (C)
ensures conservation. It also ensures exact recognition of isolated discontinuities; that
is, if the data $U_L, U_R$ are connected by single isolated discontinuities, then the approx-
imate Riemann solver recognizes this wave exactly. However, this does not mean that
the corresponding approximate Godunov method with the Roe approximate numerical
flux will in general give exact solutions for isolated discontinuities.

The construction of matrices satisfying properties(A)-(C) for general hyperbolic
systems can be very complicated and thus computationally unattractive. For a specific
case of the Euler equations of gas dynamics, Roe (1981) proposed a relatively simple
way of constructing a matrix $\tilde{A}$.
3.3.1.1 The Intercell Flux

Once the matrix $\tilde{A}(U_L, U_R)$, its eigenvalues $\tilde{\lambda}(U_L, U_R)$ and the right eigenvector $\tilde{K}^{(i)}(U_L, U_R)$ are available, one solves Riemann problem (equation 3.29) by direct application of various methods like CIR scheme, Godunov’s method etc. By projecting the data difference

$$\Delta U = U_R - U_L$$

onto the right eigenvectors we can write

$$\Delta U = U_R - U_L = \sum_{i=1}^{m} \tilde{\alpha}_i \tilde{K}^{(i)}$$

from which one finds the wave strengths $\tilde{\alpha}_i = \tilde{\alpha}_i(U_L, U_R)$. The solution $U_{i+1/2}(x/t)$ evaluated along the $t$-axis, $x/t = 0$, is given by

$$U_{i+\frac{1}{2}}(0) = U_L + \sum_{\lambda \leq 0} \tilde{\alpha}_i \tilde{K}^{(i)}$$

or

$$U_{i+\frac{1}{2}}(0) = U_R - \sum_{\lambda \leq 0} \tilde{\alpha}_i \tilde{K}^{(i)}$$

We can now find the corresponding numerical flux. As we have replaced the original set of conservation laws in equation 3.23 by the constant coefficient linear system (equation 3.28); this can be viewed as a modified system of conservation laws

$$\tilde{U}_t + \tilde{F}(\tilde{U})_x = 0$$

with the flux function

$$\tilde{F}(\tilde{U}) = \tilde{A}\tilde{U}$$

The corresponding numerical flux is not the obvious choice,

$$F_{i+\frac{1}{2}} = \tilde{A}\tilde{U}_{i+\frac{1}{2}}(0)$$

where $U_{i+1/2}(0)$ is given by either of equations 3.36-3.37. That this would be incorrect becomes obvious when, for instance, assuming right supersonic flow in (3.36) one
Chapter 3: Numerical Scheme

would compute an intercell flux. Instead, the correct expression for the corresponding numerical flux is obtained from any of integral relations

\[ F_{0L} = F_L - S_L U_L - \frac{1}{T} \int_{TS_L}^0 U(x,T) \, dx \]  \hspace{1cm} (3.41)

\[ F_{0R} = F_R - S_R U_R + \frac{1}{T} \int_0^{TS_R} U(x,T) \, dx \]  \hspace{1cm} (3.42)

Here \( S_L, S_R \) are the smallest and largest signal speeds in the exact solution of the Riemann problem with data \( U_L, U_R \) and \( T \) is a positive time. If the integrand \( U(x,t) \) in equation 3.41 or equation 3.42 is replaced by some approximate solution, then equality of the fluxes \( F_{0L} \) and \( F_{0R} \) requires the approximate solution to satisfy a consistency condition.

If \( \bar{U}_{i+1/2}(x/t) \) is the solution of the Riemann problem for the modified conservation laws 3.38 with data \( U_L, U_R \), then the integrals in 3.41 and 3.42 respectively, are

\[ \int_{TS_L}^0 \bar{U}_{i+1/2}(x,T) \, dx = T[\bar{F}(U_L) - \bar{F}(\bar{U}_{i+1/2}(0))] - TS_L U_L \]  \hspace{1cm} (3.43)

and

\[ \int_0^{TS_R} \bar{U}_{i+1/2}(x,T) \, dx = T[\bar{F}(\bar{U}_{i+1/2}(0)) - \bar{F}(U_R)] - TS_R U_R \]  \hspace{1cm} (3.44)

Substitution of equations 3.43 and 3.44 into equations 3.41 and 3.42 gives

\[ F_{0L} = \bar{F}(\bar{U}_{i+1/2}(0)) + F(U_L) - \bar{F}(U_L) \]  \hspace{1cm} (3.45)

and

\[ F_{0R} = \bar{F}(\bar{U}_{i+1/2}(0)) + F(U_R) - \bar{F}(U_R) \]  \hspace{1cm} (3.46)

Finally by using \( U_{i+1/2}(0) \) as given by equation 3.36 or 3.37 and the definition of the flux \( \bar{F} = \bar{A} \bar{U} \) we obtain the numerical flux \( F_{Roe} \) at the boundary of the element as:

\[ F_{i+1/2} = F_L + \sum_{\lambda \leq 0} \hat{\alpha}_i \bar{\lambda}_i \bar{K}^{(i)} \]  \hspace{1cm} (3.47)

or

\[ F_{i+1/2} = F_R - \sum_{\lambda \geq 0} \hat{\alpha}_i \bar{\lambda}_i \bar{K}^{(i)} \]  \hspace{1cm} (3.48)

Alternatively, we may also write

\[ F_{i+1/2} = \frac{1}{2} (F_L + F_R) - \frac{1}{2} \sum_{i=1}^m \hat{\alpha}_i |\bar{\lambda}_i| \bar{K}^{(i)} \]  \hspace{1cm} (3.49)
The relations 3.38-3.44 are valid for any hyperbolic system and any liberalization of it. In order to compute Roe’s numerical flux for a particular system of hyperbolic conservation laws, one requires expressions for the wave strengths $\tilde{\alpha}_i$, the eigenvalues, and the right eigenvectors in any flux expressions 3.47-3.49. It is important to note that the Jacobian matrix is not explicitly required by the numerical flux.

### 3.3.2 The HLL (Harten-Lax-Van Leer) Solver

The HLL solver was originally developed by Harten et al. (1983) for the Euler equation of gas dynamics but later it was extended to many other equations. The solver is based on estimating the speeds at which the informations/waves propagate away from a Riemann problem. The estimates are based on the initial data and general properties of the exact solutions (George, 2004).

![Figure 3.3: Approximate HLL Riemann solver. Solution in the star region consists of a single state $U^{\text{hill}}$ separated from data states by two waves of speeds $S_L$ and $S_R$.](image)

To derive a relation for the HLL numerical flux, consider a wave structure contained in a control volume $[x_L, x_R] \times [0, T]$ as shown in the figure 3.3 such that

\[ x_L \leq TS_L, \quad x_R \geq TS_R \quad (3.50) \]

The integral form of the conservation equation for this control volume can be written as

\[ \int_{x_L}^{x_R} U(x, T) dx = \int_{x_L}^{x_R} U(x, 0) dx + \int_0^T F(U(x_L, t)) dt - \int_0^T F(U(x_R, t)) dt \quad (3.51) \]
Evaluating the right-hand side of the above equation gives

\[ \int_{x_L}^{x_R} U(x,T)dx = x_R U_R - x_L U_L + T(F_L - F_R) \] (3.52)

Splitting the left-hand side into three integrals and subsequent evaluation results

\[ \frac{1}{T(S_R - S_L)} \int_{T S_L}^{T S_R} U(x,T)dx = \frac{S_R - S_L U_L + F_L - F_R}{S_R - S_L} \] (3.53)

The integral average of the exact solution of the Riemann problem between the slowest and fastest signals is a known constant, represented by the right hand side of the above equation and is denoted by

\[ U^{hll} = \frac{S_R - S_L U_L + F_L - F_R}{S_R - S_L} \] (3.54)

Applying the integral form of the conservation laws to left or right portion of the figure 3.3, that is the control volume \([T S_L,0] \times [0,T]\) or \([0,T S_R] \times [0,T]\), and substitution of 3.54 gives

\[ F^{hll} = F_L + S_L (U^{hll} - U_L) \] (3.55)

or

\[ F^{hll} = F_R + S_R (U^{hll} - U_R) \] (3.56)

Use of 3.54 in 3.55 or 3.56 gives the HLL flux

\[ F^{hll} = \frac{S_R F_L - S_L F_R + S_L S_R (U_R - U_L)}{S_R - S_L} \] (3.57)

The corresponding inter cell flux for the approximate Godunov method is then given by

\[ F_{i+\frac{1}{2}}^{hll} = \begin{cases} 
F_L & \text{if } 0 \leq S_L \\
F^{hll} & \text{if } S_L \leq 0 \leq S_R \\
F_R & \text{if } 0 \geq S_R
\end{cases} \] (3.58)
Chapter 3: Numerical Scheme

3.3.3 The HLLC Approximate Riemann Solver

The HLLC solver is the modification of the HLL solver, whereby the contact and the shear waves are restored. The scheme was first developed for the Euler equations but later, it was applied to many other equations. Figure 3.4 shows the wave structure of the HLLC solver. To develop the scheme, consider the complete structure of the

Figure 3.4: Approximate HLLC solver. In HLLC solver the contact and shear waves are restored.

Riemann problem contained within a control volume, that is \([x_L, x_R] \times [0, T]\), as shown in the figure. In addition to the slowest and fastest signal speeds, the middle wave speed is also included. Evaluating the integral form of conservation laws in the control volume, we obtain

\[
F^*_{L} = F_L + S_L(U_{sL} - U_L) \\
F^*_{R} = F_L + S_R(U_{sR} - U_R)
\]

(3.59)

The HLLC flux for the approximate Godunov method can be written as

\[
F^{hllc}_{i+\frac{1}{2}} = \begin{cases} 
F_L & \text{if } 0 \leq S_L \\
F_{sL} & \text{if } S_L \leq 0 \leq S_s \\
F_{sR} & \text{if } S_s \leq 0 \leq S_R \\
F_R & \text{if } 0 \geq S_R 
\end{cases}
\]

(3.60)

3.4 Numerical Methods

3.4.1 Traditional Numerical Methods

In this section traditional numerical methods are discussed. The advantage of discussing traditional methods is to outline their deficiencies and the need of modern numerical methods.
3.4.1.1 An Unstable Flux

The most simple approach to get the flux at $x_{i-1/2}$ is based on the data $U_{i-1}^n$ and $U_i^n$ to the left and right of this point. In this approach an arithmetic mean is used as

$$F_{i-1/2}^n = F(U_{i-1}^n, U_i^n) = \frac{1}{2}[f(U_{i-1}^n) + f(U_i^n)] \quad (3.61)$$

Using this in equation 3.5 gives

$$U_{i+1}^n = U_i^n - \frac{\Delta t}{2\Delta x}[f(U_{i+1}^n) - f(U_{i-1}^n)] \quad (3.62)$$

This method is generally unstable for hyperbolic problems even if the CFL condition is satisfied.

3.4.1.2 The Lax-Friedrichs Method

The classical Lax-Friedrichs (LxF) method has the form

$$U_{i}^{n+1} = \frac{1}{2}(U_{i-1}^n + U_i^n) - \frac{\Delta t}{2\Delta x}[f(U_{i+1}^n) - f(U_{i-1}^n)] \quad (3.63)$$

This method is very similar to the unstable method 3.62, but the value $U_i^n$ is replaced by the average $\frac{1}{2}(U_{i-1}^n + U_i^n)$. This method is stable for linear hyperbolic problems provided that the Courant number is less than unity. The above equation can be written in terms of numerical flux as:

$$F(U_{i-1}^n, U_i^n) = \frac{1}{2}[f(U_{i+1}^n) + f(U_i^n)] - \frac{\Delta x}{\Delta t}[(U_i^n + U_{i-1}^n)] \quad (3.64)$$

The problem with this method is that it gives badly smeared results unless a very fine grid is used.

3.4.1.3 The Richtmyer Two-Step Lax-Wendroff Method

In this approach the $u$ is approximated at the midpoint in time, $t_{n+1/2} = t_n + \frac{1}{2}\Delta t$, and then evaluate the flux at this point as

$$F_{i-1/2}^{n} = f(U_{i-1/2}^{n+1/2}) \quad (3.65)$$

where

$$U_{i-1/2}^{n+1/2} = \frac{1}{2}(U_{i-1}^n + U_i^n) - \frac{\Delta t}{2\Delta x}[f(U_i^n) - f(U_{i-1}^n)] \quad (3.66)$$
Chapter 3: Numerical Scheme

In this method the Lax-Friedrichs approach is used to obtain $U_{i-\frac{1}{2}}^{n+1/2}$ at the cell interface.

The problem of this method is that it often leads to spurious oscillations in solution, particularly when solving problem with discontinuous solutions.

3.4.1.4 The Lax-Wendroff Method

To solve the linear system $u_t + Au_x = 0$, a large number of second-order methods can be developed. Most of these methods are based on finite difference approximations of the model equation. The Lax-Wendroff method is based on the Taylor series expansion

$$u(x, t_{n+1}) = u(x, t_n) + \Delta t u_t(x, t_n) + \frac{1}{2} (\Delta t)^2 u_{tt}(x, t_n) + ...$$

Rewriting $u_t + Au_x = 0$ as $u_t = -Au_x$ and differentiating gives:

$$u_{tt} = -Au_{xt} = -A^2 u_{xx}$$

Using these expressions for $u_t$ and $u_{tt}$ in equation 3.67 gives

$$u(x, t_{n+1}) = u(x, t_n + \Delta t Au_x(x, t_n) + \frac{1}{2} (\Delta t)^2 A^2 u_{xx}(x, t_n) + ...$$

Keeping only the first three terms on the right hand side and replacing the spatial derivatives by central difference approximations gives the Lax-Wendroff method,

$$U_{i+1}^{n+1} = U_i^n - \frac{\Delta t}{2\Delta x} A(U_{i+1}^n - U_{i-1}^n) + \frac{1}{2} \left( \frac{\Delta t}{\Delta x} \right)^2 A^2 (U_{i+1}^n - 2U_i^n + U_{i-1}^n)$$

By matching three terms in Taylor series and using centered approximations, a second order accurate method is achieved.

This derivation of the method is based on a finite difference interpretation, with $U_i^n$ approximating the pointwise value $u(x_i, t_n)$. However, the above equation can be interpreted as finite volume method of the form 3.21 with the flux function

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

This looks like the unstable averaged flux (3.61) plus diffusive flux, but that the diffusion chosen exactly matches what appears in the Taylor series expansion (3.69). Actually, this shows why the averaged flux (3.61) alone is unstable.
3.4.1.5 The Beam-Warming method

The Lax-Wendroff method (3.70) is a centered three-point method. If we have a system for which all the eigenvalues are positive, then it might be preferable to use a one sided formula. Instead of the centered formula, we may use

\[ q_x(x_i, t_n) = \frac{1}{2\Delta x} [3q(x_i, t_n) - 4q(x_{i-1}, t_n) + q(x_{i-2}, t_n)] + O(\Delta x^2) \]  (3.72)

\[ q_{xx}(x_i, t_n) = \frac{1}{(\Delta x)^2} [q(x_i, t_n) - 2q(x_{i-1}, t_n) + q(x_{i-2}, t_n)] + O(\Delta x) \]  (3.73)

Using these in equation 3.69 gives a method that is again second order accurate,

\[ Q_{i+1}^n = Q_i^n - \Delta t \Delta x A (3Q_i^n - 4Q_{i-1}^n + Q_{i-2}^n) + \frac{1}{2} (\Delta t / \Delta x)^2 A^2 (Q_i^n - 2Q_{i-1}^n + Q_{i-2}^n) \]  (3.74)

This is known as the beam warming method and was originally developed by Warming & Beam (1975). It can be written as a flux-differencing finite volume method with

\[ F_{i-1/2}^n = A Q_{i-1}^n + \frac{1}{2} A (1 - \Delta t / \Delta x) (Q_{i-1}^n - Q_{i-2}^n) \]  (3.75)

3.4.2 Upwind Methods

For hyperbolic problems, information propagates with the waves along its characteristics. In a system of equation, several waves propagate at different speed and may be in different directions. Therefore, it might be a better alternative to use our knowledge of the structure of the solution to find the improved flux functions. Consider the constant-coefficient advection equation

\[ u_t + cu_x = 0 \]  (3.76)

where c is the advection speed.

Figure 3.5 shows that the flux through the left edge is solely determined by the value, \( U_{i-1}^n \), of the left element. Therefore, the numerical flux can be defined as

\[ F_{i-1/2}^n = c U_{i-1}^n \]  (3.77)

This leads to the standard first order upwind method for the advection equation,
Chapter 3: Numerical Scheme

Figure 3.5: Two representations of the upwind method for advection. (a) If $U^n_i$ represents the value at a point where information is available, then we can trace the characteristic back and interpolate. (b) If $U^n_i$ represents the cell average, then the flux at the interface is determined by the cell value on the upwind side.

$$U^{n+1}_i = U^n_i - \frac{c\Delta t}{\Delta x} (U^n_i - U^n_{i-1}) \tag{3.78}$$

This can be written as

$$\frac{U^{n+1}_i}{\Delta t} - U^n_i + c \left( U^n_i - U^n_{i-1} \right) = 0 \tag{3.79}$$

whereas, the unstable method applied to the advection equation is

$$\frac{U^{n+1}_i}{\Delta t} - U^n_i + c \left( \frac{U^n_{i+1} - U^n_{i-1}}{2\Delta x} \right) = 0 \tag{3.80}$$

The upwind method uses one sided approximation to the derivative $u_x$ in place of the centered approximation.

Another interpretation of the upwind method is suggested by the figure 3.5. If $U^{n+1}_i$ is assumed to be the value at the center of the grid points, then $u(x, t)$ is constant along characteristics. Therefore it can be written

$$U^{n+1}_i \approx u(x_i, t_{n+1}) = u(x_i - c\Delta t, t_n) \tag{3.81}$$

If the value on the right side is approximated by a linear interpolation between the grid values $U^n_{i-1}$ and $U^n_i$, we get the method

$$U^{n+1}_i = \frac{c\Delta t}{\Delta x} U^n_{i-1} + \left( 1 - \frac{c\Delta t}{\Delta x} \right) U^n_i \tag{3.82}$$
Chapter 3: Numerical Scheme

This is a simple upwind method and a rearrangement of equation 3.78, but the condition

$$0 \leq \frac{c \Delta t}{\Delta x} \leq 1$$  \hspace{1cm} (3.83)

is applied in order for the characteristics to fall between the neighboring points so that this interpolation is sensible. In fact equation 3.83 must be satisfied in order for the upwind method to be stable and also follows the CFL condition. Also, if equation 3.83 is satisfied then equation 3.82 expresses $U_i^{n+1}$ as a convex combination of $U_i^n$ and $U_{i-1}^n$.

In the above discussion, it was assumed that $c > 0$. If $c < 0$, then the upwind direction is to the right and so the numerical flux at $x_{i-1/2}$ is

$$F_{i-1/2}^n = cU_i^n$$ \hspace{1cm} (3.84)

The two formulae in equation 3.77 and 3.84 can be combined into a single upwind formula that is valid for $c$ of either sign, then we can write

$$F_{i-1/2}^n = c^-U_i^n + c^+U_{i-1}^n,$$ \hspace{1cm} (3.85)

where

$$c^+ = \max(c, 0) \quad \text{and} \quad c^- = \min(c, 0).$$ \hspace{1cm} (3.86)

For equation 3.85, the equivalent two-dimensional expression can be modified as:

$$F_{s}^n = s^-U_R^n + s^+U_L^n,$$ \hspace{1cm} (3.87)

with

$$s^+ = \text{Right going wave} \quad \text{and} \quad s^- = \text{Left going wave}.$$ \hspace{1cm} (3.88)

This information is useful in extending the methods to more general hyperbolic problems. Not all hyperbolic equations are in conservation form; for example the equations of acoustics in a heterogeneous medium (where the density and bulk modulus vary with $x$). Such equations do not have a flux function, and so numerical methods of the form 3.5 cannot be applied. However, these hyperbolic problems can still be solved using the finite volume approach of the model that results from a simple generalization of the high-resolution methods developed for hyperbolic conservation laws. The unifying feature of all hyperbolic equations is that they model waves that travel at finite speed. In particular the Riemann problem with piecewise constant initial data consists
Chapter 3: Numerical Scheme

of waves traveling at constant speeds away from the location of the jump discontinuities in the initial data.

3.4.3 Godunov’s Method

Godunov-type schemes have proved to be powerful tools for simulating discontinuous flows when they are described by systems of non-linear, hyperbolic conservation laws (Woodward & Colella, 1984), and (Toro, 1999). Guinot (2003) describes Godunov-type schemes as a good candidates for the next generation of commercial modelling software packages, the capability of which to handle discontinuous solutions will be a basic requirement. Although they have gained popularity in the research area, such schemes are rarely found in simulation packages available commercially. Such schemes use many ad hoc techniques, such as slope limiters or wave splitting for multidimensional problems. Also, because the stencil of a scheme should include the domain of dependence of the solution for stability, a CFL stability constraint is necessarily attached to fixed-stencil schemes. Therefore, in such schemes, the computational time step has to be limited so the Courant number associated with each of the eigenvalues of the hyperbolic system should often be smaller than a fixed value. Consequently, these schemes should have the capabilities to provide efficient solutions when:

- large contrasts exist between the various eigenvalues, and
- the computational grid is highly irregular.

In these cases, the schemes performance is limited by the largest of the eigenvalues of the hyperbolic system and by the size of the smallest cell, leading to a strong degradation of the numerical solution in regions where the cell is larger.

Guinot (2002) identified various ways to overcome the stability problems, as follows

1. one of them leads to a first class of numerical methods where the governing partial differential equations (PDEs) are approximated with ordinary differential equations (ODEs), thus breaking the dependence of the time step on the cell size. Although this allows the computational time step to be increased, such an approach has the drawback that it only works for smooth solutions to the PDEs.
Chapter 3: Numerical Scheme

2. Another possible approach consists of adapting the stencil of the scheme to the size of the domain of dependence of the solution. The characteristics can be traced forward or backward across several computational cells. This has been the subject of various works in the domain of Lagrangian or semi-Lagrangian techniques (Goldberg & Wylie, 1983) as well as flux-based (or conservative) Eulerian schemes. Originally, the approach consisted of seeking the departure point of the characteristic lines across several cells when needed. It was applied to problems with (almost) constant wave speeds, such as water hammer problems. The accurate determination of the feet of the characteristic lines is not straightforward and has been studied by a number of authors (Savic & Holly, 1993). Moreover, when strong non-linearity is present in the equations, the irreversible character of the solution makes it difficult, if not impossible, to determine accurately the extension of the domain of dependence using backward-tracking algorithms.

3. A third approach consists of making the scheme implicit by solving a set of linear or non-linear equations using the unknown variables at the next time level to be computed. These techniques are time consuming, inasmuch as the size of the system to be solved is equal to the number of computational cells, or to twice this number in some methods.

4. The fourth approach well suited to the solution of conservation laws in the presence of shock waves, consists of using front-tracking-based methods, except that the domain of influence and the potential merging of shocks is explored forward in time rather than backward.

Instead we have used the shock-capturing methods, where the goal is to capture discontinuities in the solutions automatically, without explicitly tracking them.

3.4.4 Godunov’s Method For Triangular Meshes

The upwind method for the advection equation can be derived as a special case of the following approach, which can also be applied to systems of equations. LeVeque (2002) referred it as the REA for reconstruct-evolve-average:

Algorithm 4.1
1. Reconstruct the states in the element, from the element state $U^n$ and define the states at the sides of the triangular control volumes. In the simple case this is a piecewise constant function that takes the value $U_n$ in the element $n$.

2. Evolve the hyperbolic equation exactly (or approximately) with this initial data to obtain $U^n(x, tn + 1)$ at time $\Delta t$ later.

3. Average this function over each grid to obtain new cell averages

This whole process is repeated in the next time step. To introduce this procedure, it must be able to solve the hyperbolic equation in step 2. Because it is starting with piecewise constant data, this can be done using the theory of Riemann problems. When applied to the advection this leads to upwind algorithm.

![Figure 3.6: An illustration of reconstruct-evolve-average algorithm for the case of linear acoustics (adapted from LeVeque (2002)). The Riemann problem is solved at each cell interface, and the wave structure is used to determine the exact solution time $\Delta t$ later. This solution is updated over the elements to determine $U_{n+1}$](image)

In step 1, we reconstruct a state variable from the discrete element state variables in the neighboring elements. In Godunov’s original approach this is the piecewise constant function. It leads most naturally to Riemann problems, but gives only a first order accurate method. To obtain better accuracy, a better reconstruction can be used,
for example a method that helps to find the appropriate slope gradient of the variables for each element. This idea forms the basics of the high-resolution methods.

The exact solution at time $t_{n+1}$ can be constructed by piecing together the Riemann solution, provided that the time step is short enough that the waves from two adjacent sides in this process have not yet started to interact. Figure 3.6 shows a systematic diagram of this process for the equation of linear acoustics with constant sound speed $c$, in which case this requires that

$$|c| \Delta t \leq \frac{1}{2} \Delta x$$

so that each wave goes at most halfway through though the grid cell. Rearranging gives

$$|c| \frac{\Delta t}{\Delta x} \leq \frac{1}{2}$$

The quantity $|c| \frac{\Delta t}{\Delta x}$ is simply the Courant number, so it is apparent that the condition is limited only within $1/2$.

For the triangular element, the equation 3.90 can be modified using CFL condition and rewritten as

$$\frac{\Delta t}{\Delta x} \max_{i=1,2,3} \left( \frac{s_i l_i}{\min(A_0, A - i)} \right) \leq \frac{1}{2}$$

### 3.4.5 Total Variation Diminishing (TVD) Method

Although first-order finite difference methods are monotonic and stable, they are also strongly numerically diffusive, causing the solution to become smeared out. Second-order or higher-order techniques are less dissipative, but susceptible to non-linear, numerical instabilities that cause non-physical oscillations. The high-resolution methods are a compromise between the traditional first-order and higher-order difference schemes. Their central idea is, on the one hand, to avoid the introduction of under- and over-shoots (numerical oscillation), and on the other hand, to maintain the numerical diffusion as small as possible, that is often achieved by different cell reconstruction techniques.

It is well known that in computing discontinuous solutions, the first order method (upwind) gives very smeared solutions while the second order method (Lax-Wendroff or Beam-Warming) gives spurious oscillations. In order to develop a method that is
of higher order and at the same time non-oscillatory and capable of capturing shocks, we need to define a powerful concept called the Total Variation Diminishing (TVD) method.

For a grid function it can be defined as:

$$TV(u^n) = \sum_{i=-\infty}^{\infty} |u^n_i - u^n_{\text{neighbour}}|$$  \hspace{1cm} (3.92)

Any oscillation in the computed result increases the total variation (TV). The Total Variation Diminishing condition

$$TV(u^{n+1}) \leq TV(u^n)$$  \hspace{1cm} (3.93)

provides a method that gives a solution without spurious oscillations near the discontinuities. Any numerical scheme which fulfils the TVD condition (equation 3.93) for all grid functions $U_n$ is called a Total Variation Diminishing (TVD) method. Therefore, any TVD method is automatically monotonicity preserving. This means, in particular, that oscillations of the physical quantities like velocity jumps and other sharp gradients cannot arise near an isolated propagating discontinuity. As we will see later, another beautiful feature of the TVD requirement is that it is possible to derive higher-order accurate methods that also satisfy equation 3.93. It can also be shown that the true (i.e., physically relevant weak) solution to a scalar conservation law possesses this TVD property (LeVeque, 1992).

### 3.4.6 High Resolution Methods for Riemann2D

Much effort has been invested to achieve better than first-order accuracy with finite volume methods. The first hurdle is Godunov’s Theorem, which states that non-oscillatory constant coefficient schemes can be at most first-order accurate (Vollmer, 2003). This can be overcome by the introduction of non-linear schemes such as Weighted Average Flux (WAF) (Billett & Toro, 1997), MUSCL (vanLeer, 1979), ENO (Toro, 1995), Flux Corrected Transport (FCT) (Boris, 1973) and Piecewise Linear (PLM) (Colella & Woodward, 1984). One of the popular ones is vanLeers MUSCL (vanLeer, 1979) approach, which belongs to the class of Godunov-type methods, a class of non-oscillatory finite volume schemes that incorporate the exact or approximate solution to Riemann’s initial-value problem.
VanLeer [vanLeer (1977a), (vanLeer, 1977b) and (vanLeer, 1979)] introduced the idea of modifying the piecewise constant data in the first-order Godunov method, as a first step to achieving higher order accuracy. This approach has become known as the MUSCL or Variable Extrapolation approach. The piecewise constant states within each cell are modified to piecewise linear ones (figure 3.7), which are carefully constructed from neighbouring states both to maintain conservation and not increase total variation.

### 3.4.7 Data Reconstruction

In all second- and higher-order schemes the application of non-linear limiters involves the introduction of a parameter termed the limiter into the gradient terms that appear in the process of cell variable reconstruction. This step is necessary for higher-order schemes in order to maintain monotonicity.

![Figure 3.7: (a) Constant reconstruction vs (b) piecewise linear reconstruction](image)

Given an initial distribution of piecewise constant values in each control volume, a piecewise linear reconstruction of any scalar variable, \( u_0 \), over an element with centroid value as \( u \) may be expressed as

\[
 u'(x, y) = u + \vec{r} \cdot \vec{L} 
\]

(3.94)

Here \( \vec{r} \) is a vector from the centroid of the control volume and \( \vec{L} \) is a gradient operator. The linear reconstruction of \( u_0 \) still has to be conservative over the control volume \( \omega_j \) in the sense that
Batten et al. (1996) recommended to construct a gradient plane through three nearby centroids A, B and C with normal vector

\[ n_P = (P_A - P_B) \times (P_C - P_B) \]  \hspace{1cm} (3.96)

with

\[ P_i = \begin{bmatrix} x_i \\ y_i \\ z_i \\ u_i \end{bmatrix} \]  \hspace{1cm} (3.97)

and the subsequent gradient operator

\[ \vec{L} = \nabla(\Delta ABC) = \begin{cases} \begin{bmatrix} -n_x/n_u \\ -n_y/n_u \\ 0 \\ 0 \end{bmatrix} & \text{if } n_u > \varepsilon \\ \begin{bmatrix} 0 \\ 0 \end{bmatrix} & \text{otherwise} \end{cases} \]  \hspace{1cm} (3.98)

The gradient operator defined by equation 3.98 is not yet limited and as such may exhibit non-physical over- or under-shoots at the points where the operator is evaluated usually at the midpoint of each edge. The limiting of the gradient operator therefore
Chapter 3: Numerical Scheme

plays an important role as it directly influences the character and accuracy of the solution. As a result we can write that the reconstruction of the state variables in the cell is limited in the form:

$$U(x, y) = U_0 + \Phi(\vec{r}, \vec{L})$$

(3.99)

where $\Phi$ is a chosen limiter. When $\Phi$ is set to zero, equation 3.99 is a first-order-accurate reconstruction.

3.4.7.1 No Limiters

"No limiters” means that the values of the state variables at the boundaries of a control volume will be equal to the ones at the center. In this case $\Phi = 0$ and it is rank 0 limiter.

3.4.7.2 Superbee and Minmod Limiters

Superbee and Minmod limiters are rank 1 limiters as they consider only one limiting triangle (see figure 3.8), i.e, $\Delta ABC$. This is defined as

$$\Phi_{Superbee/Minmod} = \min(\Phi_j)$$

(3.100)

where

$$\Phi_j = \max[\min(\beta r_j, 1), \min(r_j, \beta)]$$

(3.101)

with

$$r_j(u_j) = \begin{cases} (u_{0}^{max} - u_0)/(u_j - u_0) & \text{if } u_j - u_0 > 0 \\ (u_{0}^{min} - u_0)/(u_j - u_0) & \text{if } u_j - u_0 < 0 \\ 1 & \text{if } u_j - u_0 = 0 \end{cases}$$

(3.102)

and

$$u_{0}^{min} = \min(u_0, u_{\text{neighbour}}), \quad u_{0}^{max} = (u_0, u_{\text{neighbour}}).$$

(3.103)

In the above equation, $u_{\text{neighbour}}$ is the value of a conserved variable in the elements $A, B$ or $C$. The quantity $\beta$ in equation 3.101 can take any value between 1 and 2. In particular, $\beta = 1$ is used for Mimod limiter and $\beta = 2$ is used for Roe’s Superbee limiter Hirsch (1990).
Chapter 3: Numerical Scheme

3.4.7.3 Limited Central Difference (LCD) Limiter

LCD is one of the earliest and still most widely used limiters in the context of MUSCL-schemes (Vollmer, 2003). This is rank 1 limiter as it considers only one limiting triangle (see figure 3.8) i.e., $\Delta ABC$. This limiter’s advantages lie in its simplicity and speed. It is constructed as follows:

1. Construct the unlimited gradient operator

$$\vec{L} = \vec{\nabla}(\Delta ABC)$$  \hspace{1cm} (3.104)

2. For each edge $k$ calculate the scalar

$$\alpha^k = \begin{cases} 
\frac{\max(u_k,u_0) - u_0}{\vec{r}_{0k}.\vec{L}} & \text{if } (u_0 + \vec{r}_{0k}.\vec{L}) > \max(u_k,u_0) \\
\frac{\max(u_k,u_0) - u_0}{\vec{r}_{0k}.\vec{L}} & \text{if } (u_0 + \vec{r}_{0k}.\vec{L}) < \min(u_k,u_0) \\
1 & \text{otherwise}
\end{cases}$$  \hspace{1cm} (3.105)

3. Set

$$\Phi_{LCD} = \vec{L}_{LCD} = \left( \min_{all \ k} \alpha^k \right) \vec{L}$$  \hspace{1cm} (3.106)

3.4.7.4 Extended vanLeer (EV) Limiter

The Extended vanLeer is a rank 3 limiter because it considers three limiting triangles (Figure 3.8) i.e., $\Delta AOC$, $\Delta BOC$ and $\Delta COB$. The EV limiter is given by:

$$\Phi_{EV} = S_0 = 0.8 \frac{S_1||S_2|||S_3|| + ||S_1||.S_2||S_3|| + ||S_1||.S_2||.S_3||}{||S_2||.||S_3|| + ||S_1||.||S_3|| + ||S_1||.||S_2||}$$  \hspace{1cm} (3.107)

where $S_1$, $S_2$ and $S_3$ are the gradient vectors for $\Delta AOC$, $\Delta BOC$ and $\Delta COB$. The factor 0.8 to average value is assigned to increase the stability of the limiter, otherwise the limiter gives oscillation.

3.4.7.5 Maximum Limited Gradient (MLG) Limiter

MLG introduced by Batten et al. (1996), is a rank 4 limiter as it considers four limiting triangles (Figure 3.8) i.e., $\Delta ABC$, $\Delta AOC$, $\Delta BOC$ and $\Delta COB$. This reduces to Roe’s Superbee limiter in one dimension. The MLG limiter is based on computing various gradient operators in an LCD fashion and then retaining the steepest one of them as follows:
1. Compute
\[ \vec{L}_0 = \vec{L}_{LCD}(\Delta ABC), \quad \vec{L}_1 = \vec{L}_{LCD}(\Delta AB0), \quad \vec{L}_2 = \vec{L}_{LCD}(\Delta A0C) \quad \text{and} \quad \vec{L}_3 = \vec{L}_{LCD}(\Delta 0BC). \] (3.108)

2. Set
\[ \Phi_{MLG} = \vec{L}_{MLG} = \vec{L}_i \quad \text{such that} \quad |\vec{L}_i| = \max_{0<k<3} |\vec{L}_k| \] (3.109)

3.4.7.6 The MUSCL-Hancock Scheme

The MUSCL-Hancock method is a high-order method both in space and time. This method is the second order extension of Godunov upwind method and provide second-order accuracy in space and time. It updates the element states in the following three steps:

**Step I: Data Reconstruction**- In the presence of discontinuities, some numerical methods generate oscillation in the numerical results. Therefore, to tackle this problem, MUSCLE-Hancock reconstructs the data using boundary interpolated values within each control volume. For this interpolation, any of the limiters discussed in section 3.4.7 can be used.

**Step II: Evolution of State variables at half time**- The reconstructed data is advanced by half a time step to find an intermediate solution. The reconstructed data is used to calculate the flux for each control volume.

\[ U_{k}^{n+\frac{1}{2}} = U_{k}^{n} - \frac{\Delta t}{2A_k} \left[ \sum_{m=1}^{M} F(U_m^n) L_m - S_k^n \right] \] (3.110)

**Step III: Solution of the piecewise constant data Riemann Problem**- The intermediate solution gives the state variables in each control volume. Limiters are used to calculate the corresponding states at the boundaries of each control volume. These values are the left and right states at the boundaries of each control volume defining the Riemann problem. The Riemann problem is solved using any of the solvers discussed in the section 3.4, that will provide a set of upwind interface fluxes which are used to calculate the state variables forward by one full time step.

\[ U_{k}^{n+1} = U_{k}^{n} - \frac{\Delta t}{A_k} \left[ \sum_{m=1}^{M} F(U_m^L, U_m^R)^{n+\frac{1}{2}} L_m - S_k^{n+\frac{1}{2}} \right] \] (3.111)
Chapter 4

Shallow Water Flow

4.1 Introduction

This chapter is intended to provide a brief background on the theory of the two-dimensional depth-averaged shallow water equations. As stated in the previous chapters, the shallow water equations are hyperbolic in nature and are extended in the present study using the generic Riemann2D. The objective of this chapter is to present:

- the derivation of two dimensional shallow water equations from Saint-Venant equations
- the hyperbolic characteristics of shallow water equations;
- the extension of the generic solver presented in the previous chapter to shallow water equations;
- the source terms involved in the shallow water flow.

The practical value of this chapter is that it helps explain the significance of the input parameters and also highlights the reliability of the Riemann2D results as shown in chapter 6.
4.2 Free Surface Flows

The shallow water problem is also referred as free surface flows. The reason for the free designation arises from the large difference in the densities of the gas and liquid (e.g., the ratio of water to air is 1000). A low gas density means that its inertia can generally be ignored compared to that of the liquid. In this sense the liquid moves independently, or freely, with respect to the gas. The only influence of the gas is the pressure it exerts on the liquid surface. In other words, the gas-liquid surface is not constrained, but free. The mathematical models of the so-called free surface type governs a wide variety of physical phenomena for scientific and practical problems of scientific interest, ranging from conventional water wave problems to sloshing in fuel tanks in rocket technology.

An important class of problems of practical interest involve water flows with a free surface under the influence of gravity. This class includes tides in the ocean, tsunami waves, breaking of waves in rivers, surges, and dam-break wave modelling. A key assumption made in the derivation of the approximate shallow water theory concerns the pressure distribution; which is hydrostatics. This results from the assumption that the vertical acceleration of the water particles negligible compared to velocity of the water particles in horizontal plane.

Depth-averaged modelling is based on the basic physical principles of conservation of mass and momentum and on a set of constitutive laws which relate the driving and resisting forces to fluid properties and motions. The differential equations of flow are derived by considering a differential volume element of fluid and describing mathematically:

1. The conservation of mass of fluid entering and leaving the control volume; the resulting mass balance is called the equation of continuity.

2. The conservation of momentum entering and leaving the control volume; called the equation of motion.

Applications of shallow water models can be found in many day-to-day events. For instance, modelling tidal fluctuations for those interested in capturing tidal energy for commercial purposes; predicting tidal ranges and surges which can then be used in the development planning of coastal areas; and, upon coupling to a transport model, considering flow and transport phenomena. The latter application makes it possible
Chapter 4: Shallow Water Flow

4.3 Shallow Flow Equations

Consider the flow of water with a free surface under gravity in a three-dimensional domain. Figure 4.2 depicts the convention for spatial coordinates; $x - y$ determining a horizontal plane whilst $z$ defines the vertical direction, which is associated with the free-surface elevation.

The bottom boundary which can be called just as bottom or bed, is defined by a
Figure 4.2: River flow is one of the examples of free surface flows

Figure 4.3: River flow is one of the examples of free surface flows

function

\[ z = d(x, y) \]  \hspace{1cm} (4.1)
and the free surface is defined by

\[ z = s(x, y, z) \equiv d(x, y) + h(x, y, t) \]  \hspace{1cm} (4.2)

where \( h(x, y, t) \) is the depth of water, the vertical distance between the bottom and the free surface position. Figure 4.3 depicts the geometry for a simplified situation for a chosen value of \( y \).

Before deriving the shallow water theory the boundary conditions for the full problem are discussed. Assuming that a boundary is given by the surface

\[ \Omega(x, y, z) = 0 \]  \hspace{1cm} (4.3)

then for the free surface we have

\[ \Omega(x, y, z) \equiv z - s(x, y, t) = 0 \]  \hspace{1cm} (4.4)

and for the bottom boundary we have

\[ \Omega(x, y, z) \equiv z - d(x, y) = 0 \]  \hspace{1cm} (4.5)

Two boundary conditions are imposed on the free surface \( s(x, y, t) \), given by equation 4.4, namely the kinematic condition

\[ \frac{d}{dt} \Omega(x, y, z) = \Omega_t + u\Omega_x + v\Omega_y + w\Omega_z = 0 \]  \hspace{1cm} (4.6)

and the dynamic condition

\[ p(x, y, z) \bigg|_{z=s(x,y)} = p_{atm} = 0 \]  \hspace{1cm} (4.7)

where \( p_{atm} \) is the atmospheric pressure, which for convenience is taken as zero. For the bottom boundary \( (x, y) \), the condition in equation 4.6 also applies, with \( \Omega \) given by 4.5.

### 4.4 Derivation of the SWE

Assuming the density of the fluid is constant, the governing equations for the shallow water are given as:

\[ u_x + v_y + w_z = 0 \]  \hspace{1cm} (4.8)
Chapter 4: Shallow Water Flow

\[ u_t + uu_x + uw_y + uw_z = -\frac{1}{\rho}p_x \]  

(4.9)

\[ v_t + vv_x + vw_y + vw_z = -\frac{1}{\rho}p_y \]  

(4.10)

\[ w_t + wu_x + wv_y + ww_z = -\frac{1}{\rho}p_w \]  

(4.11)

Here we have assumed that the body force vector is \( g = (0, 0, g) \), where \( g \) is the acceleration due to gravity, taken as 9.8\( m/s^2 \), a constant. In principle, given initial conditions at time \( t = 0 \) and boundary conditions on the bottom and the free surface, the solution of the four equations 4.8 - 4.11 can be found for the four unknowns \( p, u, v, w \).

The main difficulty in solving the full problem is associated with the free surface, but the position of this boundary itself is unknown and therefore the domain on which the equations are to solved is not known. Approximate theories leading to simpler problem exist:

1. One such approximate theory assumes that the amplitude of the free-surface disturbance from the rest position is small with respect to a characteristic length, such as wave length. This assumption leads to linear boundary value problems and thus to a linear theory.

2. Another approximation, which is used for the shallow water extension of Riemann2D development, results from the assumption that the depth of water is small with respect to wavelength or free-surface curvature.

The first assumption in the derivation of the shallow water equations is that the vertical component of acceleration, given by

\[ \frac{dw}{dt} = w + uw_x + vw_y + ww_z \]  

(4.12)

is negligible. Insertion of this condition \( dw/dt = dt = 0 \) into equation 4.11 gives

\[ p_z = -\rho g \]  

(4.13)

Given the dynamic condition 4.7 that the atmospheric pressure is zero on the free surface, we obtain

\[ p = \rho g(s - z) \]  

(4.14)
Differentiation of equation 4.14 with respect to \( x \) and \( y \) gives

\[
\begin{align*}
 p_x &= \rho g s_x \quad \text{and} \quad p_y = \rho g s_y \\
\end{align*}
\] (4.15)

Here, \( p_x \) and \( p_y \) are both independent of \( z \) and thus the \( x \) and \( y \) components of the acceleration of water particles \( du/dt \) and \( dv/dt \) are independent of \( z \). Hence the \( x \) and \( y \) velocity components \( u \) and \( v \) are also independent of \( z \), that is \( u_z = v_z = 0 \). Therefore by the virtue of the above conditions and by making use of equation 4.15 in equations 4.9 and 4.10, we have

\[
\begin{align*}
 u_t + uu_x + vv_y &= -gs_x \\
 v_t + uv_x + vv_y &= -gs_y
\end{align*}
\] (4.16, 4.17)

An important step in deriving the shallow water equations now follows. We integrate the continuity equation 4.8 with respect to \( z \), the vertical coordinate, between \( z = d(x,y) \) (bottom) and \( z = s(s,y,t) \) (free surface). That is

\[
\int_b^s (ux + vy + wz)\,dz = 0
\] (4.18)

which leads to

\[
\begin{align*}
 w|_{z=s} - w|_{z=b} + \int_b^s u_x\,dz + \int_b^s v_y\,dz = 0
\end{align*}
\] (4.19)

We now apply the boundary condition in order to determine the first two terms in the above equation. Expanding equation 4.6 as applied to the free surface in equation 4.4 gives

\[
(s_t + us_x + vs_y + ws_z - w)|_{z=s} = 0
\] (4.20)

Expanding equation 4.6 as applied to the bottom boundary in equation 4.5 gives

\[
(ub - vb - w)|_{z=d} = 0
\] (4.21)

From equation 4.20 we obtain

\[
w|_{z=s} = (s_t + us_x + vs_y + ws_z)|_{z=s}
\] (4.22)
Chapter 4: Shallow Water Flow

and from equation 4.21 we obtain

\[ w|_{z=b} = (ub_x + vb_y)|_{z=d} \] (4.23)

Substituting 4.22 and 4.23 into 4.19 gives

\[ (s_t + us_x + vs_y + ws_z)|_{z=s} - (ub_x + vb_y)|_{z=d} + \int_b^s u_x dz + \int_b^s v_y dz = 0 \] (4.24)

To further simplify the above equation, we use Leibniz’s formula:

\[ \frac{d}{d\alpha} \int_{\epsilon_1(\alpha)}^{\epsilon_2(\alpha)} f(\xi, \alpha) d\alpha = \int_{\epsilon_1(\alpha)}^{\epsilon_2(\alpha)} \frac{df}{d\alpha} d\xi + f(\xi_2, \alpha) \frac{d\xi_2}{d\alpha} - f(\xi_1, \alpha) \frac{d\xi_1}{d\alpha} \] (4.25)

Applied to the last two integral terms in equation 4.24, we obtain

\[ \int_b^s u_x dz = \frac{\partial}{\partial x} \int_b^s u dz - u|_{z=s} . s_x + u|_{z=d} . d_x \] (4.26)

and

\[ \int_b^s v_y dz = \frac{\partial}{\partial y} \int_b^s v dz - v|_{z=s} . s_y + v|_{z=d} . d_y \] (4.27)

Substitution of equation 4.26 and 4.27 into 4.24 gives

\[ s_t + \frac{\partial}{\partial x} \int_b^s u dz + \frac{\partial}{\partial y} \int_b^s v dz = 0 \] (4.28)

Recall that both \( u \) and \( v \) are independent of \( z \); also \( s = d + h \) and \( d_t = 0 \). Equation 4.28 then simplifies to

\[ h_t + (hu)_x + (hv)_y = 0 \] (4.29)

This is the law of conservation of mass and is written in differential conservation law form.

The momentum equations 4.9 and 4.10 can also be expressed in differential conservation law form. To this end we add equation 4.29, pre-multiplied by \( u \), to equation 4.16, pre-multiplied by \( h \); we also make use of a relation that does assume differentiability of the water depth, namely
\[ h \frac{\partial h}{\partial x} = \frac{\partial}{\partial x} \left( \frac{1}{2} h^2 \right) \]  
\hspace{2cm} (4.30)

to obtain
\[ (hu)_t + (hu^2 + \frac{1}{2} gh^2)_x + (huv)_y = -ghd_x \]  
\hspace{2cm} (4.31)
similarly, for the \( y \) momentum equation we obtain
\[ (hv)_t + (hv^2)_x + (huv + \frac{1}{2} gh^2)_y = -ghd_y \]  
\hspace{2cm} (4.32)
All three partial differential equations 4.29, 4.31 and 4.32 can be written in differential conservation law form as the single vector equation
\[ U_t + F(U)_x + G(U)_y = S(U) \]  
\hspace{2cm} (4.33)

\[ U = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \quad F(U) = \begin{bmatrix} hu \\ hu^2 \\ huv \end{bmatrix}, \quad G(U) = \begin{bmatrix} hu \\ hv \\ hv^2 \end{bmatrix} \] and

\[ S(U) = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} \]  
\hspace{2cm} (4.34)

Equation 4.33 may be written in the quasi-linear form as
\[ U_t + A(U)U_x + B(U)U_y = 0 \]  
\hspace{2cm} (4.35)
where \( A(U) \) and \( B(U) \) are the Jacobian matrices, as defined in equation 4.13, of the fluxes \( F(U) \) and \( G(U) \) respectively.

It should be noted that equation 4.33 has the same form as the generic equation for the hyperbolic equation, as shown in equation 3.14. In the system of equations 4.34, \( U \) is the vector of conserved variables, \( F(U), G(U) \) are flux vectors in the \( x \) and \( y \) directions respectively, and \( S(U) \) is the source term vector. These equations can be compared to the generic equations for hyperbolic problems in equation 3.12.
This comparison shows that hyperbolic equations follow the same behavior and can be modeled using the generic Riemann2D model.

Due to the generic nature of the governing equations, the overall method of solving any hyperbolic equation would be the same. For the extension to superclass, it need to provide the conserved variable and thus the corresponding fluxes, such as equation 4.34 for shallow water equations.

### 4.5 Hyperbolic Character of the Shallow Water Equations

It is important to note that the arguments of the flux functions, actually their components, are the components of the vector of the conserved variables. To make this clear, we express the flux functions in terms of the components \( u_1, u_2, u_3 \) of \( U \),

\[
q = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix}
\]

\[
F(u) = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ hv \end{bmatrix} = \begin{bmatrix} u_2 \\ u_2^2/u_1 + \frac{1}{2}gu_1^2 \\ u_2u_3/u_1 \end{bmatrix}
\]

\[
G(u) = \begin{bmatrix} hv \\ hvu \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} u_3 \\ u_2u_3/u_1 \\ u_3^2/u_1 + \frac{1}{2}gu_1^2 \end{bmatrix}
\]

The variables in the quasi-linear form are conserved variables, the formulation of the equation is non conservative. Next we calculate the Jacobian matrices and express them in terms of the non-conservative variables \( u, v, c \), where \( c \) is the speed of the wave defined as \( c = \sqrt{gh} \).

\[
A = \begin{bmatrix} 0 & 0 & 0 \\ -(u_2/u_1)^2 + gu_1 & 2u_2/u_1 & 0 \\ -u_2u_3/u_1^2 & u_3/u_1 & u_2/u_1 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -u^2 + c^2 & 2u & 0 \\ -uv & v & u \end{bmatrix}
\]

Similarly
\[
B = \begin{bmatrix}
0 & 0 & 1 \\
-uv & v & u \\
-v^2 + c^2 & 0 & 2u \\
\end{bmatrix}
\quad (4.40)
\]

Now consider a matrix \( C \) that is a linear combination of the two Jacobian matrices \( A \) and \( B \), namely
\[
c = \omega_1 A + \omega_2 B,
\quad (4.41)
\]
where the coefficients \( \omega_1 \) and \( \omega_2 \) are two real parameters that define a non-zero vector \( \omega = [\omega_1 + \omega_2] \)
\[
\omega = \sqrt{\omega_1^2 + \omega_2^2} > 0
\quad (4.42)
\]
The matrix \( C \) is given by
\[
C = \begin{bmatrix}
0 & \omega_1 & \omega_2 \\
(-u^2 + c^2)\omega_1 - uv\omega_2 & 2u\omega_1 + v\omega_2 & u\omega_2 \\
-uv\omega_1 + (-v^2 + c^2)\omega_2 & u\omega_1 + 2v\omega_2 & u\omega_1 + v\omega_2 \\
\end{bmatrix}
\quad (4.43)
\]
The eigenvalues of \( C \) can be given as
\[
\lambda_1 = u\omega_1 + v\omega_2 - c|\omega|
\]
\[
\lambda_2 = u\omega_1 + v\omega_2
\]
\[
\lambda_3 = u\omega_1 + v\omega_2 + c|\omega|
\quad (4.44)
\]
The time-dependent two-dimensional shallow water equations 4.33- 4.34 are hyperbolic (Toro, 2001). For a wet bed they are strictly hyperbolic. This is based on the fact that the above eigenvalues in equation 4.44 are all real and, for \( h > 0 \), distinct.

4.6 Approximate Riemann Solvers

4.6.1 The Roe Solver

Roe’s approximate Riemann solver is discussed in chapter 3 for the generic approach. Due to dependence of the conserved variables on the type of the problem, the Riemann problem implementation is done here. The first approach of Roe’s approximate Riemann solver to the shallow water equations can be found in the literature by Glaister
Chapter 4: Shallow Water Flow

(1988). Glaister followed the Roe-Pike approach (Roe & Pike, 1984) to derive the averages at the cell boundaries. Therefore, following Roe’s approach from chapter 3, we can write the Roe averages for shallow water equations using the parameter vectors $z = h^{1/2} q$ i.e.,

$$
\begin{bmatrix}
z^1 \\
z^2 \\
z^3
\end{bmatrix} =
\begin{bmatrix}
\sqrt{h} \\
\sqrt{hu} \\
\sqrt{hv}
\end{bmatrix}
$$

(4.45)

The Roe averages can be found as:

$$
\tilde{u} = \frac{z^2}{z^1} = \frac{\sqrt{h_L} u_R + \sqrt{h_R} u_L}{\sqrt{h_L} + \sqrt{h_R}}
$$

(4.46)

$$
\tilde{v} = \frac{z^3}{z^1} = \frac{\sqrt{h_L} v_R + \sqrt{h_R} v_L}{\sqrt{h_L} + \sqrt{h_R}}
$$

(4.47)

$$
\tilde{h} = \frac{1}{2} (h_L + h_R)
$$

(4.48)

$$
\tilde{c} = \sqrt{gh}
$$

(4.49)

The average eigenvalues are:

$$
\tilde{\lambda}_1 = \tilde{u} - \tilde{c}
\tilde{\lambda}_2 = \tilde{u}
\tilde{\lambda}_3 = \tilde{u} + \tilde{c}
$$

(4.50)

and the corresponding eigenvectors are:

$$
\tilde{K}^1 = 
\begin{bmatrix}
1 \\
\tilde{u} - \tilde{c} \\
\tilde{v}
\end{bmatrix}
\tilde{K}^2 = 
\begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix}
\tilde{K}^3 = 
\begin{bmatrix}
1 \\
\tilde{u} + \tilde{c} \\
\tilde{v}
\end{bmatrix}
$$

(4.51)

The wave strength $\tilde{\alpha}_j$, in terms of the averages, are

$$
\tilde{\alpha}_1 = \frac{\Delta h (\tilde{u} + c) - \Delta (hu)}{2c}
\tilde{\alpha}_2 = -v \Delta h + \Delta (hv)
\tilde{\alpha}_3 = \frac{-\Delta h (\tilde{u} - c) + \Delta (hu)}{2c}
$$

(4.52)
Figure 4.4: Failure of the Roe solver when the left and right states, $u_l$ and $u_r$, lie in particular regions of phase space. In this example the left and right states are shallow, with opposite velocities. The arrows are in the direction of the Roe eigenvectors, the intersection of which is the middle state $u^*$, corresponding to a negative depth $h$ George (2004)

Now the values from equations 4.50, 4.51 and 4.52 can be used to find the $F_{Roe}$ in equation 3.49.

The Roe solver works well for shallow water flow problems but in certain situations it can produce negative depths. In addition to being physically incorrect, these negative depths usually pose computational difficulties introducing errors. Consider the solution to a linear Riemann problem with a Roe matrix $\tilde{A}$.

$$\begin{align*}
PDE \ u_t + Au_x &= 0 \\
IC \ u(x,0) &= u_0(x) = \begin{cases} 
    u_L & \text{if } x < 0 \\
    u_R & \text{if } x > 0 
\end{cases} 
\end{align*} \quad (4.53)$$

The solution’s middle state, $u^*$, is connected to the state on the left, $u_l$, by a jump in the first eigenvector of $\tilde{A}$ and is connected to the state on the right, $u_r$, by a jump in the second eigenvector. The left and right states are therefore joined in phase space by the two eigenvectors, the middle state, $u^*$, being the intersection of the two vectors.

This intersection corresponds to a negative depth when the left and right states lie in particular regions of phase space; see Figure 4.4. For instance, if the velocity on the
right is much greater than the velocity on the left and the flow is already very shallow, the intersection often corresponds to a negative depth. For applications where the flow is very shallow in parts of the domain, or in fact becomes dry in regions, the Roe solver is almost certain to produce negative depths.

It is generally accepted that the numerical computation of dry/wet bed fronts is very difficult (Toro, 2001). One way to resolve this difficulty is to compute the exact Riemann solution, which as mentioned is possible for the homogeneous shallow water equations. However, this is computationally expensive, and it is not clear how to extend this satisfactorily to a Riemann problem in a triangular element with a source term. Another approach often adopted is to assign the depth to a minimum level, which makes the water non-zero for all cases. The problem with this approach is that it has shock front which is generally missing in a dry bed situation.

4.6.2 The HLL Solver

Following the description and derivation of the HLL (Harten-Lax-van Leer) approximate Riemann solvers in section 3.3.2, these equations are extended for the homogeneous shallow water equations. These solvers are based on estimating the speeds that information or waves propagate away from a Riemann problem. A linear solution with two discontinuities is then constructed, using estimates for the speeds of the propagating discontinuities. Two speeds are used in the original HLL method even for equations with more than two characteristic families. This is in contrast to a method such as the Roe solver, where a constant estimate to the Jacobian matrix is constructed first, and the eigenvalues of this estimate subsequently affect the approximate Riemann solution. The estimates for the speeds are based on the initial data, and on general properties of exact Riemann solutions. A number of different estimates for the speeds have been used, and the particular choice of estimates gives the HLL method its particular properties; see Toro (1999).

Only working on dry bed problems, the HLL approach highlights better behavior, avoiding uni-dimensionalisation effects on the flow field [Caleffi et al. (2003)]. Such a reason leads to the choice of the HLL and its derived approximate Riemann solver (see section 3.3.2) for the development of the model where the dry bed problem occurs.
Rewriting equation 3.58 in terms of the normal components, we have

\[ F_{HLL} = \frac{S_R F_L.n - S_R F_L.n + S_L S_R (U_R - U_L)}{S_R - S_L} \] (4.54)

where \( n \) is the outward normal unit vector; \( F_R = F(U_R) \) and \( F_L = F(U_L) \); subscripts \( R \) and \( L \) refer to the right and to the left side of the cell interface respectively. The \( S_L \) and \( S_R \) symbols represent the wave speeds’ propagation and they can be estimated through the two expansion approach [Caleffi et al. (2003)]:

\[ S_L = \min(q_L.n - \sqrt{gh_L}, u^* - \sqrt{gh^*}) \] (4.55)
\[ S_L = \min(q_R.n - \sqrt{gh_R}, u^* - \sqrt{gh^*}) \] (4.56)

where \( q = (u, v) \) and:

\[ u^* = \frac{1}{2}(q_L + q_R) + \sqrt{gh_L} - \sqrt{gh_R} \] (4.57)
\[ \sqrt{gh^*} = \frac{1}{2}(\sqrt{gh_L} - \sqrt{gh_R}) + \frac{1}{4}(q_L + q_R).n \] (4.58)

Substituting the values the above four equations into equation 4.54 (or equation 3.58) we obtain the value of \( F_{HLL} \).

\[ S_L = \left\{ \begin{array}{ll}
    u_R - 2\sqrt{gh_R} & \text{if } h_L = 0 \\
    \text{usual estimate} & \text{if } h_L > 0
\end{array} \right. \] (4.59)

Consider a one dimensional Riemann problem where initial data is given, for e.g., the left state, \([h_L, u_L]\) is specified, and the right state is dry so that \([h_R, u_R] = [0, 0]\). The exact solution to this problem can been seen in chapter 6 of Toro (2001). The solution consists of a single rarefaction associated with the left eigenvalue \( \lambda_1 = u - a \). The expected right shock associated with the left eigenvalue \( \lambda_2 = u + a \) is absent. The wet/dry front corresponds to the tail of the left rarefaction and has exact propagation speed \( S_L = u_L + 2\sqrt{gh_L} \). It is interesting to note that the speed of this front is faster than one obtained from the usual evaluation of eigenvalues of the system. A popular way of dealing with these kinds of problems is by artificially wetting the dry bed, that is by setting the water depth on the right-hand side in our problem to some small positive tolerance, namely \( h_R = \epsilon > 0 \) in the below illustration. Having done this the solution to the Riemann problem has a different structure to that of the exact problem for the dry bed conditions. The solution contains a relatively weak, right-propagation shock of speed \( S_R \), which is meant to represent the wet/dry front speed. The speed \( S_R \) of
Chapter 4: Shallow Water Flow

Figure 4.5: Constructing a solution using conservation and estimated speeds. If the speeds \( s_1 \) and \( s_2 \) are predetermined estimates, an approximate solution can be uniquely determined by applying the conservation law to the region \( \omega = [x_l, x_r] \) surrounding the interface George (2004)

this shock is considerably slower than that of the wet/dry front, \( S_{*L} \) (see Figure 4.5, \( S_{*L} \) is the speed of the wave in the star region). In the limit as \( \epsilon \) tends to zero, the two speeds coincide. However, for practical values of the artificial bed-wetting parameter \( \epsilon \), the errors can be significant.

The above solution assumes that there exists a finite water depth everywhere. In the above example where the dry bed exists downstream \( h_R = 0 \), the two eigenvalues collapse into one and the system of equations is not strictly hyperbolic. Under these circumstances no shock exists and \( S_R \) represents the speed of the head of the rarefaction wave and \( S_L \) represents the speed of the toe of the rarefaction wave. Therefore, a general expression can be written as

\[
S_R = \begin{cases} 
  u_L + 2\sqrt{gh_L} & \text{if } h_R = 0 \\
  \text{usual estimate} & \text{if } h_R > 0
\end{cases}
\]  

(4.60)

4.7 Source Terms

In recent years, the study of a hyperbolic system with source terms has attracted much attention in the CFD community (Xu, 2002). One of the main reasons is that there exist wide engineering applications. For example, the Saint-Venant equations are widely used in ocean and hydraulic engineering to describe bore wave propagation, hydraulic jumps, and open-channel flow, among others. Many numerical schemes have

### 4.7.1 Coriolis Factor

Because the Earth rotates, a fluid that flows along the Earth’s surface experiences a coriolis acceleration perpendicular to its velocity. In the northern hemisphere, coriolis acceleration makes low pressure storm systems spin counterclockwise; however, in the southern hemisphere, they spin clockwise because the direction of the coriolis acceleration is reversed. The order of magnitude of the coriolis acceleration can be estimated from size of the Rossby number. The dimensionless ratio of inertia force to Coriolis force which gives an indication of the importance of rotation on ow in pipes. It is given by

\[
R_0 = \frac{v}{2\omega L \sin \theta}
\]  

(4.61)

where \(v\) is the speed of fluid flow, \(\omega\) is the angular velocity or rotation, and \(\theta\) is the angle between the axis of rotation and the direction of fluid motion.

The coriolis parameter, \(f\) is defined as twice the vertical component of the Earth’s angular velocity \(\omega\) about the local vertical, and is given by

\[
f = 2\Omega \sin \phi
\]  

(4.62)

at latitude \(\phi\).

### 4.7.2 Bed Resistance

Resistance to flow is typically characterized by a roughness coefficient. The most commonly used equation for flow resistance is Manning’s equation. There are other resistance coefficients in use including the Darcy-Weisbach friction factor, \(f\), and the Chezy \(C\). These can all be converted easily to Mannings \(n\).

**Chezy Formula:** This expression has the form:

\[
V = C(RS)^{1/2}
\]  

(4.63)

**Manning Formula:** The following expression for the evaluation of the Chezy roughness coefficient was suggested by Flamant (1891):

\[
C = \frac{R^{1/6}}{n}
\]  

(4.64)
where $n$ is a roughness parameter which depends only upon the roughness characteristics of the boundary surface and which is known as the Manning roughness parameter.

4.8 Conclusion

In this chapter the theory of shallow water equations is presented. Initially in the chapter the derivation of two dimensional shallow water equation from Saint-Venant equation is presented, followed by the hyperbolic characteristics of shallow water equations. The generic solver presented in chapter 3 are extended for shallow water equations. At the end of the chapter, the source terms involved in the shallow water flow are presented. This chapter helped to understand the theory of shallow water flow and its link to the general hyperbolic problems.
Chapter 5

Extension of Riemann2D

5.1 Introduction

Riemann2D, proving a generic framework for hyperbolic problems, was applied to shallow water flow over fixed bed. The model was only applicable to cases with flat bed and also was unable to deal with wetting and drying problem. In this chapter we will discuss the extension of Riemann2D to deal with flow over mobile bed problems. Also some new algorithms are included to solve the wetting and drying problem and make the model capable of handling flow over varying topography in real world problem.

Each section in this chapter discusses the extension of different parts of the model. The object oriented implementation in Riemann2D of this work is highlighted in the gray shaded area in each section.

5.2 Cao Pender Erosion Model

Riemann2D provides a generic framework to solve hyperbolic problems, which can be implemented by a new piece of code (extended model) to solve different hyperbolic problems. Jha (2006) applied the model to shallow water equations by adopting the generic part of Riemann2D.

Keeping in view the object-oriented design and the capability of easy implementation of Riemann2D, the model is applied two dimensional shallow water equations
coupled with erosion model, by adopting both the generic and shallow water parts of the model.

5.2.1 Governing Equations

Cao et al. (2004) developed a theoretical model, based on the shallow water theory, for dam-break hydraulics over mobile bed. The model was restricted to one dimensional flow in a channel with rectangular cross sections of constant width.

In this section the above framework is extended to natural channels with irregular geometry and two dimensions.

The mass conservation equation for the water-sediment mixture can be written as

\[ \frac{\partial}{\partial t}(\rho h) + \frac{\partial}{\partial x}(\rho hu) + \frac{\partial}{\partial y}(\rho hv) = -\rho_0 \frac{\partial z_b}{\partial t} + \rho_w R \]  

(5.1)

The x-component of momentum conservation equation for the water-sediment mixture is given by

\[ \frac{\partial}{\partial t}(\rho hu) + \frac{\partial}{\partial x}\left(\rho hu^2 + \frac{1}{2}\rho gh^2\right) + \frac{\partial}{\partial y}(\rho huv) = -\rho gh \left(\frac{\partial z_b}{\partial x} + S_{fx}\right) \]

(5.2)

The y-component of momentum conservation equation for the water-sediment mixture is given by

\[ \frac{\partial}{\partial t}(\rho hv) + \frac{\partial}{\partial x}(\rho huv) + \frac{\partial}{\partial y}\left(\rho hv^2 + \frac{1}{2}\rho gh^2\right) = -\rho gh \left(\frac{\partial z_b}{\partial y} + S_{fy}\right) \]

(5.3)

Mass conservation equation for the sediment material is

\[ \frac{\partial}{\partial t}(hc) + \frac{\partial}{\partial x}(huc) + \frac{\partial}{\partial y}(hvc) = E - D \]  

(5.4)

Mass conservation equation for the bed material is

\[ \frac{\partial z_b}{\partial t} = \frac{D - E}{(1 - p)} \]  

(5.5)

where \( \rho = \rho_w (1 + c) + \rho_s c \)=density of water sediment mixture; \( h \)=water depth; \( t \)=time; \( x \)=distance along the channel; \( y \)=perpendicular distance to x-axis \( u \)=depth averaged...
velocity in the x-direction; v=depth averaged velocity in the y-direction; \( \rho_0 = \rho_w p + \rho_s (1 - p) \) = density of saturated bed; \( z_b \)= bed elevation; \( \rho_w \)= density of water; \( R \)= rainfall intensity; \( S_{fx} \)= friction slope in the x-direction; \( S_{fy} \)= friction slope in the y-direction; \( c \)= flux-averaged volumetric sediment concentration;

To expedite numerical solution using conservative variables, it is needed to recast equations (5.1), (5.2) and (5.3). By substituting equations (5.4) and (5.5), equation (5.1) can be written as

\[
\frac{\partial}{\partial t} h + \frac{\partial}{\partial x} (hu) + \frac{\partial}{\partial y} (hv) = -\frac{E - D}{1 - p} \tag{5.6}
\]

Substituting equations 5.4 and 5.6, equation 5.2 can be written as

\[
\frac{\partial}{\partial t} (hu) + \frac{\partial}{\partial x} \left( hu^2 + \frac{1}{2} gh^2 \right) + \frac{\partial}{\partial y} (huv) \\
= -gh \left( \frac{\partial z_b}{\partial x} + S_{fx} \right) - \frac{(\rho_s - \rho_f) gh^2}{2 \rho} \frac{\partial c}{\partial x} \\
+ \frac{(\rho_0 - \rho)(E - D)u}{\rho(1 - p)} \tag{5.7}
\]

Similarly with substitution of equations 5.4 and 5.6, equation 5.3 can be written as:

\[
\frac{\partial}{\partial t} (hv) + \frac{\partial}{\partial x} (huv) + \frac{\partial}{\partial y} \left( hv^2 + \frac{1}{2} gh^2 \right) \\
= -gh \left( \frac{\partial z_b}{\partial y} + S_{fy} \right) - \frac{(\rho_s - \rho_f) gh^2}{2 \rho} \frac{\partial c}{\partial y} \\
+ \frac{(\rho_0 - \rho)(E - D)v}{\rho(1 - p)} \tag{5.8}
\]

Equation (5.6), mass conservation equation for the water-sediment mixture, differs from the traditional conservation equation for the clear-water. The term on the right hand side of the equation is significant for the process of sediment transport and morphological evolution (Cao et al., 2004).

Equations (5.7) and (5.8), the momentum conservation equations, have got two additional terms on the right hand sides as compared to those for clear-water. The second terms in both the equations represent the variable sediment concentrations along their particular axis. The third terms indicates the momentum transfer due to sediment exchange between the water and erodible bed. The deposition of sediment in the water will tend to increase the momentum, producing a positive source term. On
the other hand, the sediment entrainment will tend to reduce the momentum, producing a negative source term (Cao et al., 2004).

To make the above set of equations compatible with the one used for the shallow water flow, the body force terms on the left sides in equations (5.7) and (5.8) are taken to the right hand side. $h$ in equation (5.6) is replaced by $\eta - Z_b$, where $\eta$ is the free surface elevation. Also, to include the source terms of rainfall, coriolis force, bed stress, wind force and bottom friction equations (5.6), (5.7) and (5.8) are written as

\[
\frac{\partial}{\partial t} \eta + \frac{\partial}{\partial x} (hu) + \frac{\partial}{\partial y} (hv) = R \tag{5.9}
\]

\[
\begin{align*}
\frac{\partial}{\partial t} (hu) &+ \frac{\partial}{\partial x} (hu^2) + \frac{\partial}{\partial y} (huv) = -ghS_{fx} - gh \frac{\partial \eta}{\partial x} \frac{(\rho_s - \rho_f)gh^2\partial c}{2\rho} - \frac{(\rho_0 - \rho)(E - D)}{\rho(1 - p)}u + f hv + c'_f f\omega^2 \cos \alpha \\
&+ \frac{\partial}{\partial x} (huv) + \frac{\partial}{\partial y} (hv^2) = f hu + c'_f f\omega^2 \sin \alpha \tag{5.10}
\end{align*}
\]

Equations (5.9), (5.10), (5.11) and (5.12) can be written in differential conservation law form as the single vector equation

\[
U_t + F(U)_x + G(U)_y = S(U) \tag{5.13}
\]

with
\[
\begin{bmatrix}
  h \\
  hu \\
  hv \\
  hc
\end{bmatrix},
F(U) = \begin{bmatrix}
  hu \\
  hu^2 \\
  huv \\
  hc
\end{bmatrix},
G(U) = \begin{bmatrix}
  hu \\
  huv \\
  hv^2 \\
  hc
\end{bmatrix}
\]
and
\[
S(U) = \begin{bmatrix}
  R - ghS_t - gh \frac{\partial h}{\partial x} \frac{\partial c}{\partial x} + \frac{(\rho_s - \rho)(E - D)u}{\rho(1 - p)} + fhv + c'_f \omega^2 \cos \alpha \\
  -ghS_t - gh \frac{\partial h}{\partial y} \frac{\partial c}{\partial y} + \frac{(\rho_s - \rho)(E - D)v}{\rho(1 - p)} - fhv + c'_f \omega^2 \sin \alpha
\end{bmatrix}
\tag{5.14}
\]

5.2.2 Model Closure

Empirical functions are needed to close the equations discussed in the above section. For the friction slope the conventional empirical relations are used, where \( n \) is the Manning’s coefficient of roughness

\[
S_{fx} = \frac{n^2 u^2}{h^{4/3}} \tag{5.15}
\]
\[
S_{fy} = \frac{n^2 v^2}{h^{4/3}} \tag{5.16}
\]

There are two distinct mechanisms for sediment exchange between the water and the sediment material, i.e., the sediment entrainment due to the turbulence of the flow and sediment deposition due to the gravity. The following empirical relations introduced by Cao (1999) are used for sediment deposition and bed erosion are used.

\[
D = \omega_0 (1 - C_a)^m C_a \tag{5.17}
\]
\[
E = \frac{160 (1 - p) (\theta - \theta_c) dU_\infty}{R^{0.8} \theta_c} \frac{\theta - \theta_c}{h} \tag{5.18}
\]

where \( \omega_0 \) = settling velocity of the sediment particle in tranquil water; \( C_a \) = local near-bed sediment concentration in volume; \( m \) = exponent; \( R \equiv \sqrt{sgd.d/\nu} = \rho_s/\rho_w - 1 \); \( \nu \) = kinematic viscosity of water; \( U_\infty \) = free surface velocity; \( \theta \equiv u^2/sgd \) = Shields parameter; \( u_* \) = friction velocity; and \( \theta_c \) = critical Shields parameter for the initiation of sediment movement.

Cao et al. (2004) recast equations (5.17) and (5.18) by approximating \( C_a = \alpha c \), where \( \alpha = \min[2, (1 - p)/c] \) and \( U_\infty = 7u/6 \). Also the following values were considered
Chapter 5: Extension of Riemann2D

\[ g = 9.8 \text{m/s}^2, \quad m = 2.0, \quad n = 0.03, \quad p = 0.4, \quad s = 1.65 \]
\[ v = 1.2E - 6m2/s, \quad \theta_c = 0.045 \]  \hspace{1cm} (5.19)

\[ D = \alpha \omega \eta (1 - \alpha C_a)^m \]  \hspace{1cm} (5.20)

\[ E = \begin{cases} 
(\theta - \theta_c)u \eta^{-1}d^{-0.2} & \text{if } \theta \geq \theta_c \\
0 & \text{else} 
\end{cases} \]  \hspace{1cm} (5.21)

Object Oriented Design CaoPender Erosion Model is an extension of the Riemann2D, and thus all the components related to this model are included in a newly added package called CaoPenderErosion Pakage.

5.2.3 CaoPenderErosion Package

Riemann2D is developed in two levels. Level I being the highest level, contains all the common components required for the solution of hyperbolic problems. This level is not a stand-alone model, which does not solve any problem. Level II, contains the ShallowWater Package, inherits the components of Level I, solves the shallow water equations. Further packages like, Euler Equations of Gas Dynamics and Kortweg-de-Vries, can be included in this level.

Equation 5.14 shows the components of CaoPenderErosion model. It can be seen that the first three components of the vector are common to Shallow Water Equations and CaoPenderErosion Model. There is only need of the additional fourth component
and the extra source terms for the Erosion model in Riemann2D. The additional component is included in a new package called CaoPenderErosion Package and the common components are inherited from the Shallow Water Package and Generic Package as shown in the figure 5.1.

Figure 5.2 the relationship among the objects of the three packages. The objects (classes) of CaoPenderErosion Package, i.e, CaoPenderErosionMesh and CaoPenderErosionElement are subclasses of the objects of Shallow Water Package, i.e., Mesh and Element. Similarly the objects of Shallow Water Package, i.e, Mesh, Element, Side, Node and Solver are the subclasses of the objects of Generic Package, i.e., GenericMesh, GenericElement, GenericSide, GenericNode and GenericSolver respectively.
Figure 5.2: Relationship between various classes in the Generic Package, the Shallow Water Package and the CaoPenderErosion Package.

5.3 Wetting and Drying

Natural topographies involve positive and negative bed slopes that can be steep in many places (mountainous areas) and abrupt banks. The presence of extreme slopes, high roughness and strong changes in the irregular geometry represent a great difficulty that can lead to important numerical errors presumably arising from the treatment of the wetting/drying fronts and the source terms discretization. In this section such types of problems with their solution are addressed.
5.3.1 Wet and Dry Cells

In Riemann2D, the conserved variables $\eta$ (mass), $hu$ (x-component of momentum and $hv$ (y-component of momentum) are computed. From these conserved variables the velocity components $u$ and $v$ in the cell are computed as

\begin{align*}
u &= \frac{hu}{h} \quad (5.22) \\
v &= \frac{hv}{h} \quad (5.23)
\end{align*}

At the wet/dry front, sometimes, the flow depth is very small and therefore the operation of equations 5.22 and 5.23 result in very high values of velocities which are not practical in case of shallow water flows. This error grows with time and causing numerical instability for the whole problem.

To overcome this difficulty, the concept of minimum depth is used here. If a particular cell has a depth less than minimum depth, the cell is called dry cell otherwise it is called wet cell. For dry cell, the velocity components $u$ and $v$ are restricted to be zero, in other words the cell is dealt as a dead cell, where no computation takes place.

Another problem arises at the interface with dry cells on both sides of the interface. Solving Riemann problem, on such an interface, using any of the available solvers will result in non-physical fluxes.

To solve this issue the model is restricted not to solve Riemann problem at an interface having dry cells on both its sides. Instead, all the fluxes are set equal to zero.

Algorithm 5.1

\begin{itemize}
\item After each time step the depth in each cell is calculated. Cells are flagged as WET if the depth is greater than minimum depth and DRY otherwise.
\item For dry cells, the velocity components $u$ and $v$ are set equal to zero and for wet cells they are calculated using equations 5.22 and 5.23.
\item If one or both the cells at an interface are wet, the solver is allowed to solve the Riemann problem. In other case the fluxes across the interface set equal to zero.
\end{itemize}
Figure 5.3: WET and DRY cells. Side AB and BC are allowed to calculate fluxes while the fluxes across CD are set equal to zero during computation.

Object Oriented Design

In algorithm 5.1 h is used to flag a cell as WET or DRY. The constructor of the Element class in the ShallowWater Package initializes a cell to WET or DRY depending on the value of h.

5.3.2 No flow condition

Solving the flow equations on irregular triangular mesh, the Riemann problem is solved on each side of the triangles. Dealing with wet/dry fronts on variable topography, there are certain situations which need special attention while solving Riemann problem, otherwise it could create non-physical fluxes across the sides in these situations. The cases where such situation can arise are shown in the figure 5.4.

- Figure 5.4(a) shows a case where \( \eta_L < \eta_R \) and both the cells are dry. In this case no fluxes are allowed across the cell interface.

- Figure 5.4(b) shows a case with \( \eta_L < \eta_R \) and the left cell is WET and the right
Chapter 5: Extension of Riemann2D

Figure 5.4: Various scenarios of flow over variable topography. In case (a) and (b) the fluxes are set equal to zero, (c) and (e) fluxes in the left direction are allowed, (d) is allowed to permit fluxes in the right direction and for case (f) there is no need to modify the fluxes.

one is DRY. In this situation model will allow for the right going fluxes. As physically evident, there cannot be fluxes across such an interface, the fluxes are set equal to zero.

- Figure 5.4(c) shows a situation where $\eta_L < \eta_R$, $\eta_L < Z_{bR}$ and both the cells are WET, only left going fluxes are allowed.
- Figure 5.4(d) shows a situation where $\eta_L > \eta_R$ and the left cell is WET and the right one DRY, only fluxes in the right direction are considered.
- Figure 5.4(e) presents a case with $\eta_L < \eta_R$ and having a DRY cell on left side and WET on the right. In this case only left going fluxes are allowed.
- Figure 5.4(f) shows a situation where $\eta_R > \eta_L$, both the cells are WET and $\eta_L > Z_{bR}$. In this case the model has the capability to compute the fluxes.
across the interface. In such cases the fluxes can be in the right or left direction depending on the wave speed and direction in the both the cells.

Object Oriented Design
The no flow condition uses the free surface elevation, depth and bed elevation at the sides of a cell. These components are specific to the Side class in the ShallowWater Package. This algorithm is added as a new method applyWettingAndDrying in Side class.

5.3.3 Partially Submerged Cells
Various limiter are used to compute the state variables at the sides of a cell. For these variables to be computed, the limiters use information of the cells on the adjacent sides of that cell as shown in figure 5.5, in different ways depending on the limiter type.

![Figure 5.5: Data reconstruction for triangular mesh.](image)

It has been observed that these limiters may create negative depths on one or two sides of a cell near the wet/dry front.

There are three possible conditions near the wet/dry front over variable topography as shown in figure (5.6). The first one is that the state variables at all the three sides are positive. The second one is that only one side has negative depth and the third one is that two side have negative depths. It is not possible that all the three sides have negative depths, so this case does not need to be considered.

\footnote{A Debug Graphical User Interface is developed in Riemann2D. This helps to visualize a small mesh and observe the state variables and the fluxes at each time step.}
Chapter 5: Extension of Riemann2D

(a) flow depths at all the three sides of the cell are positive and hence there is no need to adjust the state at these sides.

(b) A situation where one side has negative depth, i.e., \( h_1 > h_2 > 0 \) and \( h_3 < 0 \). Two possible cases of adjustments.

(c) A case where two sides have negative depth, i.e., \( h_1 > 0 \) and \( h_1 \) and \( h_2 \) are both negative.

Figure 5.6: Adjustment of water depths at the sides of a cell in different cases.
For the first case, there is no need to modify the state variables at the sides. For the second case, the depth at the side, the side with the negative depth, is set equal to zero and the difference is distributed with the depths at the other two sides such that the mass remains conservative.

In the third case, the depths at the two sides, having initial negative depths, are set equal to zero and difference is added to the depth at the third side, thus ensuring mass conservation.

Algorithm 5.2

1. If the depths at all the three sides are positive, there is no need to modify the depths at the sides.

2. If only one side has a negative depth, say \( h_3 \), \( h_3 \) is set equal to zero. To make the mass conservative, if both the depths \( h_1 \) and \( h_2 \) are greater than \( h_3/2 \), then both of them are reduced by \( h_3/2 \). If one of the sides with the positive depths, say \( h_2 \) is having depth less than \( h_3/2 \), then \( h_2 \) is also set equal to zero and \( h_1 \) is modified by adding \( h_2 - h_1 \).

3. If two of the sides in a cell have negative depths, say \( h_1 \) and \( h_2 \), then we set \( h_1 = h_2 = 0 \) and the depth at the third side of the cell is reduced by \( h_1 + h_2 \).

Object Oriented Design

Algorithm 5.2 adjust the water depths at the sides of a cell. This is done by a method adjustStateAtSides include in the ShallowWater Package.

5.3.4 Friction Term Limitation

The friction source term in the Saint-Venant equations is often one of the dominant terms, especially in river and overland flow or in surface irrigation applications. This relevance has consequences at the discrete level particularly as far as numerical stability is concerned and it is essential to establish a stability conditions that takes them into account. From the physical point of view the friction force has an upper bound that cannot be exceeded: the maximum value able to stop the flow. This fact, evident at the physical level, can be violated at the discrete level and this is reason why friction terms produce numerical instability in the solution. To avoid this numerical instability, a suitable limitation of the numerical friction force is needed (Burguete et al., 2008).
Chapter 5: Extension of Riemann2D

Applying equation 3.21 for an \( i \)th cell in a mesh to 5.10, it can be written as

\[
(hu)^{n+1}_i = (hu)^n_i - \Delta t(F_1 - S_1) - \frac{\Delta t}{A} \sum_{s=1}^{N=3} [n_1 F(U) + n_2 G(U)]
\] (5.24)

where \( F_1 \) is the first term on the RHS of equation 5.10, representing the friction term, while the rest of the source terms are represented by \( S_1 \).

Let say we represent the updated state in the next time step without friction by

\[
(hu)^*_i = (hu)^n_i + \Delta t S_1 - \frac{\Delta t}{A} \sum_{s=1}^{N=3} [n_1 F(U) + n_2 G(U)]
\] (5.25)

That involves all the forces except friction forces. Given that the maximum effect of the friction force is to stop the water flow, a necessary condition in the solution is that the updated value of the discharge at a point \((hu)^{n+1}_i\) after the addition of the discrete friction term retains the same sign of the value at the previous time level \((hu)^n_i\), that is:

\[
(hu)^{n+1}_i (hu)^*_i \geq 0
\] (5.26)

which can be written as:

\[
((hu)^*_i - \Delta t F_1) (hu)^*_i \geq 0
\] (5.27)

providing a numerical bounding value for the allowable friction force:

\[
F_1 \leq \frac{(hu)^*_i}{\Delta t}
\] (5.28)

When the numerical friction force exceeds this value, it must be limited to the maximum value. Similarly, for the \( y \)-component of momentum

\[
F_2 \leq \frac{(hv)^*_i}{\Delta t}
\] (5.29)
Object Oriented Design: The friction term is common term for both the ShallowWater and CaoPenderErosion Packages. Therefore, the friction term limitation condition is implemented in the updateState method of the ShallowWaterElement class, so that it can be overridden by the updateState method of the CaoPenderErosionElement class.

5.4 Conclusion

In this chapter, the extension work of Riemann2D was discussed. Initially, the two dimensional form of shallow water equations coupled with erosion equations were presented. The object oriented implementation of these equations were discussed. Later in the chapter, wetting and drying issues along with simple solution for them were presented.
6.1 Introduction

It is important to thoroughly look at the performance of the Riemann2D before we apply it to our case study. This chapter is devoted to discuss various test cases for this purpose. Tests with different scenarios, which we may come across while dealing with real problems, are selected. The test cases are divided in two broad categories:

- Tests over fixed bed
- Tests over erodible bed

Each of them are presented in detail in the following sections.

6.2 Tests over fixed bed

The following tests cases are carried out to study the performance of Riemann2D over fixed bed.

- Dam-break problem
- Dam-break on flat bed with wetting and drying;
- Dam-break on slope with wetting and drying;
• Sub-critical, Super-critical and Transitional Flow

• Flow past a hump;

• Dam-break in a converging-diverging channel;

Each of these tests are discussed in the following sections.

6.2.1 Dam break problem

In this test an idealized dam-break flow over horizontal frictionless and fixed bed is simulated by Riemann2D.

The purpose of this is to assess the accuracy of Riemann2D for an idealized dam-break over a flat, frictionless and fixed bed, for which the analytical solution is known (Stoker, 1957).

The channel is 200m long and 50m wide. The dam is located at the mid (x = 100m) of the channel. The water elevation is 10m on the upstream and 1m on the downstream of the dam. The Manning’s n is taken equal to 0, to achieve a frictionless bed condition.

Figure 6.1 shows the comparison between the analytical and the numerical solution. It can be seen that the numerical results, for both free surface elevation and velocity profile, agrees with the analytical solution rather well without any distinguishable discrepancy.
Figure 6.1: Comparison between the numerical and exact solutions for a horizontal frictionless bed: (a) free surface and (b) velocity profiles.
6.2.2 Dam-break with wetting and drying

In this test a dam-break problem over a flat bed with wetting and drying is investigated. The objectives of this test are to:

- Check the applicability of Riemann2D with wetting and drying problem.
- The performance of the algorithm discussed in chapter 5.
- Compare the numerical results against the analytical ones.

The domain for this test is 1000m long and 500m wide. The dam is located at the midpoint of the domain in the longitudinal direction \((x = 500\text{m})\). The depth of the reservoir is 4m on the upstream side while the bed is completely dry on the downstream side. Manning’s \(n = 0.03\) is used for this test.

The problem is simulated by instantaneous partial dam break (200m middle portion of the dam is removed), using HLL solver and Minmod limiter.

Figure 6.2 shows the free surface profile and the velocity vectors. From figure 6.2(a) it can be seen that the wet/dry front is propagating as it can be expected. The water from the sides of the reservoir flows into the breach section of the dam due to the transverse component of momentum. The surge wave with the wet/dry front is captured without any numerical instability. These observations shows the performance of the algorithm discussed in Chapter 5.
Figure 6.2: Dam break after 24s on flat bed with wetting and drying: (a) free surface elevation (b) velocity vectors
6.2.3 Dam-break on slope with wetting and drying

A dam break problem on fixed sloping bed with wetting and drying is studied. The purpose of this test is to see:

- The performance of Riemann2D to on a sloping bed with wetting and drying.
- The applicability of the algorithm discussed in chapter 5 for a fixed bed.

The domain for this test is 1000m long and 500m wide. The dam is located at the midpoint of the domain in the longitudinal direction (x = 500m). The elevation at x = 0m is 40m and 0m at x = 1000m, making a slope of 1 in 25 in the x-direction. The free surface on the upstream side of the dam is 44m and the bed is completely dry on the downstream side of the dam. Manning’s value of 0.03 is used for the test.

The problem is simulated by instantaneous partial dam break (200m middle portion of the dam is removed), using HLL solver and Minmod limiter.

Figure 6.3 shows the results for a dam break on a sloping fixed bed with dry bed on the downstream of the dam. It is clear from the figure 6.3(a) that the wavefront propagates in the transverse direction well ahead of the initial dam site as compared to that on the flat bed case as shown in figure 6.1(a). This is because of the sloping bed, which is an obvious physical phenomenon. Furthermore, the flow pattern is as it can be expected with no instabilities.

The velocity vectors in figure 6.3(b) show, the slow accelerating flow in the reservoir and supercritical flow on the downstream sloping bed. The pattern is in a good symmetric shape as expected.
Figure 6.3: Dam break on after 24s on sloping bed with wetting and drying: (a) free surface elevation (b) velocity vectors
6.2.4 Sub-critical, Super-critical and Transitional Flow

Dealing with real flow problems, one can come across to deal with flow which changes between subcritical and supercritical flow. In this section we will discuss the ability of Riemann2D to deal with such type of flows.

The objectives of this test are:

- To access the ability of Riemann2D to deal with sub-critical, super-critical and transitional flows;
- To check the model performance over variable topography.
- Compare the numerical results against the analytical results, as derived by MacDonald (1994)

The test is conducted in five sub parts as defined in the Table 6.1 [Crowder et al. (2004c) and Crowder et al. (2004a)]. There are 101 cross sections with 1m spacing. Boundary conditions as specified in table 6.1 are used. The flow is simulated from time \( t = 0 \) to 1 hour, using HLL solver in conjunction with Minmod Limiter. The Manning’s roughness coefficient is taken as 0.03 for all the tests.

Table 6.1: Definition and type of boundary condition used for the sub-parts N.1.1 to N.1.5 in the sub-critical, super-critical and transitional flows test case.

<table>
<thead>
<tr>
<th>Test</th>
<th>Description</th>
<th>Inflow</th>
<th>Boundary</th>
</tr>
</thead>
<tbody>
<tr>
<td>N.1.1</td>
<td>Sub-critical flow</td>
<td>20.0</td>
<td>Quasi-steady</td>
</tr>
<tr>
<td>N.1.2</td>
<td>Super-critical flow</td>
<td>20.0</td>
<td>Quasi-steady</td>
</tr>
<tr>
<td>N.1.3</td>
<td>Sub-critical to Super-critical flow</td>
<td>20.0</td>
<td>Quasi-steady</td>
</tr>
<tr>
<td>N.1.4</td>
<td>Sub-critical to Super-critical to Sub-critical flow</td>
<td>20.0</td>
<td>Quasi-steady</td>
</tr>
<tr>
<td>N.1.5</td>
<td>Super-critical to Sub-critical to Super-critical flow</td>
<td>20.0</td>
<td>Quasi-steady</td>
</tr>
</tbody>
</table>

Figure 6.6 shows the results for all the sub-parts of the N.1 test. The flow direction is from right to left. The result for each part is discussed as follow:

- The results for the N.1.1 are shown in the figure 6.4(a). Initially, there is appreciable difference between the analytical and numerical solution but as the
we move downstream of the channel the difference decreases and both the results almost coincides at the end of the channel.

- Figure 6.4(b) shows the results for the test N.1.2. The numerical and analytical solutions for this test are in close agreement with no visible difference.

- The results for the Test N.1.3 are shown in the figure 6.5(a). The results show that Riemann2D can accurately calculate the flow which changes from sub-critical to super-critical.

- Figure 6.5(b) shows the results for N.1.4 test in which the flow changes from sub-critical to super-critical and then to sub-critical in the last section of the channel. The initial sub-critical flow followed by the super-critical flow and the hydraulic jump computed by Riemann2D are in very close agreement with the analytical solution. The data from final sub-critical flow computed by Riemann2D as compared to the analytical solution is slightly scattered. Most probably, this is due to the transverse component of momentum in the Riemann2D. But still, as obvious from the figure, if we take the average values of surface elevation in the Y direction, this will give us a solution more close to the analytical one.

- Figure 6.6 shows the results for the N.1.5 test. Both the numerical and analytical results are close enough with no visible difference except the one at the hydraulic jump. The hydraulic jump captured by Riemann2D is very smooth as compared to that by the analytical solution and this is one of the main capabilities of Riemann2D to capture shocks using high resolution methods.
Chapter 6: Validation of Riemann2D

Figure 6.4: Comparison of numerical and analytical results for (a) N.1.1 and (b) N.1.2 tests. The flow enters at the left and leaves on the right of the channel in both the tests.
Figure 6.5: Comparison of numerical and analytical results for (a) N.1.3 and (b) N.1.4 tests. The flow enters at the left and leaves at the right of the channel in N.1.3 test while for the N.1.4, the flow enters at the left and is reflected from the wall at the right of the channel.
Figure 6.6: Comparison of numerical and analytical results for N.1.5 test. The flow enters at the left and leaves at the right of the channel.
After discussing all the above results, it can be concluded that the Riemann2D can deal with sub-critical, super-critical and transitional flows with a very good accuracy. In these tests the ability of Riemann2D over variable topography is also assessed.

6.2.5 Flow over a bump

This test is found in work done by many researchers. The test was first introduced by (Goutal & Maurel, 1997). The purpose of this test is to:

- Study the behavior of flow computed by Riemann2D over a bump.
- Assess the ability of Riemann2D to converge to a steady state solution.

The test domain comprises of a 25m long and 10m wide channel. The bed profile of the channel is defined by

\[
Z_b = \begin{cases} 
0.2 - 0.05(x - 10)^2 & 8 < x < 12 \\
0 & \text{otherwise}
\end{cases}
\]  

(6.1)

where \(Z_b\) is the bed elevation and \(x\) is the distance is the longitudinal direction. The initial free surface elevation is taken as 3m throughout the channel and a unit discharge of 0.18 m\(^2\)/s is imposed on the upstream of the channel. The downstream of the channel is allowed to transmit the flow. The flow is simulated for 20mins to achieve the steady state.

Figure 6.7 shows the results for the flow over a bump. It can be seen that both water elevation and velocity are constant till the start of the bump. At the start of the bump location, the water level gradually starts falling and the corresponding velocity increases. Near the end of the bump, a hydraulic jump is generated and after the hydraulic jump both the elevation and velocity of flow remains constant. The results shows that a perfect steady state flow is achieved.
Figure 6.7: Flow over a bump.
6.2.6 Dam-break in a converging-diverging channel

This test is conducted in a channel with a dam break followed by gradual contraction and expansion. The objectives of this test are to:

- Look at the behavior of Riemann2D to simulate the flow caused by a dam break in a channel with gradual contraction and expansion.
- Compare the numerical results against the experimental results obtained by the European Commission’s The European Concerted Action on Dam Break Modeling (CADAM)\(^1\).

The schematic diagram of the channel setup for the experiment is shown in the figure 6.8. The channel is 19.3m long and 0.5m wide. The first 6.10m of the channel is specified for the reservoir with an initial static depth of 0.3m. The rest of the channel contains water with depth of 0.003m. The constriction and expansion are tapered at 45 degree with the channel wall. The experimental data was obtained by measuring the water depth and velocity of flow at stations S1, S2, S3 and S4. S1 is located 1m upstream of the dam, S2, S3 and S4 6.10m, 8.80m and 10.50m downstream of the dam respectively. Data was recorded from time \(t = 0\) to \(t = 10\) seconds for every 0.04 seconds (Crowder \textit{et al.}, 2004b).

\[\textbf{Figure 6.8: Schematic illustration of the contraction and expansion test.}\]

\(^1\)The CADAM project was set in motion by the European Union to investigate the methods and their use for simulation and prediction of the effects of dam failures.
The data for Riemann2D was setup as mentioned in the above paragraph. The Manning’s roughness $n = 0.01$ is used throughout the channel. The flow is simulated using HLL solver in conjunction with Minmod limiter. Transmissive boundary condition is used on the downstream boundary of the channel.

Figures 6.9 and 6.10 present the comparison of the numerical and experimental results of a dam-break in a converging and diverging channel.

Figure 6.9(a) shows the results at station S1. It is clear that initially the water level for the experimental results falls earlier as compared to the numerical results.
Figure 6.9: Comparison of the numerical and experimental results for flow in a converging and diverging channel at stations (a) S1 (b) S2.
Figure 6.10: Comparison of the numerical and experimental results for flow in a converging and diverging channel at stations (a) S3 (b) S4.
The reason for this is that during experiment, it takes some time to completely remove the gate, while the dam-break phenomenon in the Riemann2D is instantaneous. The results between 1 and 6 seconds are in close agreement but after 6 seconds the stage from experimental results is slightly higher than that for the numerical ones. This is because the reflection wave from the converging walls decelerates the flow at S1 more quickly during experiment as compared to the analytical solution.

Figure 6.9(b), 6.10(a), and 6.10(b) show the results at station S2, S3 and S4 respectively. It is clear that as we move downstream of the channel, the difference of time for the wavefront of both experimental and analytical results to reach the respective station reduces. This is because the effect of gate opening time reduces as the time progresses. The results at station S2, S3 and S4 are more accurate than that of Canalflow \(^1\) by Schende (2006). At station S3, particularly, canalflow has failed completely to capture the shock, while Riemann2D is reasonably well at this station.

### 6.3 Tests over erodible bed

The following tests are conducted for erodible bed cases.

- Dam-break over erodible bed,
- Dam-break over erodible bed with wetting and drying,
- Flow in a channel with Spur Dykes

#### 6.3.1 Dam break over erodible bed

The purpose of this test is to:

- Check the performance of Riemann2D over erodible bed.
- Study the effects of bed size material on the hydraulics of flood wave generated by dam break.
- Compare the results computed by Riemann2D with that by Cao et al 2004.

\(^{1}\)Canalflow is a hydraulic model for simulating one dimensional, unsteady flow in canal network
Chapter 6: Validation of Riemann2D

For this test the same configurations are taken as that by Cao et al. (2004). The channel length is taken as 50km, and the dam is located at the middle of the channel (x = 25km). The initial static water depths are h1 = 40m on the upstream and h2 = 2m on the downstream of the dam. The test is conducted on flat bed with Manning roughness n = 0.03. For the mobile bed cases, to study the effect of size of the bed material, two sediment particle diameter, i.e d = 4 and 8 mm, are considered.

Figures 6.11-6.14 shows the water surface and the bed profiles for a dam break on a fixed bed and mobile beds with two distinct sediment diameters (d=4, 8mm). The following observations are concluded from figures 6.11-6.14.

• The free surface profile is considerably affected by the bed mobility.

• For the mobile beds a hydraulic jump is generated at the original dam site, which depresses progressively as it propagates upstream and disappears eventually.

• Initially the forward wavefronts propagate slowly over the mobile beds as compared to the one over fixed bed, but the wavefronts for both the mobile and fixed bed cover the same distance after 20 minutes. After 20 minutes the forward wavefront for the mobile beds becomes faster than that over the fixed bed.

• The finer the bed material, the more is the bed erosion which causes the more deviation from the fixed bed case.

The same observations were made by Cao et al. (2004) for his one dimensional erosion model.
Figure 6.11: Evolution of free surface profile and bed after (a) 20s and (b) 30s.
Figure 6.12: Evolution of free surface profile and bed after (a) 40s and (b) 1min.
Figure 6.13: Evolution of free surface profile and bed after (a) 2min and (b) 8min.
Figure 6.14: Evolution of free surface profile and bed after (a) 14min and (b) 20min.
6.3.2 Dam-break over erodible flat bed with wetting and drying

In this test the dam break is simulated over a movable bed with dry bed case on the downstream side.

The purpose of this test is to see the performance of Riemann2D while dealing with erodible bed on wet and dry bed.

The domain for this test is 1000m long and 500m wide. The dam is located at the midpoint of the domain in the longitudinal direction (x = 500m). The depth of the reservoir is 4m on the upstream side while the bed is completely dry on the downstream side. Manning’s n = 0.03 and sediment diameter d = 4mm are used for this test.

Figure 6.15 shows the results for a dam-break problem on erodible flat bed. The following observations are made about the results:

- If we compare the results of erodible flat bed in figure 6.15(a) with fixed flat bed as shown in figure 6.2(a), it is clear that the wave front moving along the longitudinal direction for the fixed bed is considerably faster than that for the mobile bed. This phenomenon was also observed for the dam-break problem without wetting and drying problem.

- The wavefront in the transverse direction for the mobile bed is faster than that for the fixed. This is opposite to the observation discussed above. The reason behind this is that for a mobile bed due to the eroded bed, a part of x-component of the momentum is transferred to the y-direction and thus adding acceleration to the wavefront in the transverse direction.

- Figure 6.15(b) shows the bed erosion caused by the dam-break flow. Scour holes are observed at the points where the surge wave is created.

These observations show the ability of Riemann2D to handle with erodible bed cases with wetting and drying problem.
Figure 6.15: Dam break after 24s on flat erodible bed with wetting and drying: (a) free surface elevation (b) eroded bed (c) velocity vectors
6.3.3 Flow in Channel with Spur Dykes

In this section the numerical results, for flow over erodible bed in a channel with spur dykes, produced by Riemann2D are compared with experimental data by Muto et al. (2003) and numerical results of a 3D model by Nakagawa et al. (2004). The experiment was carried out in a straight compound channel with a slope of 1/700 as shown in the figure 6.19. Ten embayements are formed by replacing some part of the flood plain area with nine consecutive spur dykes. The initial bed is covered with 10cm-thick artificial sands that have a mean diameter $d_{50} = 1.54\text{mm}$. The hydraulic condition used in the experiment and thereof imposed in the simulation are given in the table 6.2.

<table>
<thead>
<tr>
<th>Discharge</th>
<th>Water Depth</th>
<th>Mean Velocity</th>
<th>Friction Velocity</th>
<th>Froude Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.23 l/s</td>
<td>4.30 cm</td>
<td>27.34 cm/s</td>
<td>2.45 cm/s</td>
<td>0.42</td>
</tr>
</tbody>
</table>

![Figure 6.16: Schematic illustration of the contraction and expansion test.](image)

In the experiment, severe scour holes have been observed near the head of the first and second spur dykes, and the second one is a little deeper than the first one. Near the heads of the rest of spur dykes, the scour holes are much smaller. This phenomenon has been properly reproduced by Riemann2D. The bed erosion near the first two spur dykes are shown in the figure 6.19 for the experiment and the 3D model and in the figure 6.18 for Riemann2D.

After comparing the experimental, the 3D $k-$ model and Riemann2D results, the following observation are made:
Chapter 6: Validation of Riemann2D

Figure 6.17: Eroded bed near the first two spur dykes, experimental results (top) and numerical results of the 3D model (bottom).

Figure 6.18: Eroded bed near the first two spur dykes, simulated by Riemann2D.
Figure 6.19: Velocity vectors in the channel near the first two spur dykes.

- The location of the deepest hole simulated by Riemann2D, is near the second spur dyke which is the same for experimental and the 3D model.

- The deepest hole resulted from the experiment is 4.37cm and from the 3D model the computed value is 3.99cm, while the value computed by Riemann2D is 3.29cm. The smaller value of the scour hole from Riemann2D may be attributed to the absence of vertical component of velocity and turbulence term in the two dimensional model.

- The slope of the scour hole from the simulated results by Riemann2D are close enough with the experimental results while the simulated slope of the scour hole by the 3D model is small as compared to the experimental results. This might be the result of shock capturing feature of Riemann2D, that can track discontinuities more accurately as compared to the traditional methods.

These observations show that the performance of the proposed 2D erosion model. It can be concluded that the results from the Riemann2D reasonably represents the eroded bed.
Chapter 7

Application of Riemann2D

7.1 Introduction

In this chapter the application of Riemann2D to a real world problem is discussed. First the model is applied to Malpasset Dam, France to see the performance of the model. For Malpasset Dam, that failed in December 1959, the local police recorded the maximum water levels at various points near the banks after the dam failure. The maximum water levels for the simulated flow at these locations are compared with the recorded data. In the next section the model is applied to Warsak dam located in the Khyber Pukhtoonkhwa province of Pakistan.

7.2 Malpasset Dam Break

7.2.1 Brief History

The Malpasset Dam was built between 1951 and 1954 for irrigation and the storage of drinking water. It was located in the Reyran river valley (French Riviera), in the Department of Var, 12 km upstream of Frejus. The dam was a double curvature dam with a crest length of 223m and 66.5 m maximum height. The dam failed explosively in December 1959, causing the loss of about 433 persons in the town of Frejus.
Figure 7.1: View of the Malpasset Dam break after 12 days of the failure.
7.2.2 Topography

The topography of the area has changed significantly after the dam break due to the debris of the dam carried by the flow and substantial changes in the valley bathymetry in the last 50 years. For these reasons, a map (carte 1/2,000 IGN map of saint-Tropez n°3, dated 1931) with an earlier version, made available from Electricité De France (EDF), have been used for the dam break simulation.

Figure 7.2 shows the topography of the Reyran river valley. The overall dimensions of the area is 17,500 m x 9,000 m. The dam is located in a sunken valley with two narrow bends downstream of the dam. The river then enters the flat plain and finally discharges into the sea. The elevation levels ranges from -20 m abs (absolute) at the sea to +100 m abs taken as the free surface elevation at the reservoir.
Figure 7.2: The old version of the topographic map of the Reyran river valley, that was digitized by EDF.
Figure 7.3: Bed topography of the computational domain for Malpasset Dam.
7.2.3 Field Data

Three transformers were destroyed by the flood wave which resulted in electric shut-downs. The exact times for these shutdowns are known. It could be reasonably assumed that the electric shutdown occurred at the time when the flood wave reached the electric transformers. The location and the flood wave arrival time for each of these transformers are shown in the table 7.1.

<table>
<thead>
<tr>
<th>Electric Transformer</th>
<th>X Coordinates (m)</th>
<th>Y Coordinates (m)</th>
<th>Time of Wave Arrival (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5550</td>
<td>4400</td>
<td>112</td>
</tr>
<tr>
<td>B</td>
<td>11900</td>
<td>3250</td>
<td>1131</td>
</tr>
<tr>
<td>C</td>
<td>13000</td>
<td>2700</td>
<td>1244</td>
</tr>
</tbody>
</table>

Figure 7.4: Location of the seventeen points selected from the points surveyed by the police near the banks.
Chapter 7: Application of Riemann2D

Table 7.2: Coordinates and peak water depths of the points surveyed by police

<table>
<thead>
<tr>
<th>Points surveyed by police</th>
<th>X Coordinates (m)</th>
<th>Y Coordinates (m)</th>
<th>Peak water level (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>4913.1</td>
<td>4244.0</td>
<td>79.15</td>
</tr>
<tr>
<td>P2</td>
<td>5159.7</td>
<td>4369.6</td>
<td>87.20</td>
</tr>
<tr>
<td>P3</td>
<td>5790.6</td>
<td>4177.7</td>
<td>54.90</td>
</tr>
<tr>
<td>P4</td>
<td>5886.5</td>
<td>4503.9</td>
<td>64.70</td>
</tr>
<tr>
<td>P5</td>
<td>6763.0</td>
<td>3429.6</td>
<td>51.10</td>
</tr>
<tr>
<td>P6</td>
<td>6929.9</td>
<td>3591.8</td>
<td>43.75</td>
</tr>
<tr>
<td>P7</td>
<td>7326.0</td>
<td>2948.7</td>
<td>44.35</td>
</tr>
<tr>
<td>P8</td>
<td>7451.0</td>
<td>3232.1</td>
<td>38.60</td>
</tr>
<tr>
<td>P9</td>
<td>8735.9</td>
<td>3264.6</td>
<td>31.90</td>
</tr>
<tr>
<td>P10</td>
<td>8628.6</td>
<td>3604.6</td>
<td>40.75</td>
</tr>
<tr>
<td>P11</td>
<td>9761.1</td>
<td>3480.3</td>
<td>24.15</td>
</tr>
<tr>
<td>P12</td>
<td>9832.9</td>
<td>2414.7</td>
<td>24.90</td>
</tr>
<tr>
<td>P13</td>
<td>10957.2</td>
<td>2651.9</td>
<td>17.25</td>
</tr>
<tr>
<td>P14</td>
<td>11115</td>
<td>3800.7</td>
<td>20.70</td>
</tr>
<tr>
<td>P15</td>
<td>11689.0</td>
<td>2592.3</td>
<td>18.60</td>
</tr>
<tr>
<td>P16</td>
<td>11626.0</td>
<td>3406.8</td>
<td>17.25</td>
</tr>
<tr>
<td>P17</td>
<td>12333.7</td>
<td>2269.7</td>
<td>14.00</td>
</tr>
</tbody>
</table>

The local police surveyed 100 points along the left and right banks and observed the maximum water marks. Among these points, 17 points provided by EDF are considered here. The location and the peak water level for these points are shown in the table 7.2. Location of the electric transformers and the points surveyed by the police are shown in the figure 7.4.

7.2.4 Conditions for Simulation

To apply a model to simulate the results of a real world problem, it is not possible to input the real field conditions. However effort must be done to achieve a scenario which closely agrees with the real conditions. To simulate the Malpasset Dam break problem using Riemann2D, the following conditions are used:
Chapter 7: Application of Riemann2D

- The arch dam is replaced by a straight line with coordinates (4701.18 m, 4143.41 m) and (4655.5 m, 4392.10 m).
- The concrete debris are not taken into account.
- The dam is assumed to be completely destroyed.
- The reservoir level is constant and equal to +100 m ABS.
- The base flow of the Reyran river is not considered in the simulation.
- The inflow to the reservoir was not known, so no input discharge is imposed.
- The Manning’s $n$ for the area is estimated to be 0.03.
- The dam break is instantaneous.

Mesh

In order to apply Riemann2D, the numerical scheme needs the adoption of an unstructured triangular grid. Strong irregularities of the computational domain require a particular attention in the spatial discretization step. The large extension and the non-uniformity of the simulated area requires a raised number of elements; particularly, in the area downstream of the dam, a local refinement is necessary in order to properly describe the bottom valley irregularities. An unstructured triangular mesh containing 76716 triangles is produced as shown in the figure 7.9 using ArgusOne, a commercial mesh generation package.
Figure 7.5: Unstructured triangular mesh for Malpasset Dam, showing the local refinement of the mesh in the valley downstream of the dam where the bed variation is strong.
Chapter 7: Application of Riemann2D

Numerical Scheme

The flow is simulated using HLL solver in conjunction with Minmod limiter.

7.2.5 Results

Figure 7.6 - 7.8 presents the water surface elevation, flow depth and velocity vectors for Malpasset dam, 2800 seconds after the dam break.

Figure 7.6: Water surface elevation of Malpasset dam, 2800 seconds after the dam break.
Figure 7.7: Water depths of Malpasset dam, 2800 seconds after the dam break.
Figure 7.8: Velocity vector, representing the speed and direction of the flow, 2800 seconds after the dam break.
Figure 7.2.3 presents the comparison of the simulated results from Riemann2D and the observed points for the maximum water levels at the banks of the Reyran river after the dam break. It can be seen that the results from Riemann2D are in reasonable agreement with the observed data.

![Comparison of simulated and observed water levels](image)

Figure 7.9: Comparison of the simulated results from Riemann2D and the observed points for the maximum water levels at the banks of the Reyran river after the dam break.
Table 7.3: Coordinates and flood wave arrival times of the electric transformers

<table>
<thead>
<tr>
<th>Electric Transformer</th>
<th>X Coordinates (m)</th>
<th>Y Coordinates (m)</th>
<th>Observed Time of Wave arrival (s)</th>
<th>Simulated Time of Wave arrival (s)</th>
<th>Difference (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>5550</td>
<td>4400</td>
<td>112</td>
<td>105</td>
<td>7</td>
</tr>
<tr>
<td>B</td>
<td>11900</td>
<td>3250</td>
<td>1131</td>
<td>1165</td>
<td>34</td>
</tr>
<tr>
<td>C</td>
<td>13000</td>
<td>2700</td>
<td>1244</td>
<td>1270</td>
<td>26</td>
</tr>
</tbody>
</table>

Table 7.3 shows the comparison between the simulated and observed time of the flood wave arrival at the electric transformers. The simulated flood wave arrival at the transformer A is slightly earlier as compared to the observed one. This might be due to the assumption of the instantaneous dam break failure for the simulated results. For the rest of the two transformers the simulated flood wave arrives at later times than that of the observed ones. Still, the simulated results closely agree with the observed data.

7.3 Warsak Dam-Pakistan

7.3.1 Brief History

The gigantic multi-purpose Warsak Dam is situated 30 kms north-west of Peshawar in the North West Frontier Province of Pakistan as shown in the figure 7.10. The dam was built in 1961 on Kabul river at Warsak. It has a total generating capacity of 240,000 kw and irrigates 110,000 acres of land. The dam is a concrete dam with length of 1000m and height of 80m.
Figure 7.10: Location map of Warsak Dam, Peshawar.
7.3.2 Topography

The topographic data for the study area was acquired from USGS (United States Geological Survey) in the form of Digital Elevation Models (DEMs) that consist of a raster grid of regularly spaced elevation values. The coordinate system of the data was changed from WGS 1984 (Latitude and Longitude) to National Grid Coordinate System (meters). ArcGIS was used to convert the raster data to feature data set to make the data compatible with Riemann2D.

Figure 7.11 shows the topography of the study area. The overall dimension of the area is 50 km x 50 km (approx). The dam is located in a narrow valley between the mountain range of Khyber. On the down stream of the dam, the Kabul river flows through the mountain range for about 4 km and enters the flat plain of Mohmand agency and Charsadda district. The elevation levels ranges from +440m at the hills nearby the reservoir to +300 m at downstream of the area. As the maximum level of the reservoir is +440m ABS, area with bed levels higher than +440 m is extracted from the domain.
Figure 7.11: Bed topography of the computational domain for Warsak Dam.
7.3.3 Conditions for simulation

To apply a model to simulate the results of a real world problem, it is not possible to input the real field conditions. However effort must be done to achieve a scenario which closely agrees with the real conditions. To simulate the Warsak Dam break problem using Riemann2D, the following conditions are used:

- The concrete dam is replaced by a straight line.
- The concrete debris are not taken into account.
- The dam is assumed to be completely destroyed.
- The reservoir level is constant and equal to +440 m ABS.
- As the topography of the reservoir is not included in the data, a constant bed slope is considered which changes from +390m at the upstream of the reservoir to +360m at the heel of the dam.
- The base flow of the Kubal river and the tributary rivers is not considered in the simulation.
- The inflow to the reservoir is not considered, so no input discharge is imposed.
- The Manning’s $n$ for the area is estimated to be 0.03.
- The dam break is instantaneous.

Mesh

An unstructured triangular mesh containing 90416 triangles is produced as shown in the figure 7.12 again using ArgusOne, a commercial mesh generation package. The mesh is refined near the downstream of the dam where the bed irregularities are very strong.
Figure 7.12: Unstructured triangular mesh for Warsak Dam, showing the local refinement of the mesh in the valley downstream of the dam where the bed variation is strong.
Chapter 7: Application of Riemann2D

Numerical Scheme
The flow is simulated using HLL solver in conjunction with Minmod limiter.

7.3.4 Results

Figure 7.13 - 7.21 presents the water surface elevation, depth and velocity vectors for Warsak dam break after 30 minutes, 1 hour and 2 hours. The following observations are made about the results:

- No negative depths are produced in the results. This shows the performance of the algorithm 5.1 discussed in chapter 5, that prevents the negative depths produced by the slope limiters over variable topography.

- The volume of water has been conservative throughout the simulation. The total volume of water in the domain is 364.43 million cubic meters.

- The energy of the whole system has been dissipative.

- The wetting/drying and the drying/wetting process near the banks and other island points has been observed closely and is very stable.

- Very high velocities are observed near the wet/dry fronts where the flow depth is less than the minimum depth, taken as 0.001m, but these velocities are not affecting the overall solution as the cells with depth less than the minimum depth are considered as dead cells resetting their velocity as zero.

- Overall, the water surface, depth and velocity profile shows a very good representation of the actual phenomenon.

- The results can be used with confidence for disaster mitigation strategies in an unlikely event of the dam break.
Figure 7.13: Water surface elevation 30 minutes after the dam break.
Figure 7.14: Water depths 30 minutes after the dam break.
Figure 7.15: Velocity vector, representing the speed of the flow, 30 minutes after the dam break.
Figure 7.16: Water surface elevation 60 minutes after the dam break.
Figure 7.17: Water depths 60 minutes after the dam break.
Figure 7.18: Velocity vector, representing the speed of the flow, 60 minutes after the dam break.
Figure 7.19: Water surface elevation 2 hours after the dam break.
Figure 7.20: Water depths 2 hours after the dam break.
Figure 7.21: Velocity vector, representing the speed of the flow, 2 hours after the dam break.
In this chapter a comprehensive summary of this work, followed by the conclusion is presented. At the end of the chapter, recommendations and potential future work in this area, that could not be carried out here, is presented.

### 8.1 Summary

The main contribution of this work is to extend the object oriented model, Riemann2D, and show the significance of object oriented technology in research work. The model is thoroughly reviewed and the mathematical, physical and numerical constraints of the model are identified and some simple algorithms are introduced to overcome these constraints. The model is extensively tested and the results are compared with analytical and experimental results. Also the model is validated with the actual data from a flood wave caused by breaking of the Malpasset dam in 1957. Finally, the model is applied to study the flood waves caused by Warsak Dam, Peshawar Pukhtoonkhwa, Pakistan.

Each chapter in this thesis present features that have significantly contributed to this work.

**Introduction (Chapter 1):** In this chapter the introduction, motivation, aims and objectives and methodology of this research work is presented.
Chapter 8: Summary, Conclusions, Recommendations and Future Work

Literature Review (Chapter 2): In this chapter, literature review on the relevant methods of modelling hyperbolic and shallow water problems are presented. Initially, in this chapter the fundamental of PDEs and then selection of numerical method are discussed. The sections in this chapter provided a brief background of the work done in the field of hyperbolic problems, high-resolution methods, shallow water equations, erodible bed hydraulics and object-oriented modelling.

Numerical Methods (Chapter 3) This chapter helped to understand the finite volume formulation of the one- and two-dimensional hyperbolic problem, followed by discussion on Riemann problem and its solution. Later in the chapter, traditional numerical methods followed by the high-resolution methods used in Riemann2D were discussed. This chapter helped to understand the contribution to the generic part of the model.

Free-Surface Shallow Water Flows (Chapter 4): This chapter presented the theory of the shallow water equations. Initially, in this chapter the derivation of the two dimensional shallow water equations from Saint Venaint Equations was presented, followed by the discussion of hyperbolic characteristics of the shallow water equations. Also generic Riemann solvers applied to shallow water problem were presented. This chapter helped to understand the shallow water part of the model.

Extension of Riemann2D (Chapter 5) This chapter showed the contribution towards the extension of Riemann2D. Initially the extension of Riemann2D to deal with flow over mobile bed problems was discussed. Later in the chapter some simple new algorithms were introduced to deal with flow over variable topography, wet/dry and dry/wet fronts, negative depths produced due to limiters, no flow conditions and problems due to friction terms. These algorithms made the model capable of predicting flood waves over any type of topography with any conditions in a real situation.

Validation of Riemann2D (Chapter 6) Many standard test cases, present in the literature, were conducted and presented in this chapter to see the performance of the model. The results were compared with the numerical, analytical and experimental results.
**Chapter 8: Summary, Conclusions, Recommendations and Future Work**

**Application of Riemann2D (Chapter 7)** In this chapter the application of Riemann2D to a real world problem was discussed. Early in the chapter, the results of the model applied to Malpasset Dam, France were presented. The maximum water levels for the simulated flow at 14 points were compared with the recorded data made available from EDF. In the last section of the chapter the model was applied simulate the flood waves caused by breaking of Warsak dam located in the Khyber Pukhtoonkhwa province of Pakistan.

**Summary, Conclusions, Recommendations and Future Work (Chapter 8)** This chapter

### 8.2 Conclusion

In this research work, an object-oriented framework and model, Riemann2D, is extended with the addition of an erosion model, called CaoPenderErosion model, and the model is applied to dam break flow over fixed bed. Some simple algorithms are included that can handle various problems, while dealing with flow over variable topography and wetting and drying problems. The model is extensively tested before it is applied to a real world problem.

The following points are concluded after carrying out this work.

- The model is extended by adding a package Called CaoPenderErosion package. This package is a subclass of ShallowWater package and thus using the common components of ShallowWater and Generic package. By adding this package, the Riemann2D’s capability is enhanced to solve shallow water problems over mobile beds.

- The bed mobility considerably affects the flow hydraulics in both directions.

- The new wetting and drying algorithms are very simple and efficient and can track wet/dry fronts on any type of topography with no instabilities in the solution.

- To ensure numerical stability, a limitation over the friction term is imposed beside the time step restriction using CFL condition.
• The model is thoroughly tested and simulated results are in close agreement with the available analytical, numerical and experimental results. Thus the model can be used with greater reliability as compared to the models that uses traditional methods for solving hyperbolic problems.

• The simulated results of Malpasset dam, that are in close agreement with the field data, shows the performance of Riemann2D.

• The model was applied to Warsak Dam, and has produced good results on a topography that varies from abruptly varying bathymetry in the mountainous valley near the dam site, to smooth bathymetry on the downstream area.

• The use of Geographic Information System (GIS) can play a vital role in improving the accuracy of the results by using the high resolution remotely sensed topographic data of the flooding area.

8.3 Future Work

After carrying out this research work, some new work is identified that could be carried out to improve the performance of Riemann2D in particular and some more work in the area of CFD in general.

• Integration with Geographic Information Systems (GIS): GIS is an organized collection of computer hardware, software, geographic data, and personnel designed to efficiently capture, store, update, manipulate, analyze, and display all forms of geographically referenced information. It combine relational databases with spatial interpretation and outputs, often in form of maps.

In this work remotely sensed topographic data of the Warsak Dam has been used. However, the data was processed in a GIS package, ArcGIS, before incorporating in the model. This type of linking the data is called loose coupling/linking. In loose coupling the model and GIS are two independent entities. There is no common interface between them. They are integrated via data exchange using either ASCII or binary data format without a common user interface. The advantage of loose coupling is that it needs minimal computer programming, but the data
conversion between the two packages can be tedious and error prone (Sui & Maggio, 1999). On the other side, full coupling/Integration is an embedded system, in which the model and GIS are embedded in a single manipulation framework (Crosbie, 1996).

Fully integrating Riemann2D with GIS would help Riemann2D to use the GIS database to retrieve and store information. The output from the Riemann2D, stored in GIS database, could be easily analyzed for impact assessment studies of real world problems within a stand alone package.

• Multi-particle size movable bed problems: The approach to extend Riemann2D presented in this work, can be adopted to add multi-particle size problems, like, Hairsine-Rose soil erosion model.

• Further development of the generic model: The development process of Riemann2D has been through studying requirements of a variety of hyperbolic problems. However, with addition of other hyperbolic extension models, new things could be learned and thus improvement could be made to the existing generic code.

• The two dimensional erodible bed model, CaoPender Erosion Model, needs more validation tests. Using digital photogrammetry to capture topographical data from laboratory and field tests for comparison with the numerical results could be useful for the validation of the model. On a large scale, the comparison of the bed evolution from a remotely sensed satellite data of a flood plain and numerically simulated results from Riemann2D, could be a big step forward in the erodible bed hydraulics.

• The performance of Riemann2D, with respect to the time it takes for simulating a problem, could be further improved by utilizing the multiprocessing in Java. Also, some new algorithms (see Murillo et al. (2007)) could be included in the code to increase the conventional time step calculated by the CFL condition and thus reducing the simulation time. This could be really vital in studying long term evolution of floods by reducing the simulation time and thus reducing the computational costs.


References


References


References


