High-performance simulation technologies for water-related natural hazards

Xilin Xia

A thesis submitted for the degree of Doctor of Philosophy

School of Civil Engineering and Geosciences

Faculty of Science, Agriculture and Engineering

Newcastle University

May, 2017

Abstract

Water-related natural hazards, such as flash floods, landslides and debris flows, usually happen in chains. In order to better understand the underlying physical processes and more reliably quantify the associated risk, it is essential to develop a physically-based multi-hazard modelling system to simulate these hazards at a catchment scale. An effective multi-hazard modelling system may be developed by solving a set of depth-averaged dynamic equations incorporating adaptive basal resistance terms. High-performance computing achieved through implementation on modern graphic processing units (GPUs) can be used to accelerate the model to support efficient large-scale simulations. This thesis presents the key simulation technologies for developing such a novel high-performance water-related natural hazards modelling system.

A new well-balanced smoothed particle hydrodynamic (SPH) model is first presented for solving the shallow water equations (SWEs) in the context of flood inundation modelling. The performance of the SPH model is compared with an alternative flood inundation model based on a finite volume (FV) method in order to select a better numerical method for the current study. The FV model performs favourably for practical applications and therefore is adopted to develop the proposed multi-hazard model. In order to more accurately describe the rainfall-runoff and overland flow process that often initiates a hazard chain, a first-order FV Godunov-type model is developed to solve the SWEs, implemented with novel source term discretisation schemes. The new model overcomes the limitations of the current prevailing numerical schemes such as inaccurate calculations of bed slope or friction source terms and provides much improved numerical accuracy, efficiency and stability for simulating overland flows and surface flooding. To support large-scale simulation of flow-like landslides or debris flows, a new formulation of depth-averaged governing equations is derived on the Cartesian coordinate system. The new governing equations take into account the effects of non-hydrostatic pressure and centrifugal force, which may become significant over terrains with steep and curved topography. These equations are compatible with various basal resistance terms, effectively leading to a unified mathematical framework for describing different type of water-related natural hazards including surface flooding, flow-like landslides and debris flows. The new depth-averaged governing equations are then solved using an FV Godunov-type framework based on the second-order accurate scheme. A flexible and GPU-based software framework is further designed to provide much improved computational efficiency for large-scale simulations and ease the future implementation of new functionalities. This provides an effective codebase for the proposed multi-hazard modelling system and its potential is confirmed by successfully applying to simulate flow-like landslides and dam break floods.

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Qiuhua Liang. He guided me into the research area of natural hazards modelling and taught me how to become an independent researcher. Without his patience, encouragement and unrelenting help, I could never finish this thesis. I would also like to thank Professor Fuchu Dai (Beijing University of Technology), for the knowledge and experience about landslides learned from him during the two field trips in China.

I am very grateful to Newcastle University and China Scholarship Council for their financial support in my first three years of PhD. I would also like to thank the Henry Lester Trust and Great Britain China Education Trust, for supporting me financially in my writing-up year.

My thanks also go to my colleagues Jingming Hou, Xiaodong Ming, Reza Amouzgar, Jingchun Wang, Hongbin Zhang, Luke Smith, and Samantha Mahaffey because I could always ask for help from them and discuss about my research with them. I am always grateful to my friends in Newcastle who always give me the support when I need. They are Zheng Wang, Jiajun Chen and Chen Yu. I would like to thank my friend in Durham, Jun Hou, since spending time with him was always inspiring.

I would also like to thank the administrative staff in School of Civil Engineering and Geosciences, SAgE Faculty workshop organisers, librarians and computing technicians for the services they provided. Here I especially thank Melissa Ware, Hannah Lynn, Laura Hanson and Graham Patterson.

Finally I would like to express my sincere gratitude to my mother, father, my mother-in-law, father-in-law and my lovely sister, Tian-Tian. To my beloved wife, Jingyi, who stands by me and always comfort and care me. Their love is the best treasure in my life.

List of Publications

Xia, X., Liang, Q., Pastor, M., Zou, W., and Zhuang, Y. (2013). Balancing the source terms in a SPH model for solving the shallow water equations. *Advances in Water Resources*, 59:25-38

Xia, X., Huang, R., Liang, Q., and Yu, Bin. (2015). A New Physically-Based Simulation Framework for Modelling Flow-Like Landslides . *International Journal of Geohazards and Environment*, 1(3):94-100

Xia, X., Liang, Q. (2016). A GPU-accelerated smoothed particle hydrodynamics (SPH) model for the shallow water equations. *Environmental Modelling and Software*, 75:28-43

Xia, X., Liang, Q., Ming, X., and Hou, J. (2017). An efficient and stable hydrodynamic model with novel source term discretization schemes for overland flow and flood simulations. *Water Resources Research*, 53, doi: 10.1002/2016WR020055

List of Figures

List of Tables

Contents

Chapter 1. Introduction

1.1 Background of the research

Catchments are the places where water moves and converges. They are also the places where people live and work. Rivers and streams in catchments provide not only drinking water but also water for agriculture, industry and transport. But catchments can also bring disasters to humanity; hazards related to water have always been part of the catchment process, which is the fierce side of face of catchments that people have to live with. Flash floods, debris flows and landslides are among the most common water-related natural hazards.

Each year, water-related natural hazards cause enormous damage to properties and loss of lives. For example, a flash flood that took place on 7 June 2014 in Baghlan, Afghanistan, killed 73-200 people and left 200 missing. The two landslides happened in Argo district, Badakhshan province, Afghanistan on 2 May 2014 claimed at least 350 lives (with a further 2100 still missing), and buried 300 houses (and affected another 1400). On 8 August 2010 in Zhouqu county, Gansu province, China, a debris flow induced by heavy rainfall caused more than 1400 deaths and economic loss over 17.7 million US dollars of estimation.

A number of studies (IPCC [2014]) have predicted that climate change will lead to increasing frequency and magnitude of weather extremes, particularly rainfall. This, in conjunction with more intensive human activities, may substantially increase the risk of flash flooding, debris flows and landslides in steep catchments. These natural hazards are commonly a result of both water and sediment movement within a catchment. They may interact with each other and happen as chains. For example, heavy rainfall may lead to excessive run-off and subsequently flash flooding; it may also raise groundwater table and trigger landslides; rapid surface water flow may erode loose packed material caused by landslides and develop into debris flows as the sediment concentration increases; landslides and debris flows may dam rivers and subsequently cause dam-break floods. These hazard chains can significantly amplify the damages and losses in affected areas. For example, in June 2013 the breach of a landslide dammed lake in the Himalayan Rudraprayag district of Uttarakhand (India) contributed to 4,094 fatalities. On 2nd August 2014 following heavy rainfall, a landslide dammed the Sunkoshi River to form a lake in Sindhupalchok District in Nepal, causing severe flooding; the cascading event destroyed 115 houses, killed 156 people and inundated the powerhouse of a hydropower project and other infrastructure.

Computer modelling has become an indispensable tool to assess the risk and mitigate the impact of water-related natural hazards. They can also be used to perform numerical experiments to better understand the underlying physics of these hazards as well as the impact of environmental changes on them. Traditionally, hydrological models or simplified hydrodynamic models are used to model water and sediment movements and other relevant processes at the catchment scale. The underlying physics of the catchment processes considered in these models are usually significantly simplified. For example, the surface flow routing in those models normally does not consider momentum conservation and the couplings between different processes are usually loose and empirical. Therefore, they generally lack the capability to provide detailed and reliable information that is needed for understanding the complex interactions between hazards and quantifying their risks.

Sophisticated models constituting more physical processes have been successfully applied to assess the hazard risk typically at a localised scale for a single hazard. However, these models are still computationally prohibitive for high-resolution simulations at catchment/city scales. Also they usually do not consider the interaction between hazard components except for the preliminary effort made by Chen et al. [2016], and hence may underestimate the risk to receptors and repeated engineering reconstruction. A sophisticated physically based multi-hazard model that directly considers the interactions between different types of hazards may enable much more realistic assessment of the hazard risk. It can also improve the understanding of the complex interactions between different hazards in a chain to facilitate the design of more effective counter-measures. These sophisticated models can also support scenario simulations and provide more detailed information for analysing the impact of climate and other environmental changes.

A major constraint that hinders the application of sophisticated physically based models at a broader scale is the limitation of computational power. The computing technology has been rapidly advanced in recent years, bringing a great opportunity for improving substantially the performance of the current hazard modelling tools. The most notable breakthrough is the development of general-purpose graphics processing units (GPGPU) for scientific computing. Graphic Processing Units (GPUs) are originally designed exclusively for three-dimensional image rendering that is not complex but highly computationally intensive. GPUs can offer up to 10 to 20 times more computation power compared with the equivalent Central Processing Units (CPUs). In 2007, NVIDIA released the CUDA programming platform designed for general purpose computing. Since then, the computational power of GPUs is becoming readily accessible for scientists and engineers. Its capability of accelerating scientific computing tools has been demonstrated in a variety of disciplines. Thanks to the unprecedented computational power of GPUs, large-scale simulation of surface water and sediment movement can now be conducted on a small server or even a desktop rather than a regional supercomputer centre, which has led to a new opportunity for developing and applying sophisticated physically-based models for multi-hazard simulations.

As a summary, it is essential to develop sophisticated physically based models to simulate multiple water-related natural hazards and related processes, e.g. flash flooding, landslides, excessive soil erosion, debris flows and natural dam outburst floods, which may happen simultaneously or in sequence as a hazard chain in a catchment. The recent rapid development of high-performance computing technology has made this type of simulations to become feasible.

1.2 Terminology

The definitions of flash flood, landslide and debris flow often cause ambiguity in the literature. It is necessary to clarify their definitions here at the beginning of this thesis to avoid confusion. The rest of the thesis will follow the definitions given in this section except for the cases otherwise explicitly stated.

1.2.1 *Flash flood*

There are several definitions for flash flooding as summarised by Sene [2013]. Herein we follow the definition given by the World Meteorological Organisation (WMO): "*flash floods are rapidly rising flood waters that are the result of excessive rainfall or dam-break events. Rain-induced flash floods are excessive water flow events that develop within a few hours (typically less than 6 h) of the causative rainfall event, usually in mountainous areas or in areas with extensive impervious surfaces such as urban areas*". Although most of the flash floods observed are induced by intense rainfall, failure of natural or human-made dams can also cause the release of excessive volumes of stored water in a short period of time with catastrophic consequences downstream.

1.2.2 *Landslide and debris flow*

Generally, landslides are defined as '*the movement of a mass of rock, debris or earth down a slope driven by gravity*' (Dai et al. [2002]). More precise definitions may be given to the sub-types of landslides. The most popular classification of landslides was proposed by Varnes [1978], according to which landslides can be categorised by movement types (falling, toppling, sliding, lateral spreading and flowing) and material types before the initiation of motion (e.g. bedrock, coarse rock debris, rock debris, sand and fine particles). The above definition of 'landslide' is rather broad, which literally includes the flow of debris, i.e. 'debris flow'.

A distinction between landslide and debris flow is made in the statistics of disasters in certain countries such as China and Japan. Such a distinction is justified by the fact that a debris flow may contain a large amount of water and its runout distance can be as far as kilometres. The definition of debris flow given by Takahashi [2014] is used in this work, i.e. '*debris flow is a flow of sediment and water mixture in a manner as if it was a flow of continuous fluid driven by gravity, and it attains large mobility from the enlarged void space saturated with water or slurry*'.

In the remaining part of this thesis, when the term 'landslide' is used, it refers to the general definition given by Dai et al. [2002] excluding debris flow. The term 'debris flow' refers to the definition given by Takahashi [2014]. The term 'flow-like landslide' is also frequently used in this thesis, which refers to those landslides that present fluid flow properties. Flow-like landslides encompass debris flows; but they do not necessarily contain a large amount of water like debris flows.

1.3 Water-related natural hazards as parts of a whole dynamic catchment system

Figure 1.1 illustrates possible links and interactions between those key catchment processes (e.g. Takahashi [2014], Jakob et al. [2005]). Rainfall is considered here as the main driver of water and sediment movement in a catchment. As the rainwater falls onto the ground, part of it will infiltrate into the ground and saturate soil, leading to increase of pore water pressure and reduction of metric suction and subsequently a loss of shear strength and rising ground water table. Subsequently, landslides may be triggered, which may further develop into debris flows through supplement of water and erosion of loose material on the ground. The rainwater remains on the ground surface may form overland flow that will finally converge into channels. If the channels cannot retain the water within their banks, flooding will occur. Channel flows under flood conditions can also erode the channel bed and banks to form debris flows. The erosion and deposition processes following landslides and debris flows can lead to significant topography change, which will in turn alter the flow paths of water and sediment.

Apparently, (flash) floods, landslides and debris flows are all integrated parts of water and sediment movement in a catchment. They interact with each other in two main ways: 1) transform into the other hazards through the change of sediment concentration as a result of additional water and bed erosion, and 2) influence the flow paths and thus the corresponding processes through the change of topography and channel morphology.

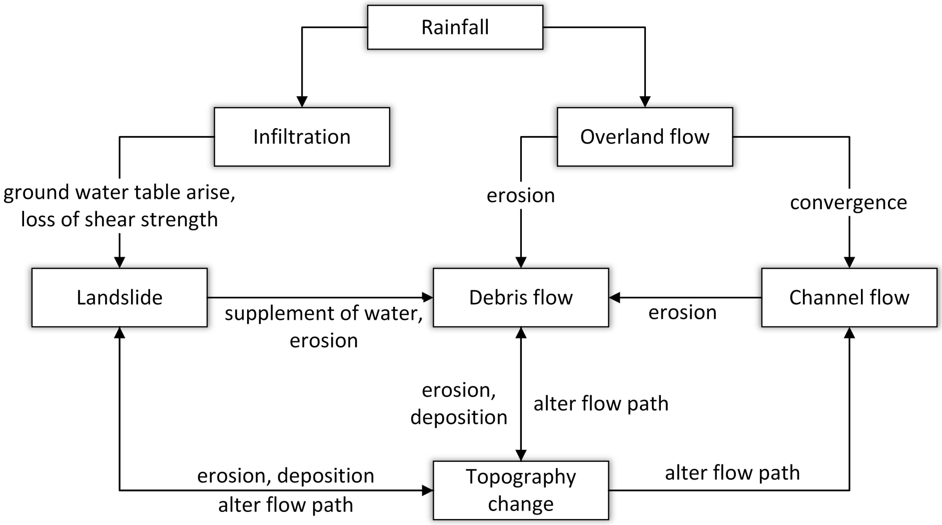


Figure 1.1: The interactions between different processes in catchments.

Based on such an understanding, we propose a concept of a catchment scale multi-hazard model with the following components: 1) A hydrodynamic module that simulates the movement of water and sediment, where a single mathematical framework is used and the rheology adapts according to sediment concentration or other criteria to enable the transitions between (flash) floods, landslides and debris flows. Because the depth of flash floods and flow-like landslides is usually much smaller than their horizontal extent, depth-averaged models will be used to provide a good compromise between accuracy and efficiency to enable large-scale simulations. 2) A sediment-transport module to model the change of sediment concentration due to erosion, deposition and transport processes. 3) An infiltration module to predict the pore-water pressure and groundwater table changes. 4) A slope failure module to represent the initiation of landslides and subsequent supplement of sediment.

The hydrodynamic module is the core of the proposed multi-hazard model because it describes the dynamic process of a hazard chain and all other modules rely on the hydrodynamical information, i.e. depth and velocity.

1.4 Research aim and objectives

Our ultimate goal is to develop a physically based modelling system for simulating multiple water-related natural hazards at a catchment scale (~1000km^2) as described by the concept in section 1.3. The aim of this PhD is to **develop the mathematical framework and high-performance modelling technologies to form the core hydrodynamic component of this physically based catchment-scale multi-hazard modelling system**. The research aim can be achieved through the following objectives:

- **Compare and choose an appropriate numerical method.**

  We set this objective because there are two competing numerical methods: Grid-based Eulerian methods such as the finite volume (FV) method and meshfree computational methods like the smoothed particle hydrodynamics (SPH), both of which have been widely used in hydrodynamic simulations.

- **Develop numerical technologies for efficient and stable overland flow simulations.**

  We set this objective because overland flow plays a key role in the development of and interaction between flash floods, landslides and debris flows.

- **Develop a high-performance multi-hazard modelling framework.**

  This objective comprises the following two sub-objectives:

  1. Develop a unified mathematical formulation for both mass flow and water flow, which can be achieved in two steps: 1) Develop a non-hydrostatic depth-averaged model for flow-like landslides over complex topography based on a global Cartesian coordinate system. 2) Combine the depth-averaged model for flow-like landslides with the Manning's friction law to simulate flood waves and demonstrate the new depth-averaged model's potential to be the unified mathematical framework for both flow-like landslides and water flow.

We set this objective because using a single mathematical framework is essential for simulating different types of flow simultaneously and one of the major difficulties is the unification of coordinate systems. Flow-like landslides are usually modelled on surface fitted curvilinear coordinate systems to account for non-hydrostatic pressure and centrifugal forces due to steep and curved topography while water flow is usually modelled on Cartesian coordinate system. To develop a unified mathematical framework, we need to either model flow-like landslides on Cartesian coordinate system or model water flow on curvilinear coordinate system. We choose the former approach because the Cartesian coordinate system is simple, consistent with the geographic information system (GIS) data format for domain topography.

2. Devise a flexible software framework on GPUs.

We set this objective because a flexible software framework will allow us to easily implement and integrate together different modelling components and using GPUs will make high-resolution catchment-scale simulations to be feasible.

1.5 Thesis outline

The rest of the thesis is organised as follows: Chapter 2 gives a comprehensive literature review in the relevant topics; Chapter 3 introduces a GPU-accelerated SPH model for solving the shallow water equations (SWEs) and its application to practical flooding simulation, compared with traditional FV model; Chapter 4 presents a new efficient and stable shallow flow model based on a first-order FV scheme for overland flow processes; Chapter 5 presents a new non-hydrostatic depth-averaged model for rapidly varying mass flow over complex terrain; in chapter 6, the depth-averaged mass flow model is applied to simulate water flows to demonstrate its potential in providing a unified mathematical framework for multi-hazard modelling and details of the overall structure and implementation of the software framework are also presented; and finally Chapter 7 draws the major conclusions from this work and provides recommendations for future research.

Chapter 2. Literature review

This chapter aims to provide a comprehensive review of the existing literature, covering the relevant topics considered in this thesis.

2.1 Numerical modelling of multiple hazards and their interactions

Very limited research has been made to model multiple hazards in a catchment simultaneously to understand their interactions and quantify their risks (Kappes et al. [2012]). Chen et al. [2016] made an attempt to develop such a multi-hazard model. Their model comprises digital terrain module, a spatial rainfall distribution module, a slope failure prediction module, a debris flow simulation module, and a multi-hazard quantitative risk assessment module. The whole domain is firstly divided into multiple cells based on the digital terrain data. Then the rainfall is spatially distributed into the cells through universal Kriging interpolation. Based on the spatial distribution of the rainfall, the surface runoff and infiltration processes can be analysed to determine the pore-pressure at each cell. Then the slope stability at each cell can be analysed based on Mohr-Coulomb theory. Empirical method is used to determine the occurrence probability and initial volumes of debris flows and their movement is analysed by a depth-averaged model. Finally, the total risk is given by the risk assessment module. This model considers the interactions among rainfall, landslides and debris flow therefore is comprehensive for quantifying multi-hazard risk. However, the assumptions used in this model is very simplified compared with reality. Though claimed to be a physically-based models, the interactions among different hazards are still considered mainly using empirical methods. More research efforts are still needed to develop models that simulate each hazard type and their interactions in a more physically sound manner.

2.2 Hydrodynamic modelling of water and mass movement

2.2.1 *Overland flow*

The water depth of surface water flow is usually much smaller than the length of gravity wave, and so the vertical flow velocities are negligible compared with the horizontal velocities. As a result, the Reynolds-averaged Navier-Stokes equations may be integrated along the vertical axis to derive the following shallow water equations (SWEs). In one spatial dimension, the SWEs may be written as

$$\frac{\partial h}{\partial t} + \frac{\partial hu}{\partial x} = R \tag{2.1}$$

$$\frac{\partial hu}{\partial t} + \frac{\partial hu^2 + \frac{1}{2}gh^2}{\partial x} = ghS - S_f \tag{2.2}$$

where x and t respectively denotes the space and time, h is the water depth, u is the depth-averaged velocity along the flow direction, g is the gravitational acceleration, R represents a source/sink term related to rainfall/infiltration, S is the bed slope and S_f is the basal resistance term which may be formulated by empirical friction laws such as the commonly used Manning or Chezy formula (Chow [1959]) which are typically expressed as functions of velocity and bottom roughness.

The SWEs provide an effective mathematical framework for surface water flows in a catchment. However, the SWEs constitute a non-linear system and analytical solutions can be only sought for limited situations where the boundary conditions and initial conditions are very simple. Therefore, early studies on overland flow modelling are mainly based on simplified equations that can either be solved analytically or using simple numerical methods. The earliest attempt may be traced back to the kinematic wave theory proposed by Lighthill and Whitham [1955]. In the kinematic wave theory, it is assumed that the gravity driving force and resistant force of the flow balance each other so that the flow can locally reach a steady state and the flow velocity solely depends on the relationship between the bed slope and bottom roughness, i.e.

$$ghS - S_f = 0 \tag{2.3}$$

while continuity equation (2.1) remains the same. The resulting equations can be solved analytically in certain situations using the method of characteristics. Henderson and Wooding [1964] derived the analytical solution for simple overland flows driven by steady rainfall and solved the equation numerically for the more complicated cases with time-dependent rainfall intensity. Since 1970s, seeking numerical solutions to the kinematic wave equation has had become increasingly popular and a brief overview can be found in Smith [1980] where the stability and convergence of several finite difference algorithms were discussed.

The validity and limitations of the kinematic wave model for different types of shallow flow hydrodynamics have also been investigated by many researchers. Woolhiser and Liggett [1967] compared the numerical solutions from the kinematic wave equation and the fully shallow water equations with experimental data and introduced a dimensionless parameter called 'kinematic wave number' to indicate the adequacy of the kinematic wave theory. They suggested that the kinematic theory is valid as long as the kinematic wave number is large, which has also been supported by many other studies (e.g. Ponce et al. [1978], Vieira [1983], Singh [2002]). Due to its simplicity and acceptable accuracy when the underlining assumptions are satisfied, the kinematic wave theory is still used in some of the current flood models (e.g. Bates et al. [2010]).

However, the assumptions of the kinematic wave theory may be over simplified in certain situations. Morris and Woolhiser [1980] suggested that the kinematic wave theory becomes less accurate when the Froude number ($Fr = \frac{u}{\sqrt{gh}}$) becomes small, in which case at least the slightly more sophisticated diffusion wave equation should be used. The diffusion wave theory assumes that the resistant force is balanced by gravity driving force (bed slope) and pressure head gradient (water depth slope), i.e.

$$ghS - \frac{\partial}{\partial x}(\frac{1}{2}gh^2) - S_f = 0 \qquad (2.4)$$

Diffusion wave theory is slightly more comprehensive than the kinematic wave theory. Govindaraju [1988] presented details of a diffusion wave model and resulting numerical solutions for overland flows.

Both kinematic wave and diffusion wave theories are 1D in its original appearance and extensions to 2D problems are not popular for a long time since their genesis. This is not a surprise because the emergence of kinematic wave theory and diffusion wave theory mainly attributes to their simplicity and availability of analytical or semi-analytical solutions but analytical solutions to their 2D counterparts are usually not available. But in the last two decades, kinematic wave and diffusion wave theories have been implemented in several 2D numerical models (e.g. Bates and De Roo [2000], Bradbrook et al. [2004], Liu et al. [2004]) because numerically solving these simplified governing equations is more straightforward and less computational demanding than solving the full SWEs for certain applications.

The limitations of the kinematic wave and diffusion wave theories originate from their over-simplified description of the physics of shallow flows, e.g. overland flows and rapidly varying flood waves. In both of these theories, an equilibrium state between basal resistance and bed/depth slope must be assumed as a priori. In reality, an overland flow may usually involve highly transient processes such as water running up/down a slope and shock waves/hydraulics jumps. Several comparative studies (Cea et al. [2010], Costabile et al. [2012]) have suggested that the simplified models based on kinematic wave or diffusion wave theories may lead to significant errors for overland flows over complex topography. Furthermore, these simplified models may not always lead to more computationally efficient simulations. For a diffusion wave model, the time step needed to stabilise the numerical scheme is proportional to the square of cell size while the Courant–Friedrichs–Lewy (CFL) stability condition for solving the full SWEs model only requires the time steps to be proportional to the cell size (Hunter et al. [2005]). Therefore the time steps for a diffusion wave model may be significantly smaller than that for a full SWEs model when the simulation resolution increases, leading to a much slower simulation (e.g. Cea and Blade [2015]).

A more appropriate alternative for overland flow modelling is to solve the full SWEs. In the last two decades, due to the substantially increased computational power and rapid progress on developing more efficient numerical methods for conservation laws, models based on the solution to the full SWEs have become increasingly popular for a wide range of application involving flood modelling (e.g. Fiedler and Ramirez [2000], Sanders et al. [2008], Simons et al. [2014], Yu and Duan [2014], Cea and Blade [2015], Rousseau et al. [2015], Caviedes-Voullième et al. [2012], Liang et al. [2015a]). Compared with those simplified models, the models based on full SWEs are more accurate and robust in handling complex flow hydrodynamics and topography. Compared to those models solving the 3D Navier-Stokes equations, models based on the full SWEs are able to represent the essential information of the flows (i.e. unsteady water depth and flow velocity) accurately with much less computational effort. Research effort in recent years for overland flow and flood modelling has mainly focused on developing numerical methods for solving the SWEs and improving their computational efficiency, which will be reviewed in more details in sections 2.3 and 2.4.

2.2.2 *Flow-like landslide*

Over the last few decades, different types of computer models have also been developed and used to predict the runout distance and velocity of flow-like landslides. The simplest approach is based on the lumped mass method (e.g. Hutchinson [1986], Federico and Cesali [2015]), in which a landslide or debris flow is idealised using a single slab of mass. Obviously, the internal deformation and lateral spreading of a landslide or debris flow cannot be represented, restricting its wider applications.

A more physically-based approach is based on the continuum theory which takes into account the conservation of both mass and momentum. In most of the real-world cases of flow-like landslides, the vertical dimension is much smaller than the horizontal extent and so the depth-averaging procedure used for deriving the SWEs from the Navier-Stokes equations can be applied, resulting in a formulation similar to the SWEs (2.1-2.2). Savage and Hutter [1989] formally derived a set of depth-averaged governing equations for the first time for modelling granular flows on inclined slopes. Since then, numerous research efforts have been made to increase the predicting capabilities of their depth-averaged model mainly in two aspects: 1) improving the understanding of the underlying physics of flow-like landslides and debris flows; and 2) improving the mathematical formulation of depth-averaged equations to more accurately represent the gravity driving force and the flow dynamics over complex real-world topography.

Regarding the underlying physics of flow-like landslides including debris flows, most models assume that the flowing material is at a single phase and adopt different rheological laws for different types of flow material. In the depth-averaged equations, the fluid rheology is packed as basal resistance source terms added to the momentum equations. A summary of the commonly used rheological laws for flow-like landslides and debris flows is given in Table 2.1. The Newtonian rheology represents the most basic one, which is normally not adequate to model real world flow-like landslides. If the material is assumed to yield after reaching a critical stress and behaves like Newtonian fluid after yielding such as the situation of hyper-concentrated flow containing a large proportion of fine particles, the Bingham fluid model can be applied. The Bingham fluid model was firstly derived by Yano and Daido [1965] and has since then been further developed and applied to simulate debris flows in real-world applications (e.g. Hungr [1995], Medina et al. [2008], Pastor et al. [2014], Pastor et al. [2015]). The Herschel-Bulkley fluid model (Coussot et al. [1998]) adds an additional parameter to the original Bingham fluid model to account for a wider range of debris flows. Takahashi [2014] introduced a dilatant fluid

Table 2.1: Mathematical expressions for rheological laws in depth averaged equations[1].

Names of rheological laws	Original Equations	Depth-averaged equations
Newtonian	$\tau = \mu \dfrac{\partial v}{\partial z}$	$\tau_b = 3\mu \dfrac{u}{h}$
Bingham[2]	$\tau = \tau_0 + \mu \dfrac{\partial v}{\partial z}$	$u = \dfrac{\tau_b h}{6\mu}(1 - \dfrac{\tau_0}{\tau_b})^2(2 + \dfrac{\tau_0}{\tau_b})$
Herschel-Bulkley[3]	$\tau = \tau_0 + \mu(\dfrac{\partial v}{\partial z})^n \ n \leq 1$	$\tau_b = \tau_0[1 + 1.93(\dfrac{\tau_0}{\mu}(\dfrac{h}{u})^n)^{-0.9}]$
Dilatant	$\tau = \mu(\dfrac{\partial v}{\partial z})^n \ n > 1$	$\tau_b = 3\mu(\dfrac{u}{h})^n$
Frictional	$\tau = \mu_f \sigma$	$\tau_b = \rho h(g \cos\theta + \dfrac{u^2}{r})\mu_f \dfrac{u}{\lVert u \rVert}$
Voellmy	$\tau = \mu_f \sigma \dfrac{v}{\lVert v \rVert} + \xi \rho v \lVert v \rVert$	$\tau_b = \rho h(g \cos\theta + \dfrac{u^2}{r})\mu_f \dfrac{u}{\lVert u \rVert} + \xi \rho u \lVert u \rVert$
Plastic	$\tau_b = \tau_0 \dfrac{v}{\lVert v \rVert}$	$\tau_b = \tau_0 \dfrac{u}{\lVert u \rVert}$

[1] τ is the shear stress, σ is the normal stress, τ_b is basal resistant stress, ρ is density, v is velocity, u is the depth-averaged velocity, μ is viscous coefficient, μ_f is friction coefficient, θ is the slope angle, g is gravitational acceleration, r is basal curvature, ξ is Voellmy coefficient, n is a dimensionless parameter.

[2] For Bingham rheology, the depth-averaged equation of the basal resistant stress is written in an implicit form in which τ_b is the root of the equation. Detailed derivation can be found in Pastor et al. [2015].

[3] For Herschel-Bulkley rheology, rigorous derivation of even an implicit depth-averaged equation like the Bingham fluid is non-trivial. Herein we included the empirical equation obtained by Laigle and Coussot [1997].

model to consider the collisional stresses between the grain particles in stony debris flows.

Frictional rheology may be more appropriate for the relatively dry granular material. The simplest frictional rheology is based on the Mohr-Coulomb theory, where the friction coefficient is assumed to be constant. Savage and Hutter [1989] derived a depth-averaged model by assuming Mohr-Coulomb internal rheology. This assumption leads to a basal friction source term with a constant friction angle and a nonlinear lateral pressure coefficient. Pouliquen and Forterre [2002] proposed a new velocity-dependent empirical friction law based on their granular flow experiments. Their new friction law is able to give more accurate prediction of the run-out distance and deposit morphology of the experimental granular flows on inclined slopes. In their study, it is also suggested that the simulation is not sensitive to the lateral pressure coefficient and good results can be obtained by taking the lateral pressure coefficient to be unity. This conclusion is also supported by molecular dynamics simulations (e.g. Silbert et al. [2003], GDR Midi [2004]). Mangeney et al. [2007] successfully applied the empirical friction law of Pouliquen and Forterre [2002] to simulate the self-channeling phenomenon of unconfined granular flow. One of the major breakthroughs on granular flow mechanics is the introduction of μI rheology (Jop et al. [2006]), which is also frictional, having a formulation similar to the Mohr-Coulomb rheology. The key difference is that the friction coefficient in the μI rheology depends on an inertial parameter characterising the ratio between a macroscopic deformation timescale (shear rate) and an inertial timescale (particle collision). The validity of μI rheology has been confirmed by several experimental and numerical studies (e.g. Lagrée et al. [2011], Staron et al. [2012]). The μI rheology has also been incorporated into the depth-averaged models (Gray and Edwards [2014]) to give more accurate simulations for granular flow experiments. However, the validity of frictional rheology for simulating real-world flow-like landslides is still in debate (Legros [2002]). An apparent shortcoming of all these frictional rheology is that sometimes a friction angle much smaller than the material friction angle has to be used to reproduce satisfactorily real-world events (Legros [2002]). In addition, smaller friction coefficients have to be used in the models to reproduce landslides with larger volume. Such a phenomenon is often referred to as frictional weakening, for which the reasons are still not very clear and cannot be effectively represented by most of the existing rheological laws. Through fitting a large amount of data, Lucas et al. [2014] proposed an empirical frictional resistance formula to take this into account. However, this may still not be adequate for the simulation of real-world landslides because the real-world events could significantly deviate from their fittings.

Voellmy rheology (Voellmy [1955]) combines frictional rheology with a u^2 term to consider turbulence and/or collision between particles. It has been used for simulating avalanches (e.g. Hergarten and Robl [2015]) and debris flows (e.g. Medina et al. [2008], Ouyang et al. [2014]). Kelfoun [2011] studied the suitability of several empirical rheological laws for long-runout avalanches. He found that the plastic rheology is superior in terms of giving more realistic morphological features of the deposits of natural long-runout granular flows. Wang and Sassa [2010] assumed that there is a thin sliding layer at the bottom of moving landslides and proposed a model with a varying apparent friction coefficient. In the model reported in Wang and Sassa [2010], the apparent friction angle increases as the normal pressure decreases, leading to model performance qualitatively similar to the plastic rheology. Although the plastic rheology gives remarkable results for a few real world landslides (Kelfoun and Druitt [2005] and Kelfoun [2011]), a comprehensive analysis has not yet been found and it may only be valid for certain types of landslides.

There are several micro-scale mechanisms contributing to the macro-scale rheology of flow-like landslides material, including inter-particle friction, inter-particle collision, fluid viscosity, fluid buoyancy and fluid turbidity. However, none of the aforementioned rheology is comprehensive in the sense that these micro-scale mechanisms are all considered.

In order to avoid the shortcomings of the single-phase models, in particular, the deficiency of simple rheological laws, a few more sophisticated models assume the material of flow-like landslides including debris flows to be a two-phase mixture. Iverson [1997] pointed out that the pore water pressure plays a major role in those rapidly moving debris flows, especially the formation of multiple surges, which is also supported by comprehensive experimental evidences (Major and Iverson [1999]). The first depth-averaged two-phase model was presented in the companion papers by Iverson and Denlinger [2001] and Denlinger and Iverson [2001]. In this model, the solid phase is assumed to comply with the Mohr-Coulumb friction law while the interaction between the solid and fluid phases is represented through the reduction of the basal friction coefficient and lateral pressure coefficient due to excessive pore pressure. The evolution of excessive pore pressure follows an advection-diffusion equation derived from the consolidation theory. Their model was validated against flume tests. Pitman and Le [2005] presented a full two-phase depth-averaged model where the solid phase and fluid phase are characterised by different variables. This model may be further reduced to a simpler model if the relative movement between solid phase and fluid phase is small and negligible. Pastor et al. [2009] applied the Zienkiewicz-Biot theory (Zienkiewicz et al. [1990]) to model flow-like landslides. To improve the predictive capability, Iverson and George [2014] and George and Iverson [2014] considered the effects of dilatancy on pore water pressure change. To set up and apply these two-phase models, the initial pore water pressure distribution is required which is hard to measure. So it is not trivial to apply these models for real-world applications. Nevertheless, these models provide a more complete physical description of the physical mechanisms than the single-phase models, and so they are more suitable for investigating the influence of different factors on the movement of flow-like landslides. But it should be noted that the aforementioned two-phase models are only applicable when the grain frictional force is dominant over the collisional force, which is not always valid as is suggested by Takahashi [2014].

It is noteworthy that some typical features of natural flow-like landslides such as development of levee, frontal accumulation of coarse grains and formation of multiple surges are still very difficult to be modelled by existing depth-averaged models. These difficulties come from both the complexity of the micro-scale mechanisms and macro-scale heterogeneity of the geological settings. A good summary of these difficulties can be found in Delannay et al. [2017]. As a conclusion, the physics of flow-like landslides is still an active research topic in nowadays and a conclusive understanding has not yet been given. This inevitably hinders the development of more accurate numerical models.

Regarding the formulation of the depth-averaged equations, a major difference between flow-like landslides and water flows is that the flow-like landslides usually take place on steep slopes rather than nearly horizontal and flat ground surface. This brings in major challenges in developing depth-averaged models. As is shown in Figure 2.1, the vertical acceleration ($\mathbf{a}_v$) is non-zero, therefore the pressure distribution along the vertical direction is no longer hydrostatic. In addition, there exists a centrifugal force ($\mathbf{a}_c$) along the direction normal to the bed when the bed is curved. Incorporating the non-hydrostatic pressure and centrifugal force into depth-averaged models is a major challenge.

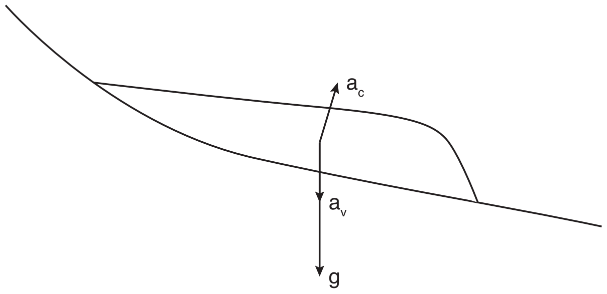


Figure 2.1: Vertical acceleration and centrifugal force for flow-like landsldes running on slopes

In the context of the depth-averaged models, the flow direction is normally assumed to be parallel to the domain surface; the pressure term normal to the domain surface becomes trivial to calculate. Therefore, surface fitted curvilinear coordinate systems have been widely adopted for developing depth-averaged granular flow models. The first of such attempts was made by Gray et al. [1999], using an orthogonal curvilinear coordinate system; but their model assumes predominant topographic variation only occurs in one direction and so presents major difficulty in adapting to real-world terrains with more complex topographic features. Later on, Pudasaini and Hutter [2003] introduced a model on a non-orthogonal curvilinear coordinate system for avalanches in arbitrary curved and twisted channels. Such a non-orthogonal curvilinear coordinate system has also been adopted in other granular flow models (e.g. Pudasaini et al. [2005], Pudasaini et al. [2007]) and achieved certain level of success.

A curvilinear coordinate system relies on the thalweg along the bed to define its main axis. For real-world complex topography, however, it is usually difficult to define the downslope direction or the thalweg due to large variations of topography in different directions. Furthermore, to facilitate the simulation of geophysical flows (e.g. landslides, debris flows and avalanches) in real world, Digital Elevation Models (DEMs) are commonly used to describe the terrain topography, which is defined on a Cartesian coordinate system. Transformation of topographic data from the Cartesian coordinate system to the curvilinear coordinate system must be applied, inevitably increasing computational effort and leading to a loss of accuracy, particularly in the cases where the topography is featured with abrupt changes. To avoid this, Bouchut and Westdickenberg [2004] introduced a shallow water flow model on an arbitrary coordinate system for simulations over topographies with small curvature. Their overall governing equations were derived on a fixed Cartesian coordinate system while the variables were defined in a local reference coordinate system aligning with the local topography, i.e. the flow depth is normal to the bed and the velocities are parallel to the bed. This model was later extended and applied to granular flows by Mangeney et al. [2007] and real world landslide by Kuo et al. [2009]. In their model, the terrain topography may be directly described by a DEM, but coordinate transformation is still needed to provide the initial depth along the vertical axis.

In practice, it is desired to develop a model based on a global Cartesian coordinate system in which the vertical axis is aligned with the gravity direction so that DEMs can be directly used to support model set up without the necessity of coordinate transformation. Using a global Cartesian coordinate system will also facilitate the development of a unified modelling system for both landslides/debris flows and water flows. But a global Cartesian coordinate system based model also has its limitation, as the determination of pressure/stress terms becomes a much more difficult task within such a configuration. Granular flows commonly happen on steep inclined slopes, and the flow acceleration along the gravity direction has a magnitude comparable to the gravity and so is not negligible (Iverson [2014]). The vertical normal pressure is no longer hydrostatic and essentially more difficult to calculate. For the same reason, the centrifugal force caused by bed surface curvature is no long trivial to quantify.

In order to utilise a global Cartesian coordinate system and meanwhile maintain solution accuracy, Denlinger and Iverson [2004] subtracted the vertical acceleration from the gravity acceleration to take into account the non-hydrostatic pressure; their model provided better results than a hydrostatic model for a granular dam-break test. More recently, Castro-Orgaz et al. [2014] suggested that the Boussinesq-type models which retain non-hydrostatic pressure to a certain level and have been successfully applied in shallow water waves modelling may be also implemented for simulating gravity-driven granular flows. They also pointed out that Denlinger and Iverson's model (Denlinger and Iverson [2004]) can actually be categorised as a Boussinesq-type model for granular flows. These Boussinesq-type non-hydrostatic models generally predict better results than their hydrostatic counterparts. However, more sophisticated numerical schemes must be used to solve the Boussinesq-type governing equations due to the presence of additional higher-order derivative terms. Therefore, a Boussinesq-type model is computationally much more demanding than a model solving the SWEs or similar depth-averaged equations.

On the other hand, other simpler global Cartesian coordinate system based models have also been reported (e.g. Juez et al. [2013], Hergarten and Robl [2015]). These models simply modified the original shallow water type equations by including a projection factor to the pressure and source terms, determined by the consideration of bed or surface topography gradients according to heuristic geometric arguments. These models produce very similar results to those models based on local curvilinear coordinate systems, which was confirmed by Xia et al. [2015]. Compared with the Bousinessq-type models, the numerical implementation of these shallow water type models can be much easier to achieve and many well-documented numerical schemes developed for the shallow water equations can be directly used. However, although these models have been proved to be successful for certain applications, they have not been fully justified in a mathematically rigorous way and all of them do not consider the effect of the centrifugal force induced by bed curvature which may become significant for applications involving complex topography.

Except for the depth-averaged models, other approaches, such as discrete element method or direct solution to the three-dimensional governing equations, have also been developed for modelling flow-like landslides (e.g. Zhou and Ng [2010], Utili et al. [2015], Huang et al. [2012]). These methods can give much more details of the flow field and so may be used for studying the detailed mechanics of flow-like landslides. But they are unsuitable for simulation of large-scale real-world problems because they demand much more computational resources.

2.3 Numerical methods for solving the depth-averaged equations

The SWEs and depth-averaged equations for flow-like landslides are both hyperbolic partial differential equations (PDEs) and mathematically similar. Most numerical techniques initially developed for solving the SWEs can be applied to the depth-averaged equations for landslides/debris flows. Therefore this section starts from reviewing the numerical methods for solving the SWEs. Several challenges need to be addressed for developing accurate and robust numerical methods for solving these equations:

- **Automatic shock capturing:** As a non-linear hyperbolic system, shock-like discontinuous solutions may arise even the initial conditions are smooth. So the shock-capturing capability is an essential requirement of a numerical scheme to maintain numerical stability and obtain accurate results. In the meantime, numerical accuracy at the smooth regions should not be sacrificed.

- **Handling wetting and drying:** The SWEs are indeed only valid in wet cells. Numerical methods may face singularities and break down at wet-dry front if no special treatment is made.

- **Correct discretisation of source terms:** Certain source terms associated with the SWEs may alter the mathematical behaviour of the governing equations. For example, the slope source terms may lead to non-trivial solutions for steady state conditions, such as the lake at rest solution. Other source terms like basal friction terms may be linked to strong relaxation towards equilibrium. Therefore, discretisation of these source terms must be done very carefully to reproduce the associated essential physical properties without causing numerical issues.

In the past few decades, research efforts have been dedicated to develop new and improve existing numerical methods to address the above challenges. Numerical methods for solving the SWEs can be categorised as two types: 1) the Eulerian grid based methods, e.g. the finite difference (FD) method, the finite volume (FV) method and the finite element (FE) method; and 2) the Lagrangian meshfree methods such as the Smoothed Particle Hydrodynamics (SPH).

Among all of the three Eulerian methods, FV method has become the most popular for solving the SWEs since 1980s. Because it seeks solutions to the integral form of the conservation laws (e.g. SWEs), FV method exactly conserves mass and momentum and is consistent with the conservation laws in a weak form (i.e. it does not enforce the solution to be continuous) so that shock-capturing schemes can be constructed more intuitively and naturally. FV method is usually incorporated with Godunov-type schemes to evaluate numerical interface fluxes. The original Godunov scheme was firstly proposed for solving the compressible Euler equations for gas dynamics (Godunov [1959]). The basic idea is to compute the numerical fluxes by solving Riemann problems locally at cell interfaces. A Riemann problem is an initial value problem defined by piecewise constant data to give a single discontinuity. Godunov scheme has a few extraordinary advantages, including: 1) the appropriate amount of numerical diffusion is implicitly included so that the shock wave fronts or flow discontinuities can be captured without suffering from numerical oscillations; and 2) the entropy condition is automatically satisfied, i.e. discontinuous solutions (e.g. shock wave fronts) only appear at where it should be. In the original Godunov scheme, the local Riemann problems are solved exactly. However, seeking exact solutions to the Riemann problems is not only computationally expensive but also unnecessary. In the classic work of Harten, Lax and van Leer (Harten et al. [1983]), it was pointed out that not all the information obtained from the Riemann solutions is actually utilised because of the cell averaging process during the flux calculation. This may justify the derivation and use of the approximate Riemann solvers and there have been a few successful attempts for deriving approximate Riemann solvers. Roe [1981] derived one of the most popular approximate Riemann solvers, which essentially involves linearisation of the nonlinear conservation laws and solves exactly the resulting linear equations. Another popular approximate Riemann solver was firstly proposed by Harten et al. [1983], assuming a solution structure of the Riemann problem as a priori. The original work of Harten et al. [1983] assumed a left-moving and right-moving shock wave in the approximate solution but did not indicate how to calculate the wave speed. Later on, Einfeldt et al. [1991] supplemented a choice of wave speeds to give a complete Riemann solver that has been proven to be very effective and robust. This Riemann solver is often known as Harten, Lax and van Leer (HLL) or Harten, Lax, van Leer and Einfeldt (HLLE) Riemann solver. However, because of the absence of contact wave in the solution, HLL/HLLE Riemann solver can be excessively diffusive when dealing with 2D problems or species transport. As an improvement, Toro et al. [1994] proposed the Harten, Lax, van Leer with contact wave restored (HLLC) Riemann solver that included an additional contact wave into the solution structure. The aforementioned approximate Riemann solvers have all been successfully applied to solve the SWEs (Fraccarollo and Toro [1995]). A comparative study (Erduran et al. [2002]) suggested that the HLLC Riemann solver is probably the most suitable choice for solving the SWEs in terms of a combined consideration of ease of implementation, accuracy, applicability, numerical stability and efficiency.

For handling wetting and drying, the simplest method is to assume a thin layer of water covering the dry areas in the beginning of the simulations. This traditional wetting and drying technique has been widely implemented for practical simulations (e.g. Liang et al. [2007]).

However the addition of water mass, even though it is of small amount, is physically incorrect and may significantly degrade the accuracy of the numerical solution near the wet-dry front as pointed out by Toro [2001]. The HLL-type approximate Riemann solvers directly provide solutions to the cases involving wetting and drying, in contrast to the Roe's scheme where modifications are needed to the original solution structure (Brufau et al. [2004]). In the HLLE and HLLC Riemann solvers, distinct wave speeds are attributed to dry cells that are identified by an infinitesimal water depth (10^{-6} m) so that wetting and drying can be computed automatically without causing numerical instability (Fraccarollo and Toro [1995]).

For the discretisation of source terms, different techniques must be introduced and implemented according to the physical and numerical properties that the source terms represent. For those source terms describing processes such as rainfall that happen much slower than the hydrodynamic time scale which represents the timescale that the flow property of a cell can be updated by convection, fractional splitting methods (e.g. Toro [2001]) may be directly applied without causing any numerical issues. However cares must be taken when discretising other source terms, e.g. the bed slope and friction terms.

The time scale associated with the gravitational acceleration due to bed slope is comparable to the hydrodynamic time scale and a numerical scheme solving the SWEs with bed slope terms may lead to a non-trivial steady state for the lake at rest conditions. As suggested by Greenberg and Leroux [1996a] and Bermudez and Vazquez [1994], a viable numerical scheme for solving the SWEs must be able to preserve the lake at rest solution, in order to obtain stable converged numerical solutions because it is not rare that the shallow flow problems are small perturbations to the lake at rest solutions. A scheme with this property is well-balanced (Greenberg and Leroux [1996b]) or satisfying the C-property (Bermudez and Vazquez [1994]). A variety of numerical schemes have been developed to satisfy the C-property. Bermudez and Vazquez [1994] developed an upwind scheme to discretise the slope source term for solving the 1D SWEs. Later on, Bermúdez et al. [1998] extended the method to solve the 2D SWEs on unstructured grids. The major limitation of the upwind discretion of slope source terms is that it depends on the choice of the approaches for numerical flux calculation. Although it has been proven to be accurate for both steady and unsteady flow problems, successful implementation of this upwind discretisation approach is only reported for the Roe's scheme and flux splitting methods. Zhou et al. [2001] proposed the much simpler surface gradient method for slope source term discretisation, which is independent of the formulation for numerical flux calculation. In the surface gradient method, the water depths at the cell interface are reconstructed based on the water surface level so that the water depths at both sides of the cell interface are always equal to each other of the water levels are equal to each other; this obviously will not introduce spurious fluxes to the lake at rest problem and therefore preserve the C-property. Audusse et al. [2004] introduced a hydrostatic reconstruction method by adopting a similar idea to the surface gradient method, i.e. to properly modify the Riemann variables before solving the Riemann problem. Compared with the surface gradient method, the formulation of the hydrostatic reconstruction is even simpler and does not require linear interpolation of water surface level. Another important advantage of the hydrostatic reconstruction method is that it automatically ensures the positivity of water depth and therefore facilitates the derivation of a positivity-preserving technique for handling wetting and drying. The hydrostatic reconstruction scheme has also been applied to general unstructured grids (Audusse and Bristeau [2005]) and implemented and applied successfully for various real-world applications (e.g. Hou et al. [2013a], Wang et al. [2011], Marche et al. [2007], Mangeney et al. [2007], Liang et al. [2015b]).

Mathematical balancing approaches have also been widely reported to construct well-balanced schemes (e.g. Rogers et al. [2003], Liang and Marche [2009], Duran et al. [2013]). The basic idea is to reformulate the SWEs through algebraic manipulation so that the C-property of the numerical schemes can be preserved using trivial source term discretisation (e.g. a central difference scheme). It is also possible to achieve well-balanced numerical schemes through reformulating the slope source terms into the Riemann problems and subsequently seeking modified Riemann solutions (LeFloch and Thanh [2007], LeFloch and Thanh [2011]), George [2008]). The benefit is that these schemes can usually preserve even the non-motionless steady states.

In order to stabilise numerical simulations, the aforementioned well-balanced schemes usually constrain the calculated slope gradient (e.g. Audusse et al. [2004], Liang and Marche [2009], George [2008], Murillo and García-Navarro [2010]) either in the whole domain or near wetting-drying front. The constraint of slope gradient effectively avoids unrealistic large velocities in places where topography is discontinuous and the slope gradient is large such as building walls and river banks, thus enabling much more stable and efficient simulations. For overland flow simulations, most of the domain is wet regardless of the local topography because of rainfall. This means that water can flow on those discontinuous places. Therefore, constraining slope gradient in the whole domain seems to be inevitable otherwise simulations may become unstable. But this comes with a price. A recent study (Delestre et al. [2012]) suggests that constraining slope gradient may leads to inaccurate calculation of the slope gradient if the water depth is vanishingly small. This limitation could lead to inaccurate results when simulating rainfall-induced overland flows and therefore must be circumvented.

There are also other source terms that represent physical processes with much shorter timescales than the hydrodynamic process. A typical example is the friction terms. The friction source terms are usually expressed as a non-linear functions of the flow velocity and depth, e.g. the Manning formula. Its nonlinear nature tends to relax the flow velocity towards an equilibrium state that features the balance between the friction and gravity driving force caused by the existence of a bed slope. If the water depth is small, the friction source terms can be stiff, i.e. their relaxation timescale is much smaller than the hydrodynamic timescale determined by the CFL condition. In such a situation, implicit numerical schemes have to be used to maintain numerical stability. Yet the numerical stability is not the only concern and the correct calculation of the equilibrium is also important because it features the essential physical processes of overland flows. The commonly used kinematic wave theory for overland flow is indeed based on the assumption that the equilibrium is reached everywhere in the domain. In other words, the SWEs reduce to the kinematic wave equation in the limit of disappearing water depth. Therefore, when constructing a numerical scheme to solve the SWEs for overland flows or applications involving wetting and drying over steep slopes, a basic requirement is that that the numerical scheme should relax the flow towards an equilibrium state that is consistent with the kinematic wave theory in a single time step if the relaxation timescale is much smaller than the hydrodynamic timescale (Jin [2012], Teyssier [2014]). The choice of the Riemann solvers also requires further consideration. Depending on the roughness size and water depth, the kinematic wave equation may give a signal speed that is smaller than the shallow water wave speed. But the Riemann solvers currently being used generally do not consider the nonlinearity of friction source terms and the reduction of the signal speed. Because of this, certain Riemann solvers may significantly overestimate the flow discharge or velocity as we will discuss later in Chapter 4, which must be further investigated and avoided when constructing a numerical scheme. Several attempts have been reported in the literature in the field of mathematics to provide a preliminary theoretical background for solving the hyperbolic systems with stiff source terms (e.g. Liu [1987], Chen et al. [1994], Jin [2012]). But further research effort is still needed to further explore these approaches for engineering applications, particularly the simulation of overland flows over complex terrains with wetting and drying, which will be considered in this thesis.

An alternative to the grid-based Eulerian methods is the meshfree Lagrangian approaches. The most widely used meshfree method is the smoothed particle hydrodynamics (SPH). SPH was originally developed for astrophysical hydrodynamic simulation by Lucy [1977] and Gingold and Monaghan [1977]. In SPH, the flow information such as density and velocities are carried by a set of particles which may be interpreted as the Monte-Carlo sampling of the flow field. A particle is advected by the corresponding flow velocity which is updated by interacting with the surrounding particles of the particle under consideration within a specified radius (smoothing length). The contributions of all the surrounding particles are averagely weighted using a smoothing kernel function. In this way, there is no need of defining a mesh that connects the computational nodes that are arranged in order. Therefore, SPH does not suffer from the problem of grid distorting that a grid-based lagrangian method usually faces. Since it was introduced, SPH has been applied to solve a variety of complex fluid problems including astrophysical flows (e.g. Gingold and Monaghan [1977], Springel [2001]), free surface flows (e.g. Monaghan [1994]) and multi-phase flows (Monaghan and Kocharyan [1995]).

When solving the depth-averaged equations for water-related natural hazards, another advantage of using SPH is that the spatial variance of the material properties such as density and water content can be easily represented in the model because the material properties are carried by the particles during simulations. In applying SPH to solve the SWEs, Wang and Shen [1999] firstly applied a standard SPH formulation to solve the 1D SWEs and the numerical predictions were reported to match very well with the analytical solutions, even in the presence of shocks. Ata and Soulaïmani [2005] proposed a stabilised SPH method to solve the SWEs neglecting the bed slope and friction terms; their approach was based on Lax-Friedrich numerical fluxes and boasted of good shock-capturing capability. Rodriguez-Paz and Bonet [2005] introduced a corrected SPH-SWEs scheme, in which the internal and external forces were expressed in the form of energy terms and the vertical velocity component was also taken into account. De Leffe et al. [2010] proposed an algorithm to periodically redistribute the particles in order to improve solution accuracy. However, most of these models are designed and tested for solving the SWEs that neglect the source terms and therefore are not applicable to the practical problems where the domain topography is generally very complex.

In SPH, the challenge of correctly discretising the source terms, especially the slope source terms, has just started to gain attention from the CFD researchers. An initial attempt of developing a well-balanced SPH-SWEs model was reported by Vacondio et al. [2012a], which explicitly included the slope source terms into their SPH discretisation and presented a new model that was able to reproduce the still water surface to a certain level. Their results indicated that the lake at rest solution in a domain with topography could not be preserved without special numerical treatments. To better resolve this problem, Vacondio et al. [2013] reported a corrected method for balancing the source terms. However, their model is well-balanced only under the assumption that the particles are equally distributed in space and have the same smooth length. Furthermore, their model has not been validated by a more challenging real-world problem with complex topography. There is a research gap to be filled here to enable practical shallow flow simulations using SPH.

Most of the numerical techniques developed for solving the SWEs can also be applied to solve the depth-averaged equations for landslides and debris flows. Denlinger and Iverson [2001] reported a FV model implemented with an HLLC Riemann solver for two-phase debris flows. Later, the authors presented another model implemented with the Roe's approximate Riemann solver for granular flows (Denlinger and Iverson [2004]). George and Iverson [2014] proposed a novel Riemann solver for the multi-phase debris flow equations taking into account dilation. Mangeney-Castelnau et al. [2005] applied the hydrostatic reconstruction method and treated the basal friction as an apparent slope to obtain a well-balanced scheme. Meshfree methods have also been used for simulation landslides and debris flows. For example, Mcdougall and Hungr [2004] applied SPH to simulate landslides. Pastor et al. [2009] used SPH to solve the two-phase governing equations for flow-like landslides.

Compared with water flows, the underlying physics associated with the flow-like landslides is still underdeveloped and requires further research effort. So most of the modelling studies to date for flow-like landslides and debris flows have been focused on improving the physical descriptions in the mathematical formulation rather than the numerical methods. Many existing models are still based on simple numerical methods without comprehensively considering shock-capturing, wetting and drying and correct discretisation of source terms (e.g. Wang and Sassa [2010], Mcdougall and Hungr [2004], Hungr [1995], Kelfoun and Druitt [2005], Sassa et al. [2010]). However, numerical artefacts such as spurious oscillations may introduce noises to the solutions and make it difficult to interpret the results and verify the underlying physical assumptions. Therefore, whilst it is important to improve the understanding of physical processes themselves, it is also essential to develop well-posed numerical schemes to solve the governing equations and support efficient and accurate simulation of flow-like landslides.

2.4 High-performance computing in the context of water-related natural hazards simulation

High-performance computing (HPC) has been widely developed and applied for water-related natural hazards modelling in the recent years. HPC models can be divided into two types, i.e. CPU-based and GPU-based, according to the hardware being used. CPU-based models utilise Open Multi-Processing (OpenMP) for share memory parallelisation and Message Passing Interface (MPI) for distributed memory parallelisation. TELEMAC-2D (Telemac Software [2016]) and Delft-3D (Deltares Open Software [2016]) are the examples of CPU-based HPC models for flood simulations. FLO-2D (FLO-2D Software INC [2016]) is also a CPU-based parallelised flood modelling software that can be also used to simulate debris flows. But the CPU-based HPC models pose specific requirements on hardware and need access to multi-node clusters or super computers for large-scale simulations.

In the last 5–10 years, GPU-based high-performance computing has been rapidly developed to become a more promising approach for scientific computing. A single GPU can provide computational power that is equivalent to almost 10 multi-core CPUs (NVIDIA [2015]). Therefore the computational power of a super computer can be packed into a small workstation, which makes HPC much more accessible to scientists and engineers. In recent years, GPU-based models are becoming increasingly popular for modelling water-related natural hazards, e.g. floods, landslides and debris flows. Programming platforms to develop computing codes on GPUs include OpenCL, CUDA or OpenACC. OpenCL is a cross-platform and open-source standard maintained by the Khronos group. The major benefit of using OpenCL is that it allows the resulting codes to run on both CPUs and GPUs. An attempt to implement a finite volume SWEs solver using OpenCL was made by Smith and Liang [2013]. An alternative GPU programming framework is Compute Unified Device Architecture (CUDA) developed by NVIDIA. Compared with OpenCL, CUDA only supports GPUs manufactured by NVIDIA but is much more user friendly and supports better the object oriented programming. The performance of CUDA and OpenCL is on par with each other. Due to its ease to implement, CUDA have been more widely used. Several CUDA based models for simulating rainfall-runoff and flood inundation have been developed and reported in the literature. Brodtkorb et al. [2012]

implemented a second-order finite volume SWEs solver on GPU together with an OpenGL component for visualisation on the same GPU. JFlow (JBA Consulting [2016]) is a commercial code solving the diffusion-wave equation or the full SWEs for flood simulation, which has been recently further developed to support simulations on GPUs. Zhai et al. [2015] introduced a model based on a Godunov-type finite volume scheme to solve the Savage-Hutter equations for simulating granular flows on GPUs. For simulations on a single GPU, a speedup of 30-40 times against a single CPU core can usually be expected for solving the depth-averaged equations using grid-based methods.

GPUs have also been used to accelerate SPH models. Hérault et al. [2010] implemented an SPH model on GPUs using CUDA and compared the model performance against its FORTRAN counterpart running on a CPU core. Domínguez et al. [2013a] reviewed the prevailing optimisation strategies for implementation of SPH models on GPU; later, they also improved their SPH model to run on multiple GPUs (Domínguez et al. [2013b]). The performance of these GPU-accelerated SPH models has been substantially improved against their CPU counterparts in terms of computational efficiency. In the works of Lee and Han [2010] and Solenthaler et al. [2011], GPU-accelerated SPH models were used to solve the SWEs for real-time visualisation of water surface movement. Xia and Liang [2016] applied GPUs for solving the SWEs with variable smoothing length and obtained 5-10 times of speedup against a quad-core CPU.

Regarding the software implementation, a code base that has a flexible structure to allow easy extension of functionality is essential for developing a physically based multi-hazard model which will involve long-term research effort. GeoClaw (Berger et al. [2011]) presents an example for such a flexible software framework although it only allows simulations on CPUs. GeoClaw is an adaptive mesh refinement (AMR) based computer model for solving the hyperbolic conservation laws (i.e. SWEs), which allows users to develop new functionality through adding new Riemann solver subroutines. But a GPUs-based flexible software framework for multi-hazard modelling is still currently unavailable.

2.5 Conclusions

Based on the literature review in the previous sections, the following research gaps have been identified, corresponding to the aim and objectives of this thesis introduced in the previous chapter.

1. Most of the existing multi-hazard models seem to be over simplified or not physically based. There is an urgent need to better understand the underlying physics of flow-like landslides or debris flows and their interaction with other hazards and catchment processes, and to subsequently develop the next generation of physically based modelling tools to support multi-hazard simulations and hazard risk management. Because both flooding and flow-like landslides or debris flows may be mathematically described by the depth-averaged equations, a natural idea is that multiple hazards and their interaction can be simulated using an integrated depth-averaged formulation solved in a single codebase. However, such a modelling system has not been developed and reported in the literature.

2. A well-posed numerical scheme must be identified and further developed to solve the integrated governing equations to support accurate and stable simulations of multiple water-related natural hazards and the related processes.

3. Rainfall-runoff and overland flows play a central role in the formation of surface flooding and initiation of landslides and debris flows. Most of the existing numerical schemes developed for overland flows suffer from inaccurate discretisation of the bed slope and friction source terms in the limit of disappearing water depth, leading to unreliable simulations. New numerical methods must be developed for accurate and efficient simulation of overland flows in the context of the proposed integrated multi-hazard modelling system.

4. Most depth-averaged models for flow-like landslides are built upon surface fitted curvilinear coordinates in order to consider non-hydrostatic pressure and curvature effects more accurately. There is a need to develop a unified mathematical framework on Cartesian coordinate system to better support real-world applications.

5. A flexible and GPU-accelerated software framework does not currently exist and must be developed to support long-term development of a robust modelling system for simulating water-related natural hazard chains and facilitating hazard risk reduction and management.

This thesis will deliver research to fill these knowledge gaps and achieve the aim and objectives as identified in the previous chapter.

Chapter 3. A well-balanced SPH model for solving the shallow water equations

3.1 Introduction

As an attractive alternative to the Eulerian finite volume method, the performance of SPH on solving depth-averaged equations is worth a further investigation to determine whether it is the more suitable numerical method for developing the proposed catchment-scale physically-based multi-hazard model. In this chapter, SPH is applied to solve the SWEs and compared with a Godunov-type finite volume method. As already mentioned in Chapter 2, correct balancing of slope source terms and internal pressure is an issue that must be resolved. For this purpose, the well-balanced problem is firstly analysed and discussed after briefly reviewing the stand SPH formulation. The error introduced by the standard SPH approximation to the water depth over a varying bed profile is then derived and subsequently corrected to obtain a new SPH formulation for both the continuity and the momentum equations, which enables the source term balancing. To enhance the shock-capturing capability, the Riemann solver based artificial viscosity proposed by Monaghan [1997] is applied. The new corrected SPH-SWEs model is then validated by applying it to simulate several 1D and 2D shallow flow benchmark tests. Then the GPUs are used to accelerate the SPH-SWEs model. Finally, the SPH-SWEs model is compared with the Godunov-type finite volume model developed by Liang [2010] in the context of practical flood simulations.

3.2 Brief review of the SPH method

The essence of SPH method is that any generic function can be represented as

$$f(\mathbf{x}) = \int_{\Omega} f(\mathbf{x}')\delta(\mathbf{x} - \mathbf{x}')d\mathbf{x}' \tag{3.1}$$

where Ω is the integral domain and $\delta(\mathbf{x}-\mathbf{x}')$ is the Dirac delta function defined as

$$\delta(\mathbf{x}-\mathbf{x}') = \begin{cases} \infty, & \mathbf{x} = \mathbf{x}' \\ 0, & \mathbf{x} \neq \mathbf{x}' \end{cases} \tag{3.2}$$

In practice, $\delta(\mathbf{x}-\mathbf{x}')$ is conventionally replaced by a smoothing kernel function $W(\mathbf{x}-\mathbf{x}', \hbar)$ with $\hbar$ being the smoothing length that determines the influence domain of the kernel. Hence (3.1) becomes

$$f(\mathbf{x}) \approx \int_{\Omega} f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}', \hbar)d\mathbf{x}' \tag{3.3}$$

which is called the kernel approximation. In the context of SPH, kernel approximation operator is usually marked with <> and so (3.3) can be rewritten as

$$< f(\mathbf{x}) >= \int_{\Omega} f(\mathbf{x}')W(\mathbf{x}-\mathbf{x}', \hbar)d\mathbf{x}' \tag{3.4}$$

The kernel approximation of the gradient of $f(\mathbf{x})$ can be obtained by

$$< \nabla f(\mathbf{x}) >= \int_{\Omega} f(\mathbf{x}')\nabla W(\mathbf{x}-\mathbf{x}', \hbar)d\mathbf{x}' \tag{3.5}$$

There are several mathematical conditions that the kernel function must satisfy, which are

1. $\lim_{\hbar \to 0} W(\mathbf{x}-\mathbf{x}', \hbar) = \delta(\mathbf{x}-\mathbf{x}')$
2. $\int_{\Omega} W(\mathbf{x}-\mathbf{x}', \hbar)d\mathbf{x}' = 1$, which ensures the zero-order consistency.
3. $W(\mathbf{x}-\mathbf{x}', \hbar) = 0$, if $|\mathbf{x}-\mathbf{x}'| > \kappa h$, where κ is an integer (usually < 4)
4. $W(\mathbf{x}'-\mathbf{x}, \hbar) = W(\mathbf{x}-\mathbf{x}', \hbar)$

In SPH calculations, it is practical to subdivide the integral domain Ω into a finite set of N particles. Each particle carries a mass of m_j and has a density of ρ_j, with $j = 1, 2, 3, \ldots, N$. The sum of the mass for every particle gives the total mass of the fluid body under consideration. The SPH approximation of $f(\mathbf{x})$ and $\nabla f(\mathbf{x})$ at $\mathbf{x}_i$ can then be calculated by replacing the integral with the summation of respective

$$f_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j W_{ij} \tag{3.6}$$

$$\nabla f_i = \sum_{j=1}^{N} \frac{m_j}{\rho_j} f_j \nabla W_{ij} \tag{3.7}$$

where $W_{ij} = W(\mathbf{x}_i - \mathbf{x}_j, \hbar)$. Since $W(\mathbf{x} - \mathbf{x}', \hbar) = 0$ when $|\mathbf{x} - \mathbf{x}'| > \kappa\hbar$, the total number of the contributing particles is actually much less than N.

In this work, a Gaussian kernel (Liu and Liu [2003]) is adopted, which leads to a second-order accurate approximation. Detailed discussion of other kernel functions may be found in (Liu and Liu [2003]).

3.3 Standard formulation of SPH-SWEs model

Neglecting the Coriolis effects and fluid viscosity, the shallow water equations can be written in the Lagrangian form as

$$\frac{Dh}{Dt} = -h\nabla\mathbf{u} \tag{3.8}$$

$$\frac{D\mathbf{u}}{Dt} = -g(\nabla h + \nabla b) - g\frac{n^2\mathbf{u}||\mathbf{u}||}{h^{4/3}} \tag{3.9}$$

where D/Dt represents the total derivative, $\mathbf{u}$ denotes the depth averaged velocity vector, g is the acceleration due to gravity, h is the water depth, b is the bed elevation, and n is the Manning coefficient.

When implementing an SPH algorithm, the fluid body under consideration is discretised into a series of particles. With a 2D SPH-SWEs model, the particles represent water columns of height h_j and volume V_j. Obviously, the corresponding volume of each water column or particle is constant by assuming water as an incompressible fluid but the height can vary. So the volume V_j and the water depth h_j of an arbitrary particle j in the 2D SPH-SWEs are, respectively, analogous to the mass m_j and density ρ_j in a traditional 3D SPH model as expressed in (3.6-3.7). Consequently, we may replace m_j and ρ_j in (3.6) by V_j and h_j, respectively, to obtain an SPH approximation of water depth h_i at $\mathbf{x}_i$, i.e.

$$h_i = \sum_{j=1}^{N} V_j W_{ij} \tag{3.10}$$

This is essentially similar to the continuity equation in the traditional SPH method. If the bed friction term is ignored, from (3.9), the momentum component of the SWEs for particle reads

$$\frac{D\mathbf{u}_i}{Dt} = -g\nabla h_i - g\nabla b_i \tag{3.11}$$

The SPH formulation of the momentum conservation equation (3.11) is then derived by applying the SPH approximation to its right hand side, i.e.

$$\frac{D\mathbf{u}_i}{Dt} = -g\sum_{j=1}^{N} V_j \nabla W_{ij} - g\sum_{j=1}^{N} \frac{V_j}{h_j} b_j \nabla W_{ij} \tag{3.12}$$

(3.10) and (3.12) forms the standard SPH formulation for the SWEs, which conserves both mass and momentum (e.g. Wang and Shen [1999]). The first part of right hand side of (3.12) is the internal force, the second part gives the source term. However, as it will be discussed in the following section, this standard form does not maintain the solution of lake at rest and hence is not well-balanced.

3.4 Well-balanced SPH-SWEs model

3.4.1 *Well-balanced issue in SPH*

The lake at rest problem is defined as

$$h + b = const = C \text{ and } \mathbf{u} = 0 \tag{3.13}$$

where C is a constant. Without external driving forces, the water body should stay tranquil against time. A numerical scheme that is able to preserve the lake at rest solutions is called 'well-balanced'.

In the grid based Eulerian methods(e.g. finite volume and finite difference), spurious perturbation to the lake at rest is a result of discretising the momentum equations. However, in SPH, the discretised continuity equation may also drive the fluid into motion due to the numerical error introduced by particle approximation. In the standard form of the SPH-SWEs model, the water depth is evaluated by (3.10). Since it is a weighted average from the surrounding particles, the approximated water depth h_i may not be same as the 'actual value' at the same location. This may mistakenly modify the water depth and hence disturb the still water surface. Therefore, unlike the Eulerian approaches (e.g. finite volume and finite difference) where the well-balanced problem is only caused by discretising the momentum equation, the still water body can be also driven into motion by the discretised continuity equation due to the error introduced by particle approximation in an SPH model. In return, the perturbed water surface will violate the relationship defined in (3.13) and consequently generate unphysical velocity when updating (3.12). This may be demonstrated by a simple example as shown in Figure 3.1, where a 1D still water body is assumed on horizontal bed with a step. Numerical calculation is performed for one time step using the standard SPH-SWEs formulation given by (3.10) and (3.12). As illustrated in Figure 3.1(a), the water depth is 'smoothed' near the step whereas the change of the original depth is abrupt. Consequently, the peaceful water surface is perturbed as indicated in Figure 3.1(b), which will obviously move the water body when updating the fluid velocity using (3.12).

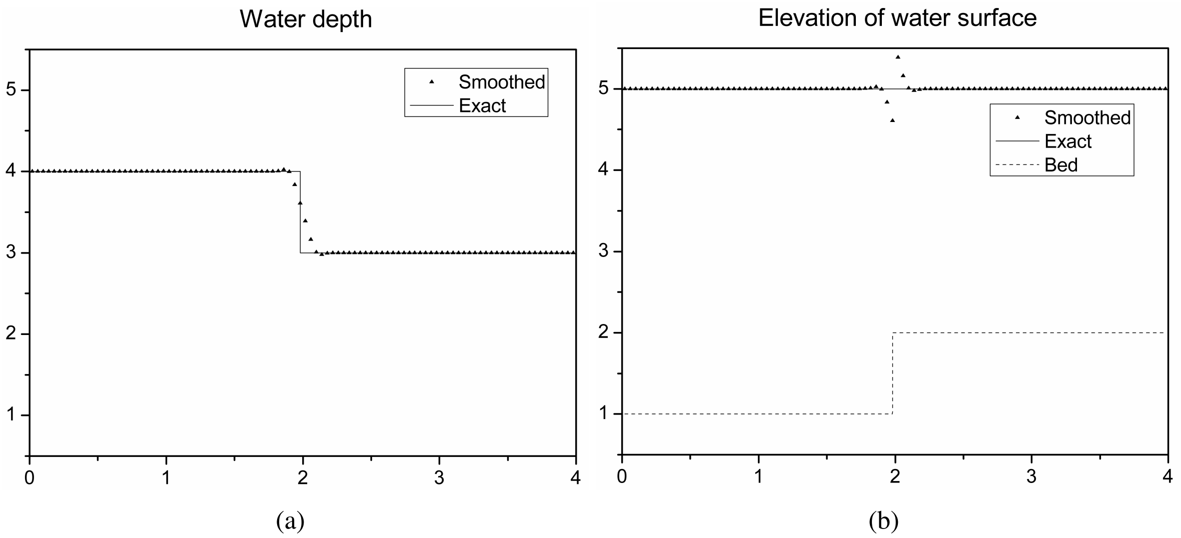


Figure 3.1: Well-balanced problem caused by the standard SPH discretization of SWEs: (a) water depth; (b) surface elevation.

In SPH, the discretised momentum equation may also cause a well-balanced problem. Firstly, we consider the simplest condition where $b = 0$ and $h = C$. By eliminating b, (3.12) may be rewritten as

$$\frac{D\mathbf{u}_i}{Dt} = -gC\sum_{j=1}^{N}\frac{V_j}{C}\nabla W_{ij} = -gh_i\sum_{j=1}^{N}\frac{V_j}{h_j}\nabla W_{ij} \tag{3.14}$$

In SPH, $\sum_{j=1}^{N}\frac{V_j}{h_j}\nabla W_{ij}$ is approximately equal to zero and the discretisation error is determined by the truncation error of particle approximation. In most cases, this error is small. However, in the case when the particles are disorderly distributed, this error may not be negligible (Quinlan et al. [2006]). Obviously, under this circumstance, although initially the water depths of all particles are equal and their velocities are zero, the particles are not necessarily motionless after the SPH approximation. This motion is not caused by the unbalance of source terms and internal forces but the result of the truncation error of particle approximation. However, this truncation error is essential in maintaining the momentum conservation of the numerical scheme. In this work, this truncation error is maintained in order to ensure the momentum to be exactly conserved.

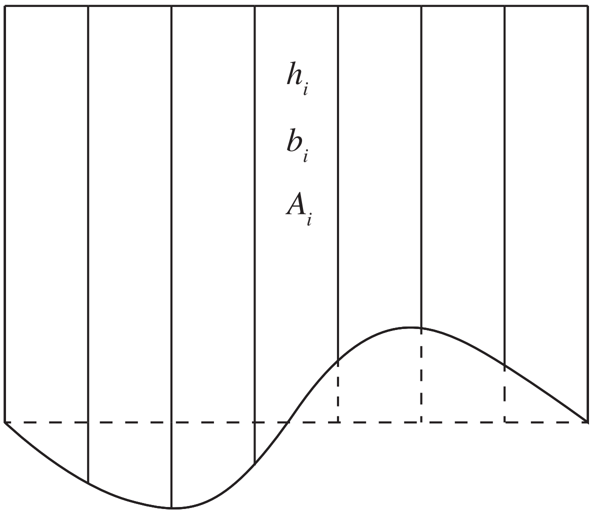


Figure 3.2: Balancing the source terms and internal forces in SPH.

For the lake at rest problem with $b \neq 0$, the momentum equation (3.12) can be rewritten by replacing b with $C - h$ as

$$\frac{D\mathbf{u}_i}{Dt} = -g \sum_{j=1}^{N} V_j \nabla W_{ij} - g \sum_{j=1}^{N} \frac{V_j}{h_j}(C - h_j)\nabla W_{ij} = -gC \sum_{j=1}^{N} \frac{V_j}{h_j} \nabla W_{ij} \tag{3.15}$$

(3.15) is also only approximately equal to zero. As discussed before, the water particles are not necessarily motionless after SPH calculation even when the particles have constant water surface elevation and zero velocities. This obviously violates the well-balanced assumption.

When the conditions described by (3.13) are satisfied, the acceleration that each particle holds is defined as the residual acceleration $\mathbf{a}_r$. Next, a definition is given to the balance of source terms and internal forces in SPH. Two conditions are considered. Condition I involves particle i and its neighbouring particles having equal water level but different bed elevation. The water depth h, bed elevation b and projection area A are indicated by the solid lines in Figure 3.2. In condition II, particle i and its neighbours have equal water depth (all equal to h_i), zero bed elevation and projection areas that are equal to those in condition I, as indicated by the dashed line in Figure 3.2.

Definition: If the residual acceleration for an arbitrary particle in conditions I and II are equal, the source terms and internal forces in the discretised momentum equation are defined to be balanced.

This definition can be regarded as a weaker form of the well-balancedness. In the standard SPH formulation for the SWEs, it is straightforward to obtain the residual accelerations for particle i in condition I and II:

Condition I:

$$\mathbf{a}_{r,i} = -g(h_i + b_i)\sum_{j=1}^{N} A_j \nabla W_{ij} \tag{3.16}$$

Condition II:

$$\mathbf{a}_{r,i} = -gh_i\sum_{j=1}^{N} A_j \nabla W_{ij} \tag{3.17}$$

Obviously, the residual accelerations in conditions I and II are not equal, and thus the source terms and internal forces in momentum equation from the standard form of SPH for SWEs are not balanced according to the aforementioned definition.

Most of the existing SPH models reported in literature for solving the SWEs employ the standard formulation, i.e. adopting (3.10) to calculate the water depth (e.g. Wang and Shen [1999], Ata and Soulaïmani [2005], Rodriguez-Paz and Bonet [2005]). Therefore the well-balanced problem is an issue that must be resolved before applying these models in practical simulations.

3.4.2 *SPH discretisation of water depth*

In order to devise a well-balanced scheme for calculating water depth using SPH, it is necessary to firstly analyse the error introduced by the SPH discretisation of (3.10). Defining the 'true' water depth of particle i at the current time to be $\tilde{h}_i$, the error caused by the kernel approximation can be obtained by finding the difference between the approximated and the 'true' depths

$$E_h = \sum_{j=1}^{N} V_j W_{ij} - \tilde{h}_i \tag{3.18}$$

which may be re-formulated as

$$E_h = \sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j} \tilde{h}_i W_{ij} - \tilde{h}_i + \sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j} (\tilde{h}_j - \tilde{h}_i) W_{ij} \tag{3.19}$$

where $\sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j} \tilde{h}_i W_{ij} - \tilde{h}_i$ reflects the accuracy of particle approximation and diminishes when enough number of particles are involved in the discretization. $\sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j} (\tilde{h}_j - \tilde{h}_i) W_{ij}$ is the discretisation error induced by the initial variation of the water depth, which is termed *depth variation error* in this work and contributes to the well-balanced problem as discussed previously.

Defining η as the water surface elevation (water level), the relationship among the water surface elevation, water depth and bed elevation is $\eta_i = h_i + b_i$ for an arbitrary particle i. The depth variation error can thus be rewritten as

$$\sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j} (\tilde{h}_j - \tilde{h}_i) W_{ij} = \sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j} (\eta_j - \eta_i) W_{ij} + \sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j} (b_j - b_i) W_{ij} \tag{3.20}$$

The depth variation error can be further categorized into two parts, which are respectively $\sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j} (\eta_j - \eta_i) W_{ij}$ (caused by the variation of water surface elevation and defined as *surface variation error*) and $\sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j} (b_j - b_i) W_{ij}$ (caused by varying bed elevation and therefore called *bed variation error*). Evidently, for a lake at rest problem, the surface variation error disappears and it is the bed variation error that introduces the discretisation error to (3.10) and eventually dissatisfies the well-balanced condition. Therefore, a corrected SPH equation for calculating water depth in the SWEs may be introduced as follows

$$h_i = \sum_{j=1}^{N} V_j W_{ij} - \sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j} (b_j - b_i) W_{ij} \tag{3.21}$$

The 'true water depth' is unknown. But it is logical to assume that the 'true water depth' is the same as the corrected water depth. So (3.21) becomes

$$h_i = \sum_{j=1}^{N} V_j W_{ij} - \sum_{j=1}^{N} \frac{V_j}{h_j} (b_j - b_i) W_{ij} \tag{3.22}$$

The variable 'h' appears on both sides of (3.22) and hence the equation cannot be solved explicitly. Vacondio et al. [2013] introduced a similar equation but replaced h_j by the values obtained at the previous time step to allow explicit calculation. However, this is not mathematically rigorous and numerical experiments indicate that such a treatment may lead to inaccurate numerical solutions in certain test cases. Therefore, (3.22) is solved iteratively in this work and a simple iteration procedure is given by

$$h_i^{k+1} = \sum_{j=1}^{N} V_j W_{ij} - \sum_{j=1}^{N} \frac{V_j}{h_j^k}(b_j - b_i)W_{ij} \tag{3.23}$$

where k refers to iteration step number (but not the time level). At the first iteration, the water depth without correction is used as the initial value to start the iterative procedure, i.e.

$$h_i^0 = \sum_{j=1}^{N} V_j W_{ij} \tag{3.24}$$

Herein, we will demonstrate that the above corrected continuity equation preserves the lake at rest solution. Since $E_h = \sum_{j=1}^{N} V_j W_{ij} - \tilde{h}_i$, (3.21) can be rearranged to give

$$\begin{aligned} h_i &= \tilde{h}_i + E_h - \sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j}(b_i - b_j)W_{ij} \\ &= \tilde{h}_i + \sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j}\tilde{h}_i W_{ij} - \tilde{h}_i + \sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j}(\eta_j - \eta_i)W_{ij} \\ &= \sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j}\tilde{h}_i W_{ij} + \sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j}(\eta_j - \eta_i)W_{ij} \end{aligned} \tag{3.25}$$

As discussed in section 3.2, the integral of the kernel function over the domain is 1 and $\sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j} W_{ij}$ is approximately equal to 1 when enough particles are involved. In the condition of lake at rest, $\sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j}(\eta_j - \eta_i)W_{ij} = 0$. Therefore, (3.21) calculates the actual water depth after neglecting the inherent truncation error of particle approximation introduced by $\sum_{j=1}^{N} \frac{V_j}{\tilde{h}_j} W_{ij}$, which is theoretically small and negligible if the particles are placed orderly at the beginning. Consequently, the relationship defined in (3.13) is satisfied and thus the new formula (3.21), and subsequently (3.22), preserves the solution of lake at rest. It is trivial to prove that (3.21) is the same as (3.10) when bed is horizontal, e.g. the modified continuity equation is identical with the standard SPH when bed is horizontal.

No specific assumption is made to the kernel function in the above analysis and therefore any kernel function can be adopted to combine with the current scheme provided that it satisfies the mathematical conditions given in section 3.2. The above modification is also valid for an SPH formulation with either constant or variable smooth length.

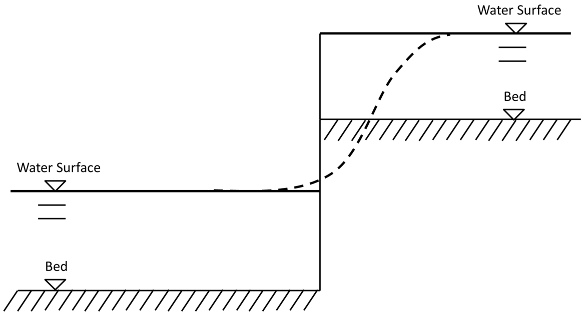
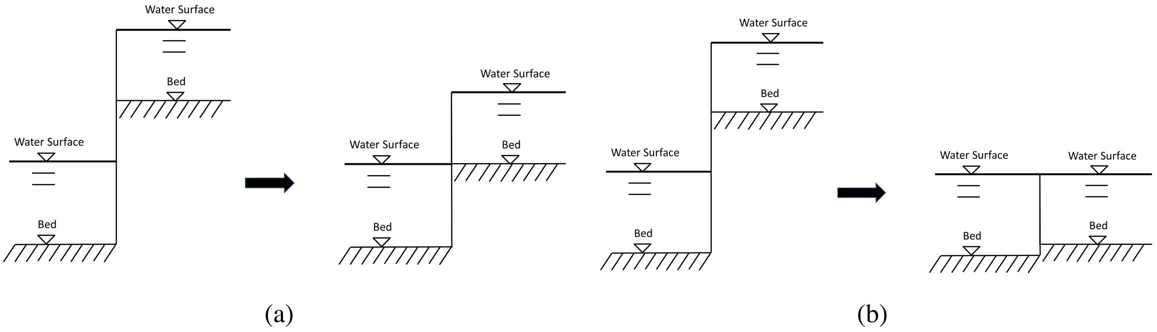


Figure 3.3: Cause of negative water depth in SPH.

However, the modified formula of water depth, i.e. (3.22), may result in negative water depth in certain numerical simulations. To demonstrate this, (3.22) is rewritten as:

$$h_i = \sum_{j=1}^{N} \frac{V_j}{h_j}(h_j + b_j - b_i)W_{ij} \tag{3.26}$$

where h_j may be less than $b_i - b_j$ and so h_i may become negative. This happens when a particle has a bed level that is above its neighbour's water surface, as illustrated in Figure 3.3. At each iteration step, the initial water depth is corrected. The corresponding corrected water surface (dashed line in Figure 3.3) is different from the initial water surface (solid line in Figure 3.3). Obviously, if the corrected water surface is below the bed level, the water depth will consequently become negative, which is physically incorrect and must be avoided.



(a)

(b)

Figure 3.4: Modification of bed profile in SPH.

Herein, a local bed modification method is proposed in order to avoid the generation of negative water depth. At the beginning of every iteration step, the bed elevation is temporally modified for the two neighbouring particles as shown in Figure 3.3 by removing the gap between the higher bed level and the lower water surface, which is illustrated in Figure 3.4(a). However, numerical experiments show that the calculation may still become unstable when the gradient of bed profile is very large. Therefore, a further reduction to the higher bed elevation is implemented until the water surface elevation of two particles becomes equal, as presented in Figure 3.4(b). The maximum reduction is constrained to maintain the monotone relationship of bed elevation between the two particles. For example, if the bed elevation of one particle is originally larger than that of the other one, it cannot become smaller after modification. As a whole, the local bed modification scheme is implemented for the two neighbouring particles i and j as:

$$b_i = \begin{cases} \max(b_j, b_j + h_j - h_i) & , b_i > b_j + h_j \\ b_i & , b_i \leq b_j + h_j \end{cases} \tag{3.27}$$

It can be concluded by revisiting (3.26) that negative water depth will be no longer calculated after the local bed modification. It is worth pointing out that the local bed modification is only necessary when updating water depth. The bed elevation remains to be the original value when velocity is updated.

Finally, in order to reduce further the spurious oscillation of water depth, a further correction is implemented, as suggested by Vacondio et al. [2012b],

$$h_i = \frac{h_i}{\sum_{j=1}^{N} \frac{V_j}{h_j} W_{ij}} \tag{3.28}$$

3.4.3 *SPH discretisation of momentum equation*

As discussed in section 3.3.1, the internal forces and source terms in the standard SPH-SWEs model are not fully balanced. In order to balance the internal forces and source terms, the following discretisation scheme for source term in momentum (3.12) is adopted:

$$\mathbf{S}_b = -g \sum_{j=1}^{N} \frac{V_j}{h_j} (b_j - b_i) \nabla W_{ij} \tag{3.29}$$

The internal forces are the same as that in the standard SPH. It is trivial to demonstrate that the residual acceleration in condition I and condition II are equal for both constant smooth length and variable smooth length, i.e. $\mathbf{a}_{r,i} = -gh_i \sum_{j=1}^{N} A_j W_{ij}$. Thus the internal force and source term are balanced, as defined previously.

To reduce the unphysical oscillations on the numerical solution, an artificial viscosity term is added to the momentum equation. In this paper, the artificial viscosity proposed by Monaghan [1997] is used, which is given as

$$\Pi_{ij} = -\frac{0.5 v_{sig} \mathbf{u}_{ij} \mathbf{r}_{ij}}{\bar{h}_{ij} ||\mathbf{r}_{ij}||} \tag{3.30}$$

where $\mathbf{u}_{ij}$ is the relative velocity, $\mathbf{r}_{ij}$ is the distance vector, $\bar{h}_{ij}$ is the average water depth, and v_{sig} is the signal velocity given as

$$v_{sig} = c_i + c_j - 2\mathbf{u}_{ij}\mathbf{r}_i j/||\mathbf{r}_{ij}|| \tag{3.31}$$

where $c = \sqrt{gh}$ is the wave speed.

The bed friction term is calculated as

$$\mathbf{S}_f = -g\frac{n^2 \mathbf{u}_i ||\mathbf{u}_i||}{h_i^{4/3}} \tag{3.32}$$

Taking into account the bed friction term and artificial viscosity, the SPH discretized momentum equation is finally obtained

$$\frac{D\mathbf{u}_i}{Dt} = -\sum_{j=1}^{N}(gV_j + V_j\Pi_{ij})\nabla W_{ij} - g\sum_{j=1}^{N}\frac{V_j}{h_j}(b_j - b_i)\nabla W_{ij} - g\frac{n^2\mathbf{u}_i||\mathbf{u}_i||}{h_i^{4/3}} \tag{3.33}$$

3.4.4 *Variable smoothed length*

Constant smoothed length is widely used in SPH models. However, when solving the SWEs, constant volume and mass of the each particle imply that the change of water depth may lead to substantial expansion or contraction of particles during a simulation. Therefore it is desirable to employ a variable smoothed length in order to maintain better solution accuracy, which is adopted in this work. As proposed in Benz [1990], a variable smoothed length may be calculated by

$$\hbar = \hbar_0 (\frac{h_0}{h})^{1/d_m} \tag{3.34}$$

where d_m is the number of spatial dimensions, $\hbar_0$ and h_0 are the initial smoothed length (two times of the initial particle distance) and water depth, $\hbar$ and h are the new smoothing length and water depth respectively.

3.4.5 *Boundary treatment*

In SPH, implementing boundary conditions, especially solid wall boundaries, has been a challenging problem. A number of different approaches have been reported in the literature for handling solid wall or closed boundary conditions. For example, Monaghan [1994] proposed an approach by distributing a series of ghost particles or virtual particles outside the solid boundaries. These particles possess a high repulsive force and hence prevent the particles inside the computational domain from penetrating the solid boundaries. Libersky et al. [1993] introduced image particles with opposite normal velocities to reflect symmetrical surface boundary conditions. An illustration of two different types of solid boundary treatment is given in Figure 3.5. In this work, these two methods are used together for imposing boundary conditions in all of the test cases under consideration.

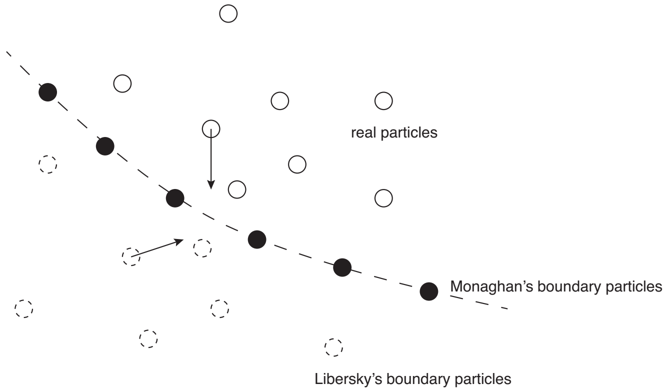


Figure 3.5: An illustration of two different types of solid boundary treatment.

For inflow boundary conditions, virtual particles are generated outside the boundary during the simulation. As can be seen in Figure 3.6, five layers of virtual particles placed outside the boundary are moving into the computational domain. Once the front layer enters the domain, another layer of particles is generated at the end of the queue. If the flow is subcritical, we fix either the depth or velocity of the virtual particles outside the boundary and the other variable is calculated from the particles inside the domain using SPH interpolation. If the flow is supercritical, we fix both the depth and velocity.

For outflow boundary conditions, the particles are removed from the computational domain once they moves out of the boundary.

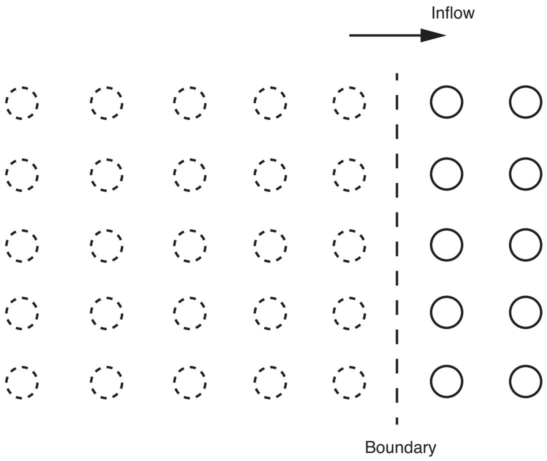


Figure 3.6: An illustration of the inflow boundary treatment.

3.4.6 *Time integration*

A second-order Runger-Kuta method is used for time integration:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \frac{\Delta t}{2}(\mathbf{a}^{n+1} + \mathbf{a}^n) \tag{3.35}$$

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \frac{\Delta t}{2}(\mathbf{u}^{n+1} + \mathbf{u}^n) \tag{3.36}$$

where n represents the time level, Δt is the time step and $\mathbf{a}$ is the acceleration of the particle under consideration. The stability of this explicit time integration is strictly restricted by the time step. In this work, an adaptive approach is used and the time step is predicted according to the following equation:

$$\Delta t = \text{CFL} * \min\left\{\frac{\hbar_i}{\max(\|\mathbf{u}_i\|, \sqrt{gh_i})}\right\} \tag{3.37}$$

where CFL is the Courant number and is a constant with a value less than 1 but greater than 0.

3.5 Validation

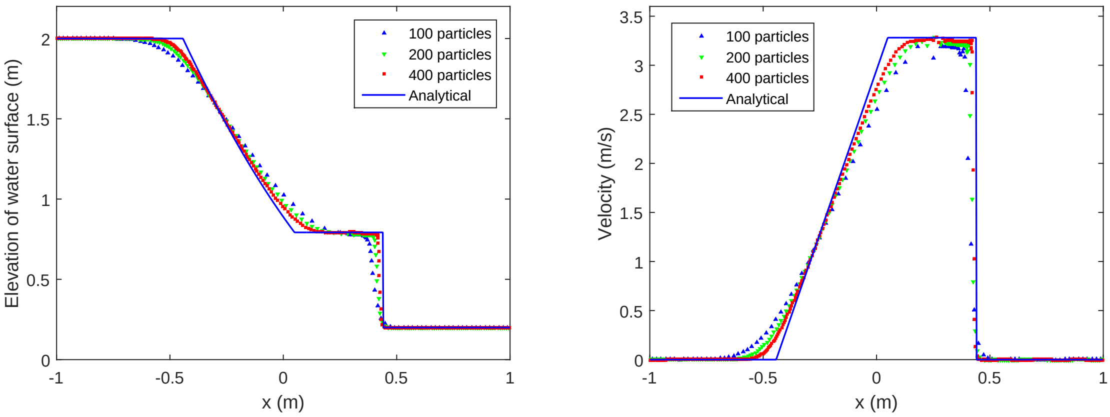
In the section, the new well-balanced SPH-SWEs model featured by (3.22) and (3.33) is validated by applying it to simulate several idealised benchmark tests.

3.5.1 *1D dam-break problem (wet bed)*

A 1D dam-break problem with flat bed topography is first simulated to demonstrate the shock-capturing capability of the new corrected SPH model. In a $[-1, 1]$ domain, the initial water depth is given by

$$h = \begin{cases} 2 & , x < 0 \\ 0.2 & , x > 0 \end{cases} \tag{3.38}$$

Analytical solution for this simple dam-break test was derived by Stoker [1958], which consists of a backwards-propagating rarefaction and a forwards-moving shock wave. Three SPH simulations are run, respectively with 100, 200 and 400 particles. Figure 3.7 shows the numerical results produced by different simulations in terms of water surface elevation and velocity. Generally, satisfactory agreement with the analytical solution is achieved for all of the simulations although numerical diffusion is observed near the solution with steep surface gradient. The numerical predictions seem to converge to the analytical solution when more particles are included in the simulations.



(a)

(b)

Figure 3.7: Numerical and analytical solutions of the 1D dam-break problem (wet bed) at t = 0.1 s: (a) Elevation of water surface; (b) velocity.

3.5.2 *1D dam-break problem (dry bed)*

In this section, the previous wet-bed dam-break test is modified to test the capability of the current SPH-SWEs model in handling wetting front and small water depth. The initial conditions are modified, such that

$$h = \begin{cases} 2 & , x < 0 \\ 0 & , x > 0 \end{cases} \tag{3.39}$$

The simulation is run for $t = 0.1$ s after the dam instantaneously fails and the results are shown in Figure 3.8. Generally, the numerical outputs agree satisfactorily with the analytical solution. However, because of the implementation of variable smoothed length, the particle space at the front is large due to the small water depth, which leads to poor resolution at the wave front. This problem may be resolved by introducing a particle splitting method Vacondio et al. [2012b].

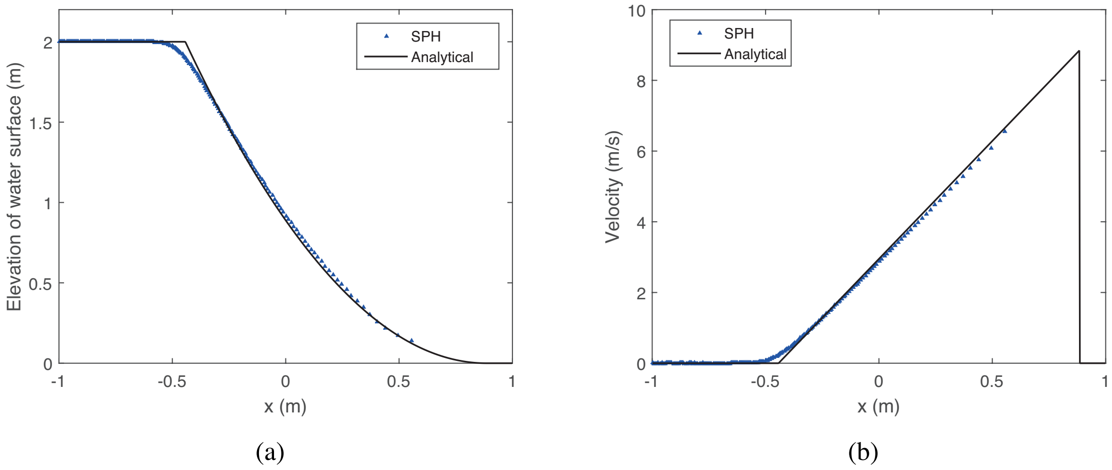


Figure 3.8: Numerical and analytical solutions of the 1D dam-break problem (dry bed) at t = 0.1 s: (a) Elevation of water surface; (b) velocity.

3.5.3 *1D lake at rest test*

It is also necessary to test the capability of the current SPH-SWEs model in preserving well-balanced solutions. For this purpose, a 1D lake at rest test is considered. The bottom topography features a bump in the middle and is defined as

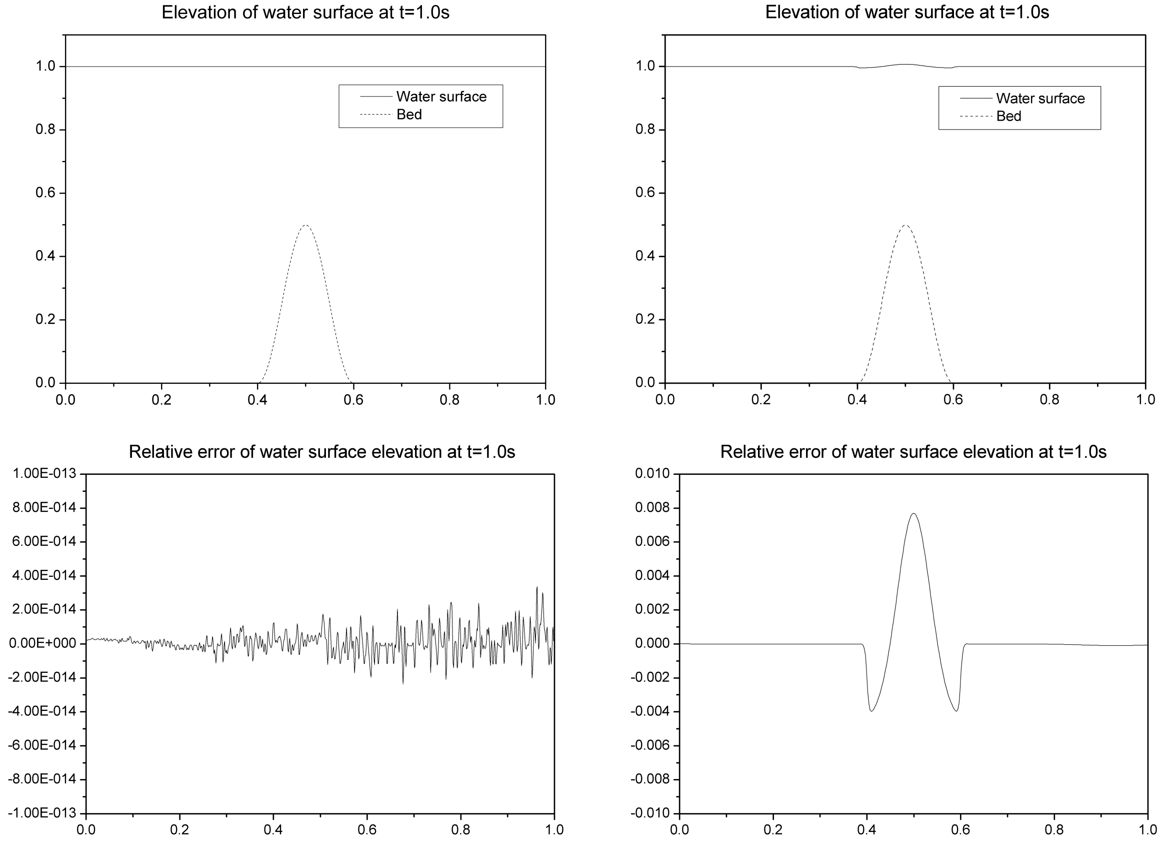
$$b = \begin{cases} 0.25\cos\pi(x-0.5)/0.1 + 0.1, & |x-0.5| < 0.1 \\ 0, & \text{otherwise} \end{cases} \tag{3.40}$$

which is used by many other researchers (e.g. LeVeque [1998], Audusse et al. [2004]).

The initial flow conditions are given by

$$h + b = 1, \mathbf{u} = \mathbf{0} \tag{3.41}$$

Simulations are carried out for up to $t = 1.0$ s using both the well-balanced SPH and standard SPH models and the results are shown in Figure 3.9. The well-balanced SPH model is observed to preserve perfectly the steady-state still water surface. The relative error of water surface elevation is quantified at the order of 10^{-14}, which is at the same level as the machine precision. However, the performance of the standard SPH model is much less attractive and the still water conditions have evidently been disturbed with the relative error of water surface elevation calculated to an order of 10^{-3}.



(a) (b)

Figure 3.9: Numerical solution of 1D lake at rest test at t = 1.0 s, produced by (a) well-balanced SPH-SWEs model; (b) standard SPH-SWEs model.

3.5.4 *1D dam-break over a step.*

In this section, a 1-D dam-break over a step is simulated to validate the model's ability to treat discontinuous bed topography. The bottom topography is given as:

$$b = \begin{cases} 1.0 & , x > 0 \\ 0 & , x < 0 \end{cases} \tag{3.42}$$

And the initial conditions are given as:

$$h = \begin{cases} 1.0 & , x > 0 \\ 4.0 & , x < 0 \end{cases} \quad , \mathbf{u} = \mathbf{0} \tag{3.43}$$

The analytical solution of this test is provided in Alcrudo and Benkhaldoun [2001], which is featured by a backward-propagating rarefaction, a forward-moving shock wave and the abruptly change of water depth at the bottom step. Herein, this problem is simulated by the well-balanced SPH model and standard SPH model with 1000 particles. The numerical results are compared with the analytical solution in Figure 3.10. Generally, the result from the well-balanced SPH model agrees well with the analytical solution. Using the standard model, however, spurious oscillations are predicted near the bottom step and the water depth at the left hand side of bottom step does not match well with the analytical solution.

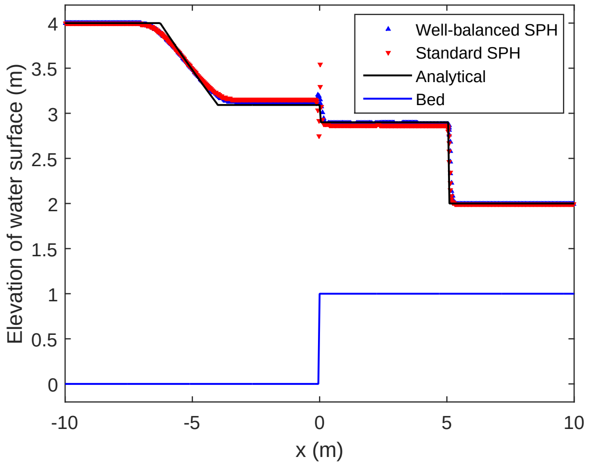


Figure 3.10: Numerical and analytical solutions of the 1D dam-break problem over a bottom step at $t = 1.0$ s.

3.5.5 *Perturbed water surface to a 2D lake at rest test*

This test involves a small perturbation to the still water surface, which is again a classic numerical test adopted by many other researchers (e.g. LeVeque [1997], Liang and Borthwick [2009], Noelle et al. [2006]). The computational domain becomes $[x, y] \in [0, 2] \times [0, 1]$ with four open lateral boundaries. The bottom topography is given as

$$b(x, y) = 0.8 \exp[-5(x - 0.9)^2 - 50(y - 0.5)^2] \tag{3.44}$$

which is shown in Figure 3.11. The initial perturbed still water surface are described as

$$h = \begin{cases} 1.01 - b & ,0.1 < x < 0.2 \\ 1 - b & ,\text{otherwise} \end{cases} \quad , \ \mathbf{u} = \mathbf{0} \tag{3.45}$$

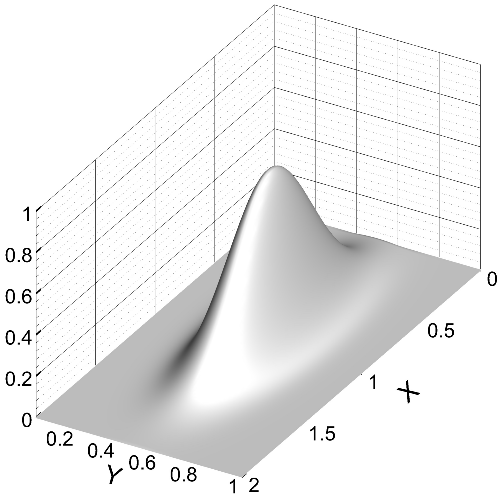


Figure 3.11: 2D perturbed lake at rest: bed topography.

As in the previous tests, both the standard SPH-SWEs model and well-balanced SPH-SWEs model are used to simulate this test with 400×200 particles. Figure 3.12 presents the numerical predictions, compared with the FV reference solution (Liang [2010]). As the perturbed water surface moves towards the hump as a shock-like wave front, the central part slows down due to the reduced wave speed as a result of smaller water depth. Consequently, the wave surface shoots up in order to conserve energy. The wave pattern induced by wave-topography and wave-wave interaction is actually very complex. As shown in Figure 3.12, the numerical prediction resulting from the well-balanced SPH-SWEs model is able to capture most of the small scale wave structures and agrees closely with that predicted by the FV, although the SPH solution appears to be slightly more diffusive due to the use of artificial viscosity. However, this is not the case for the standard SPH-SWEs model. As it is clear from Figure 3.12 (a), the result calculated by the standard model is physically meaningless, where the error of water depth caused by the change of topography is at the same order as the magnitude of perturbation.

3.5.6 *Dam break over three humps.*

This test case considers a dam-break wave over an initially dry floodplain with three humps, which was proposed by Kawahara and Umetsu [1986]. This problem is simulated herein to

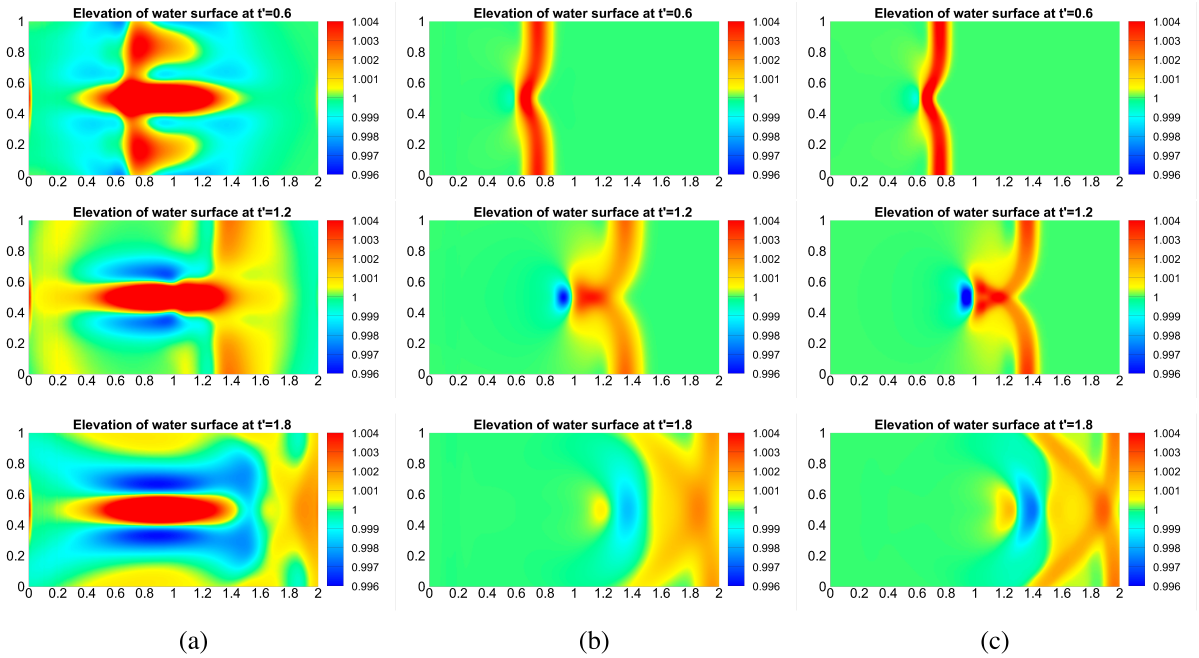


Figure 3.12: 2D perturbed lake at rest: numerical solutions at different output times obtained by different models: (a) standard SPH-SWEs model; (b) well-balanced SPH-SWEs model; (c) FV model.

demonstrate the capability of the current well-balanced SPH model in handling wetting and drying. The bed topography of the 70 m × 30 m rectangular domain is defined as:

$$\begin{aligned} b(x,y) = \max[0, 1 - \frac{1}{8}\sqrt{(x-30)^2 + (y-6)^2}, \\ 1 - \frac{1}{8}\sqrt{(x-30)^2 + (y-24)^2}, \\ 3 - \frac{3}{10}\sqrt{(x-47.5)^2 + (y-15)^2}] \end{aligned} \tag{3.46}$$

Initially the dam is located at $x = 16$ m and retains a still water body with level of $\eta = 1.875$ m in the reservoir upstream of the dam. The dam collapses instantly at $t = 0$. A constant Manning coefficient of 0.018 s/m$^{1/3}$ is used during the entire simulation. 10000 particles are used in the simulation for this challenging problem that involves complex flow hydrodynamics, bottom topography and wetting and drying. Solid wall boundary conditions are imposed at all four boundaries.

The numerical results at different output times are shown in Figure 3.13 in terms of 3D water surface and depth contours. At $t = 2.0$ s, the wave front has reached the two small hills and begun to climb over them. Meanwhile, curved bore-like reflection is created and begins to propagate to upstream while the central part of the wave front continues to propagate forwards through the space between the two small hills. At t = 6.0 s, the two small hills are entirely submerged and the wave front has hit the large hill. The curved reflective bores continue to move upstream and begin to interact with each other and the side walls of the container. At t = 12 s, the wave front passes through both sides of the large hill and begins to submerge the lee of the hill. Moving further upstream, the reflective shock-waves from the small hills have developed into a bore with a front close to a straight line. At the meantime, the maximum water level over the large hill decreases as the water retreats. Finally the flow becomes steady due to the dissipation caused by bed friction, as shown at t = 300 s when the flow is nearly motionless and the peaks of the hills are no longer submerged.

3.6 GPU acceleration

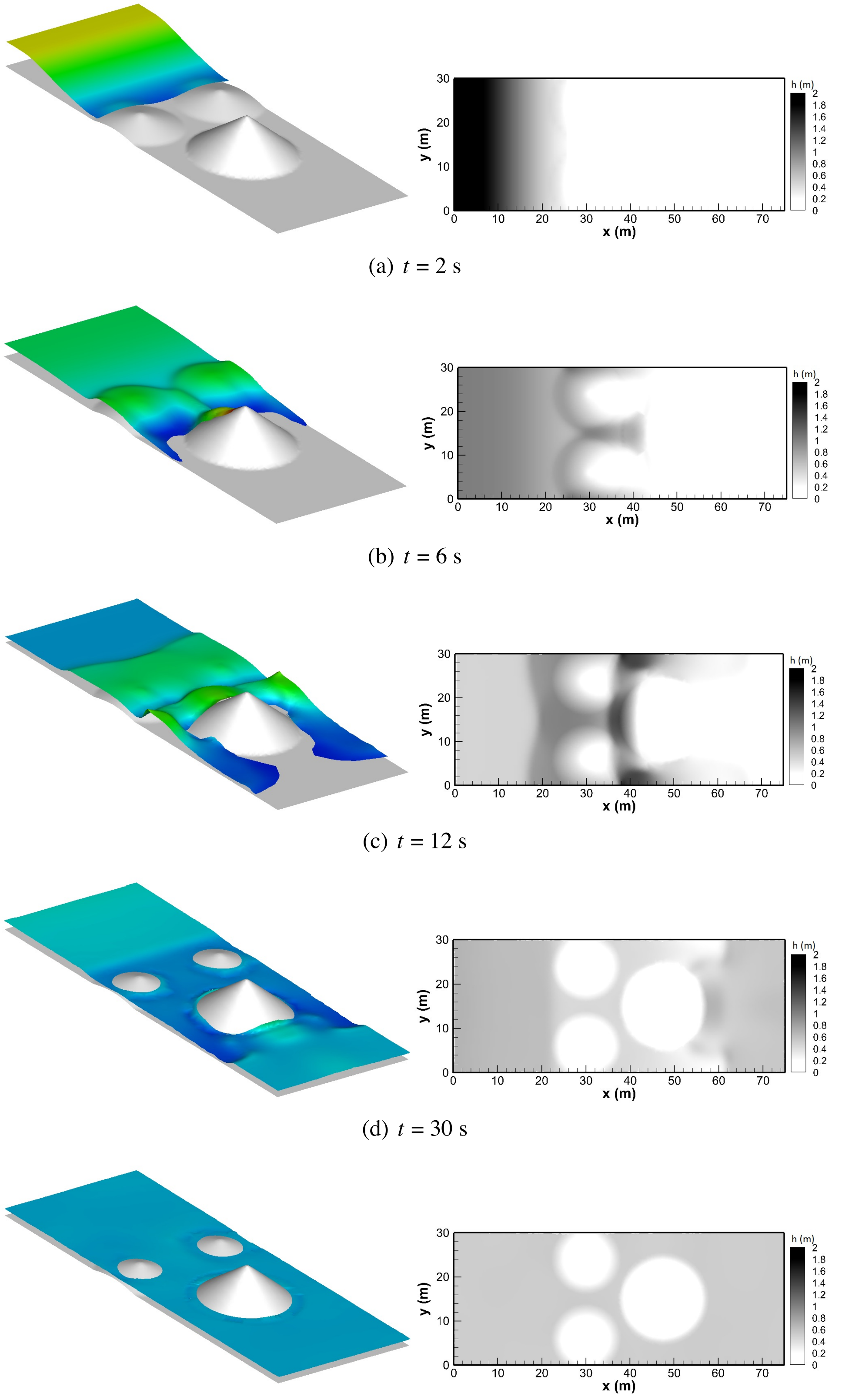
GPUs are used for accelerating the current well-balanced SPH-SWEs model so that it can simulate large scale problems more efficiently. This section presents the details about implementing the SPH-SWEs model on GPUs.

3.6.1 *CUDA architecture*

In this work, CUDA is chosen to implement the SPH-SWEs model on GPUs. CUDA is a parallel programming and computing platform released by NVIDIA to enable scientific computing on the NVIDIA GPUs. A CUDA-enabled GPU contains streaming multiprocessors (SMs) and each of them is composed of multiple processors known as CUDA cores. Each SM has its own on-chip memory called shared memory. Furthermore, all SMs share the on-board memory known as global memory. Shared memory has a much higher bandwidth (1.3TB/s compared with 177GB/s for NVIDIA Tesla M2075) but a much lower latency than the global memory (20-40 clock cycles compared with 200-400 clock cycles for NVIDIA Tesla M2075) (NVIDIA [2015]).

The CUDA hardware platform is accessible to software developers through CUDA-accelerated libraries, compiler directives (such as OpenACC), and extensions to industry-standard programming languages, including C, C++ and Fortran. C/C++ programmers may choose ‘CUDA C/C++’, compiled with ‘nvcc’, i.e. NVIDIA’s LLVM-based C/C++ compiler. Fortran users can adopt ’CUDA Fortran’, compiled with the PGI CUDA Fortran compiler from the Portland Group. The current SPH-SWEs code is written in CUDA C/C++.

A CUDA program is generally composed of several host and kernel functions. Host func-



(a) $t = 2$ s

(b) $t = 6$ s

(c) $t = 12$ s

(d) $t = 30$ s

(e) $t = 300$ s

Figure 3.13: Numerical solution of the 2D dam break over three humps at different output times in terms of 3D water surface and depth contours.

tions run on CPUs (hosts) and call kernel functions that are entirely performed on GPUs (devices). A kernel function is executed on a device of multiple instances that are called threads and threads are grouped into multiple blocks. When mapping the model software to the hardware platform, blocks are mapped to SMs and the threads inside a block are mapped to the CUDA cores inside an SM. Threads in a block can communicate through the shared memory and global memory while blocks can only communicate through the global memory. Unlike CPUs, GPUs feature with executing many concurrent threads slowly rather than running a single thread fast.

3.6.2 *Structure of the SPH-SWEs code on GPUs*

As illustrated in Figure 3.14, the current GPU-accelerated SPH-SWEs model may be divided into six components that are related to input, neighbour searching, calculating depth and force, integration and output. The input data including initial conditions, values of those physical parameters and topographic information is firstly read into the host and then copied onto the memory on the device. Then the water depth and forces are calculated at the same time when identifying the neighbouring particles by a grid searching method. Finally, data is copied back to the host memory and written onto files on the hard drive for post-processing if necessary. Except for the input and output, the code is fully executed on the device to avoid the overhead introduced by the communication between the host and the device.

In CUDA-enabled GPUs, data access to the global memory is based on the units of 64 bytes or 128 bytes. More precisely speaking, threads are being executed on GPUs in the units of a warp (32 threads) and each warp reads 64 bytes (8 double precision numbers) or 128 bytes (16 double precision numbers) of data (a chunk) at one time. If the data that a warp needs is in different chunks, the warp will have to read the global memory more than once. Accessing to global memory is only fully optimised or fully coalesced when the data requested by a warp is within one chunk. In order to optimise the data assessing efficiency, data is stored on the device by means of SoA (Structure of Arrays) rather than AoS (Array of Structures). In AoS, variables of a particle such as the position and velocity are firstly grouped into a structure, and then all these ‘structures’ are arranged together into an array to store the data in the memory. On the contrary, in SoA, different variables are not grouped together. Values of one variable of all particles, e.g. velocity, are firstly grouped into an array, and then all these similar ‘arrays’ are grouped into a structure. Storing the data in SoA can compact the data needed by a warp

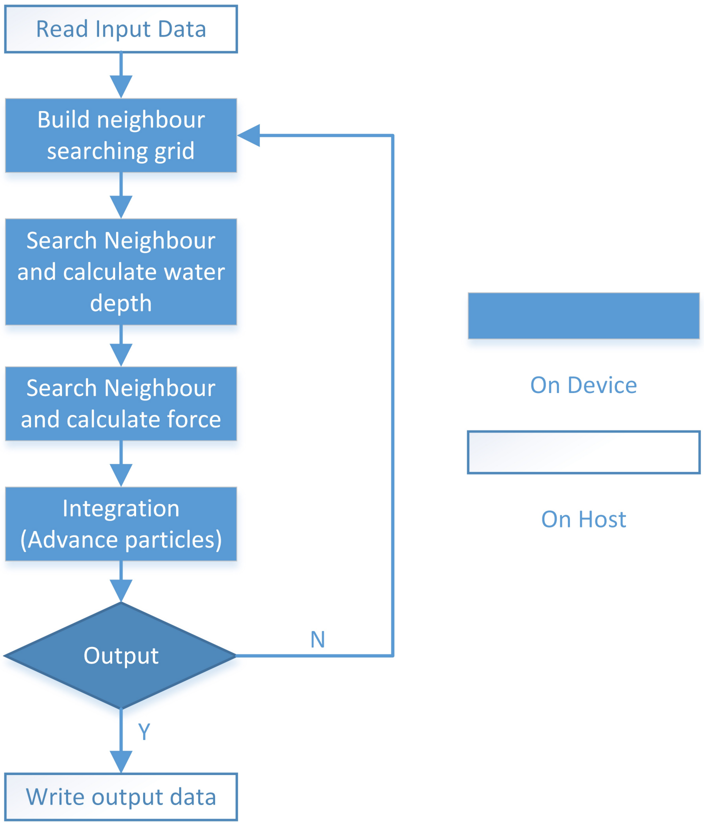

Read Input Data
Build neighbour searching grid
Search Neighbour and calculate water depth
Search Neighbour and calculate force
Integration (Advance particles)
Output
N
Y
Write output data
On Device
On Host

Figure 3.14: Flowchart of the GPU-accelerated SPH code.

into two chunks and make data accessing become more coalesced.

3.6.3 *GPU Implementation of the Key Solution Procedures*

As indicated in the previous section, four out of six parts of the current SPH-SWEs code are designed for execution on the device, which will be introduced in more details in this section.

1. Building neighbour searching grid

In SPH, an arbitrary particle must interact with a finite number of neighbouring particles in order to update its own properties to the next time step and so it is necessary to identify these neighbours. The simplest approach is direct searching. For a given particle i in a domain containing N particles, particles from $j = 1$ to N are checked one after another to identify those within an impacting radius (usually 2 or 3 times of smoothing length) of particle i. The corresponding computational time of direct searching is $O(N^2)$, which is not efficient.

Uniform grid searching represents a more efficient neighbour searching technique for an SPH algorithm with a constant smoothing length. This approach has been widely used for SPH simulation of incompressible free surface flows (Domínguez et al. [2013a]). By implementing this approach, all of the particles are firstly fit into a uniform grid with a width that equals the maximum searching radius (usually 2 or 3 times of the maximum smoothing length of all particles). For an arbitrary particle i, all of its effective neighbouring particles must be located in the cell containing particle i and its neighbouring cells, as shown in Figure 3.15. The computational time associated with the uniform grid searching is $O(N)$, which is much more efficient than direct searching thus adopted in this work.

As shown in Figure 3.16 (a), when implementing a uniform grid searching method, each particle is given a unique particle index and each grid cell is also provided with a grid index. Because the size of each SPH particle is infinitely small, it can only be within a single cell. The index of a cell in which the particle of interest is located is identified and stored in an array called *hash*, as shown in Figure 3.16 (b). The *particle index* array is then sorted according to the hash array as indicated in Figure 3.17. The hash array and particle index array provide the necessary information for determining the grid structure to find cell neighbours. For better memory coalescing, all corresponding data (volume, depth, velocity and position) are redistributed according to the sorted particle index to ensure that particles spatially close to each other have similar indices.

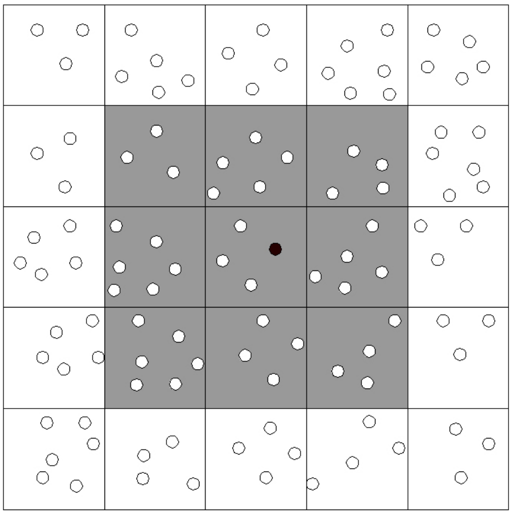
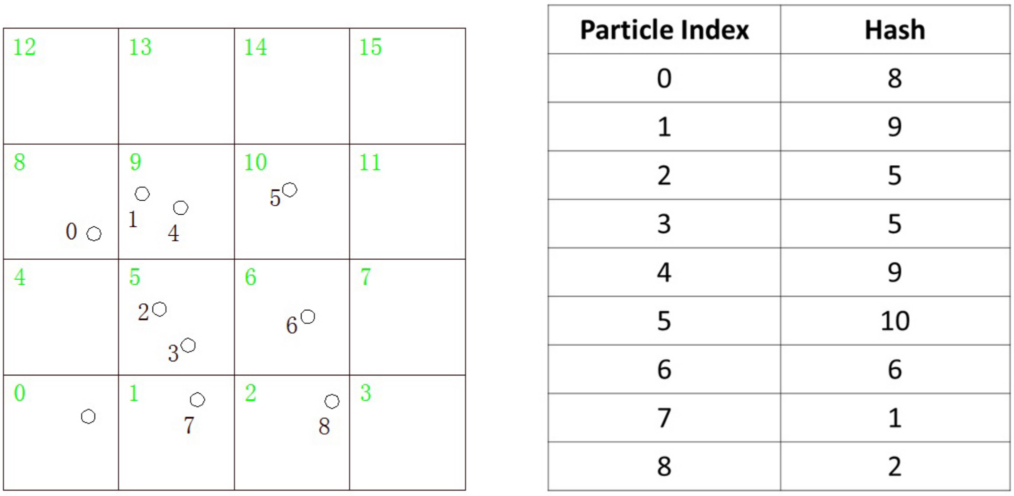


Figure 3.15: Uniform grid searching method.

2. Searching neighbour and calculating depth

The following simple algorithm is used to determine the first and last particles in each grid cell: any two contiguous elements in the array *hash* are compared; obviously, the particles they refer to belong to the same cell if they have the same value; on the contrary, they belong



Particle Index	Hash
0	8
1	9
2	5
3	5
4	9
5	10
6	6
7	1
8	2

(a) (b)

Figure 3.16: Example of uniform grid searching method: (a) uniform grid; (b) particle indices and the corresponding hash array storing the cell indices

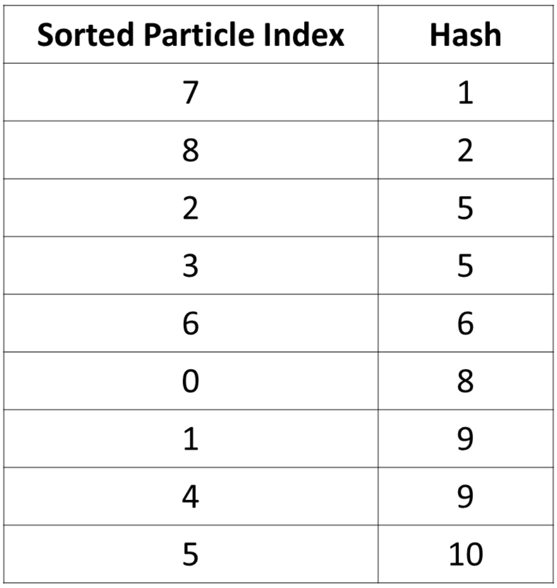
to different cells if they have different values. For example, element i has a value n while element $i+1$ has a value $n+1$, which means that particle i belongs to cell n and particle $i+1$ belongs to cell $n+1$. Because the particles are already sorted in a sequential manner, particle i is the last particle of cell n, and particle $i+1$ is the first particle of cell $n+1$. After knowing the first and last particles of each cell, it is straightforward to sweep all the particles in each cell to identify neighbours. Once the cell neighbours have been found, they are used to calculate depth.

3. Searching neighbour and calculating forces

This component simply repeats the previous one with the exception that the forces are calculated to update the momentum.

4. Integration

In the integration component, each thread is allocated a particle to perform the calculation. The data accessing process is fully coalesced because the data that each thread needs is sequential. After integration, the reduction algorithm proposed by Sanders and Kandrot [2010] is employed to find the maximum velocity for updating the time step according to specific stability criteria.



Sorted Particle Index	Hash
7	1
8	2
2	5
3	5
6	6
0	8
1	9
4	9
5	10

Figure 3.17: Sorting particle indices according to *hash*

3.7 Application to flood simulation and comparison with finite volume method

In this section, the GPU accelerated SPH-SWEs model is applied to simulate flood waves on real world topography and compared with an intensively validated second-order Godunov-type FV model (Liang [2010]) in terms of accuracy and efficiency. The FV code is written in CUDA and runs on GPUs (Amouzgar et al. [2014]). We also compare the GPU-accelerated SPH-SWEs model with its CPU implementation written in FORTRAN and parallelised through OpenMP. For the simulations, the CFL number is set as 0.2 for SPH and 1.0 for FV respectively, which are the maximum value found to be adequate for stabilising simulations.

3.7.1 *Malpasset dam-break flood*

Fiirstly, the current well-balanced SPH-SWEs model is applied to simulate a real-world dam-break flood. Malpasset dam was located in a narrow gorge of the Reyran river valley and collapsed at night on 2nd December 1959. It was a double curvature arch dam of 66.5 m maximum height, with a crest length of 223 m. The maximum capacity of the reservoir was designed to be 55106 m^3. The flash flood following the dam-break caused more than 400 casualties. After the disaster, a survey was done by local policemen to record the high water marks on both banks of the valley at the locations as shown in Figure 3.18 where the location of dam is also indicated.

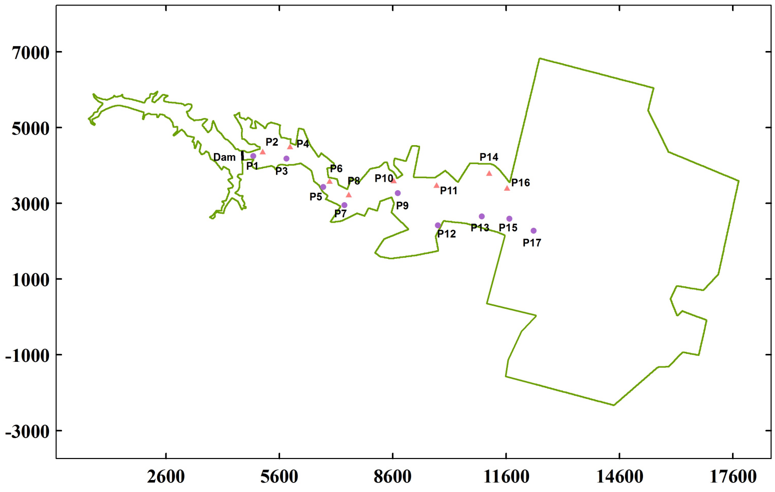


Figure 3.18: Malpasset dam-break flood: locations of the dam and police survey points.

We use two different spatial resolutions during the SPH simulations. The initial distance between particles are 10 m and 20 m with 6000 and 25000 particles respectively. The FV simulation uses 10 m spatial resolution, which is comparable with the initial particle distances in the SPH simulations with 0.5 million cells in total. The uniform Manning coefficient is set to be 0.033 s/m$^{1/3}$ throughout the simulations. The predicted flood event is presented in Figure 3.19 and Figure 3.20. We can see that the flood water travels downstream rapidly along the valley and the water level in the reservoir reduces gradually after the dam break. The high-resolution SPH simulation performs similar to the FV simulation but the flow front moves slower in the low resolution SPH simulation. The simulated highest water levels are compared with those obtained from the police survey in Figure 3.21, where the SPH simulations give results slightly different from the FV simulations but they all agree reasonably well with the police survey data. In order to quantify their accuracy, we compared the Root Mean Square Error (RMSE) of different simulations, which is defined as

$$RMSE = \sqrt{\frac{\sum(\eta_i - \hat{\eta}_i)^2}{N}} \tag{3.47}$$

where η denotes the computed water level, $\hat{\eta}$ denotes the measured water level, i is the survey point index and N is the total number of survey points. The RMSEs are summarised in Table 3.1. Three different simulations give generally comparable RMSEs, and the best agreement is obtained by the SPH simulation with 25000 particles while the SPH simulation with 6000 parti-

Table 3.1: Malpasset dam-break flood: RMSEs for simulated highest water level at 17 survey points.

SPH high resolution	SPH low resolution	FV
2.89 m	3.47 m	3.43 m

cles gives the largest error. The results imply that the current well-balanced SPH-SWEs model is robust for real world dam-break flood simulation and can give results that are comparable with Godunov-type FV models.

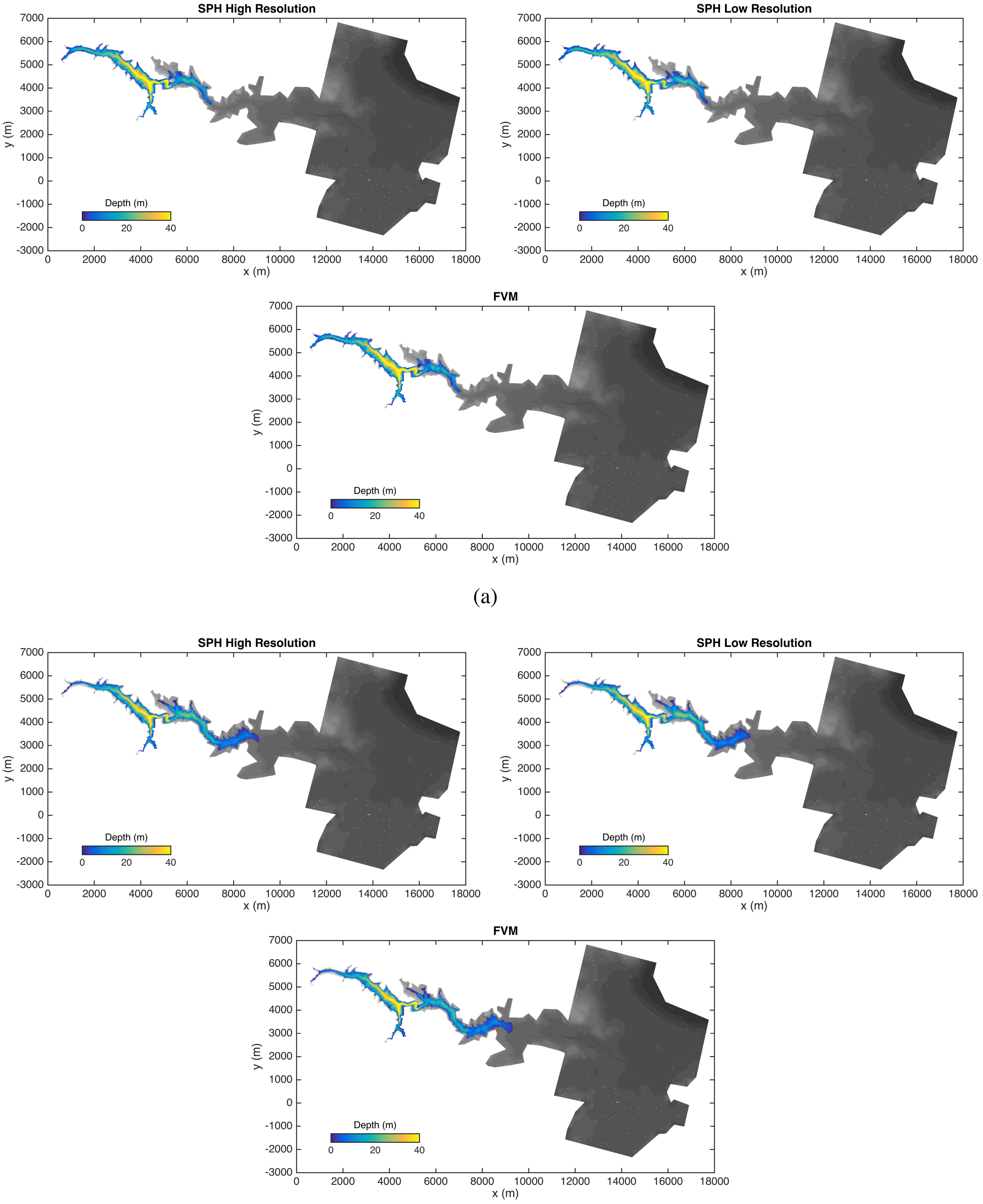
The run time of different models are compared in Table 3.2. For SPH simulations with both high and low resolutions, the GPU code running on a Tesla M2075 GPU is about 3-4 times faster than the CPU code running on a Xeon E5620 quad-core CPU. The speedup ratio indicates that the GPU acceleration is effective. The FV simulation finishes in 7 minutes, which is only about 10% of the time needed by the low resolution SPH simulation and 2% of the time consumed by the high resolution SPH simulation. Such a big difference can be explained by the fact that an SPH particle interacts with about 50 to 100 surrounding particles but a FV cell only interacts with 4 neighbouring cells.

Table 3.2: Malpasset dam-break flood: runtime on different devices by different models

Devices	SPH high resolution	SPH low resolution	FV
Xeon E5620	1124 min	250 min	N/A
Tesla M2075	305 min	71 min	6 min

3.7.2 *Thamesmead hypothetic flood*

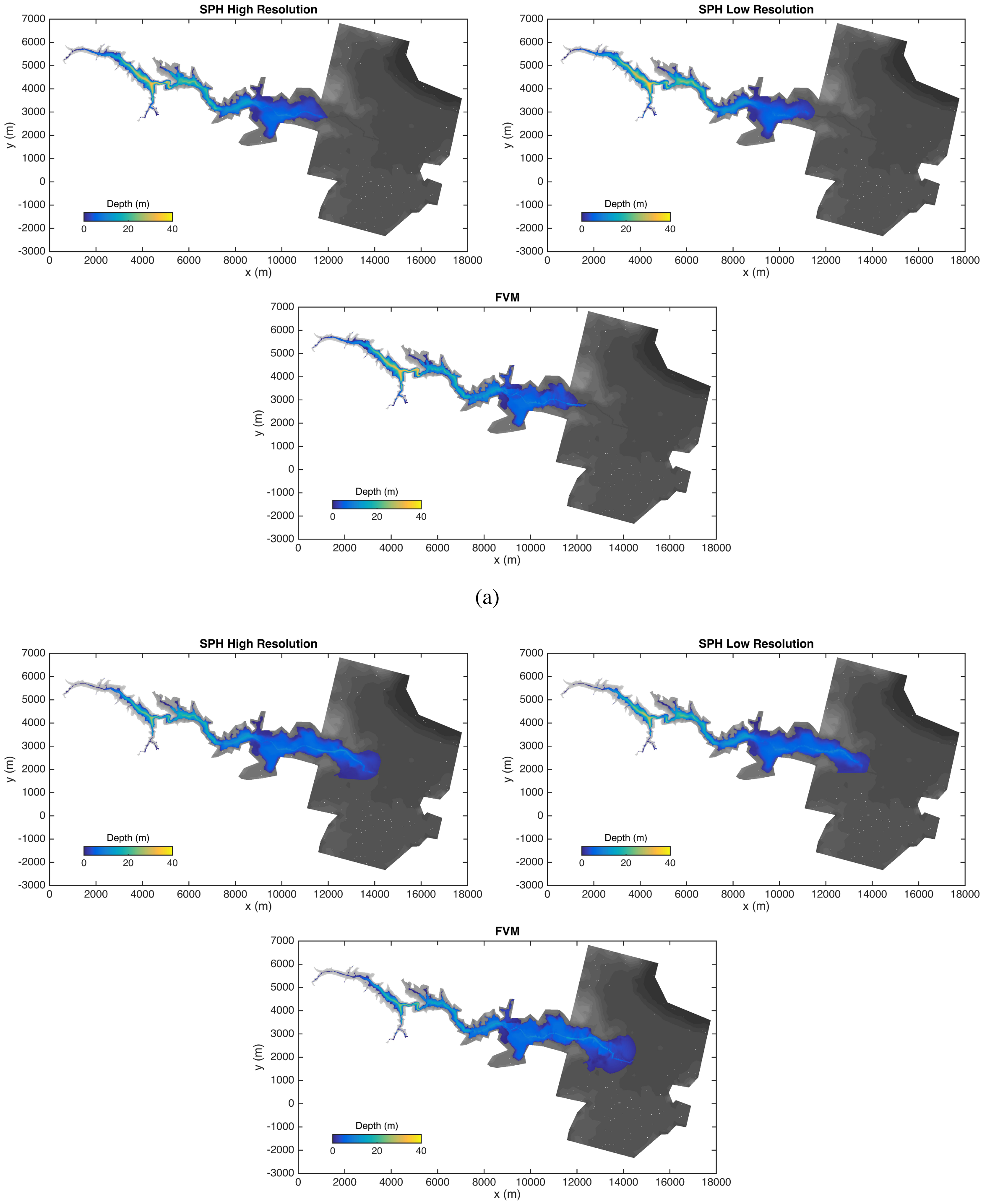
The floodplain is located on the south bank of the River Thames estuary in the United Kingdom. The topography is described by a digital elevation model (DEM) at a 2 m resolution, as shown in Figure 3.22 (a), where the urban domain features, e.g. buildings and roads, are clearly represented. In the floodplain, those features that cannot be properly represented by the DEM, e.g. small tunnels and alleyways for pedestrian access, are modified to allow the flood water to flow through. A 150 m wide breach on the flood defence is assumed and located at the northern part of the floodplain, as indicated in Figure 3.22 (a), through which the prescribed inflow discharge provided in Figure 3.22 (b) is imposed to drive a flood event. Four gauge points, as marked in Figure 3.22 (a), are used for recording temporal change of flood depth during the event. A constant Manning coefficient $n = 0.035$ s/m$^{1/3}$ is assumed for the whole domain.



(a)

(b)

Figure 3.19: Malpasset dam-break flood: inundation maps predicted by different models at different times: (a) $t = 5$ minutes. (b) $t = 10$ minutes.



(a)

(b)

Figure 3.20: Malpasset dam-break flood: inundation maps predicted by different models at different times: (a) t = 20 minutes. (b) t = 30 minutes. (continued from Figure 3.19).

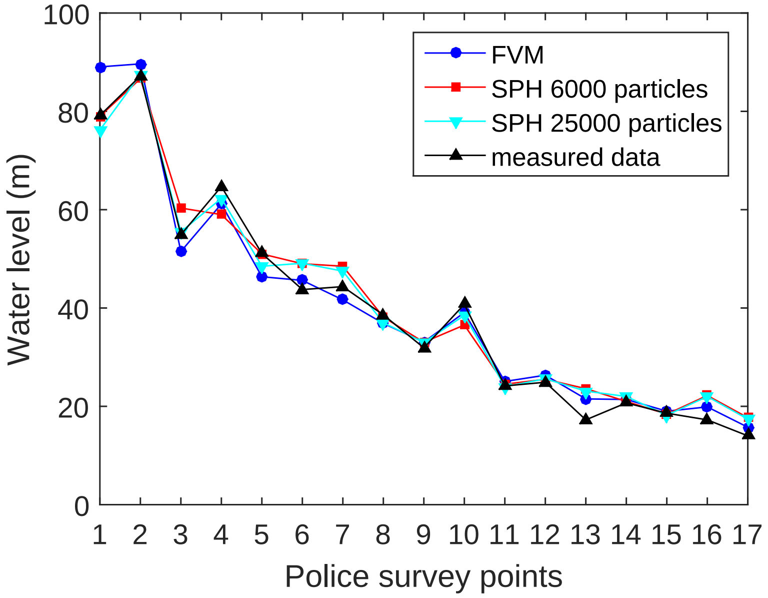
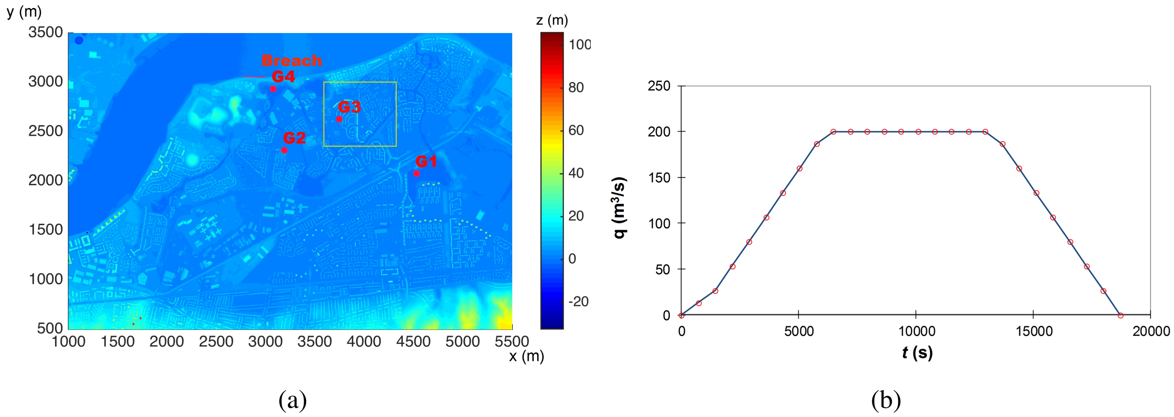


Figure 3.21: Malpasset dam-break flood: comparing simulated water level with police survey data.

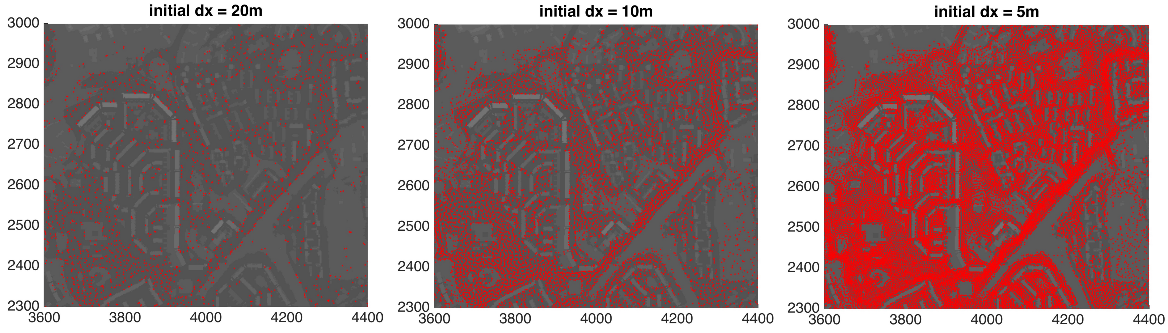


(a)

(b)

Figure 3.22: Thamesmead hypothetic flood: (a) floodplain; (b) inflow discharge hydrograph.

All of the simulations are carried out for 5.2 hours after the flood defence breaches. During the simulations, new particles are being generated with an initial distance of 5 m, 10 m and 20 m, respectively, to simulate the inflow discharge through the breach. The water depth of the newly generated particles is fixed as 0.45 m (the depth of the breach) and the velocity is calculated from the hydrograph of the inflow discharge based on the water depth. At the end of the simulations, the total number of particles inside the domain is respectively 13,000, 60,000 and 230,000. Figure 3.23 shows the particle distribution in the squared region of Figure 3.22 (a). When the initial particle distance is 20 m, it can be observed that the buildings resolved at this resolution block the particles, which consequently travel only through the larger roads between the buildings. Obviously, the complex urban features, e.g. buildings and roads, play an important role in regulating the flood flow in the SPH simulation. However, because too few particles are used, the shapes of the urban features are not well-defined in the flood inundation map. When reducing the initial particle distance to 10 m, more particles come into the domain during the flood event and the buildings and roads are certainly better represented than the 20m simulation. When the initial distance is further reduced to 5 m, it can be now clearly seen that numerous particles flow through and cover the paths between buildings and accumulate at the low-lying regions. The urban features are clearly represented. Figure 3.24 presents the corresponding inundation maps of different resolution at t = 5.2 h, comparing with the solution obtained by the FV model, at 2 m resolution (Figure 3.24(d)). The SPH prediction with initial resolution at 5 m generally agrees well the FV reference solution at higher resolution except for certain regions where very small water depth exists. The time histories of the water depth at the four gauges are shown in Figure 3.25. The result of the simulation with 5 m initial distance matches well with the FV simulation in terms of arriving time and magnitude. In terms of arriving time, the largest discrepancy is observed in gauge 1, in which the SPH solution is 20 minutes later than the FV prediction. In all other gauges, the discrepancy is less than 10 minutes. This level of uncertainty may be acceptable in relation to the duration of the event, i.e. 5.2 h. Regarding the peak water depth, the differences at the four gauges are 28 cm, 20 cm, 18 cm and 30 cm respectively, corresponding to a relative error of 10% – 15%. Although higher resolution may not always lead to better agreement because the particles among the streets are already dense enough as seen in Figure 3.23 (c), much greater discrepancy is observed between the lower-resolution results and the FV simulation in this case. Therefore, the highest resolution result of the SPH simulation is favourable in this test case. It is still not entire clear why certain level of difference is presented between the SPH and FV simulations. A possible reason may be that SPH models are generally numerically more diffusive than a FV scheme. Generally, the SPH simulation with 5 m initial distance is comparable with the FV simulation.

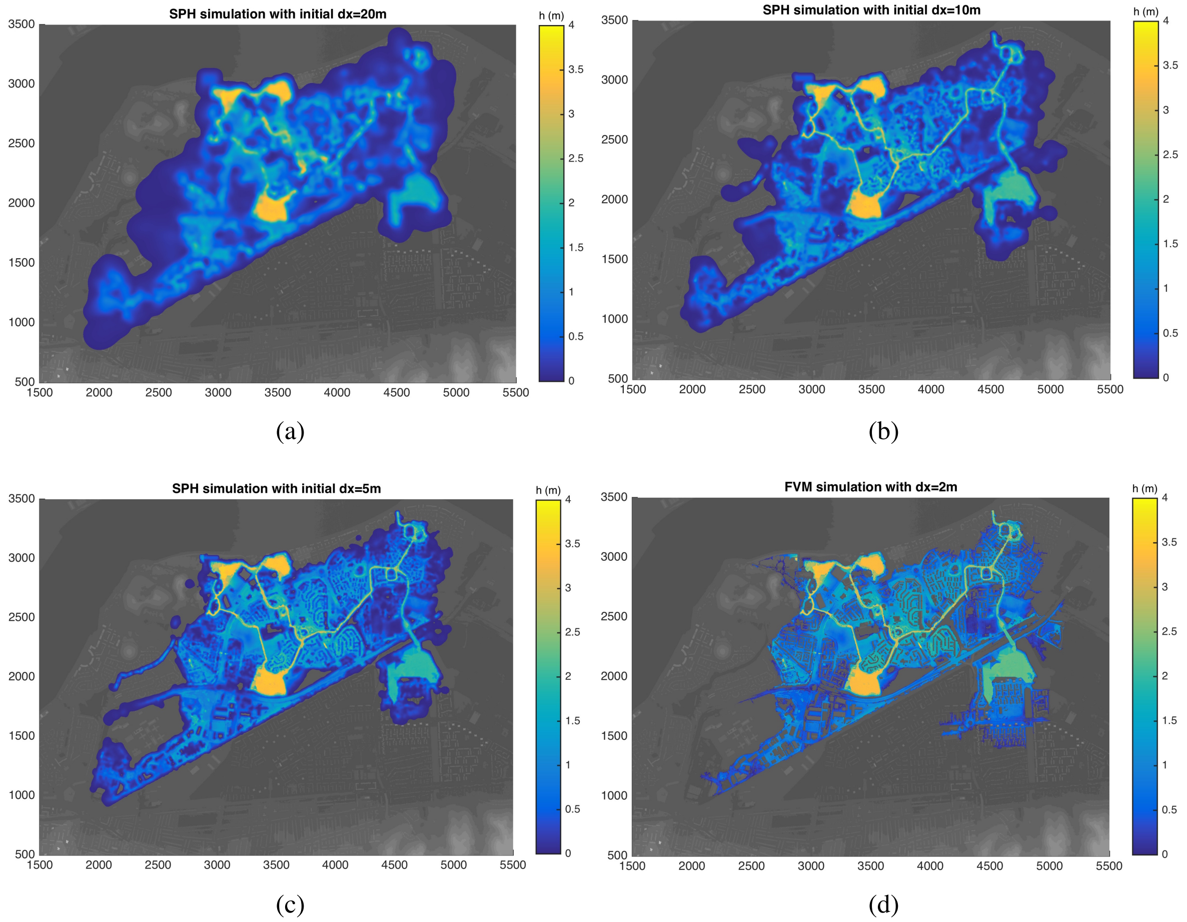


(a) (b) (c)

Figure 3.23: Thamesmead hypothetic flood: particle distribution of the SPH simulations at different initial resolutions in a selected region: (a) Δx = 20m; (b) Δx = 20m; (c) Δx = 5m.

The runtime on different devices is provided in Table 3.3. Compared with those OpenMP parallelised simulations on a quad-core Xeon E5620 CPU, it is evident that the SPH-SWEs model has been substantially accelerated on GPUs: for the simulations with initial resolution of either 20 m or 10 m, the speedup is about 9 times; when initial particle distance reduces to 5 m, the runtime on CPU becomes unacceptably long and the simulation was terminated after it ran for 200 hours. As can be observed from the simulation results, favourable results can be only obtained when the initial particle distance becomes 5 m or less. The runtime of the SPH-SWEs model on a Tesla M2075 GPU for the finest resolution is 135 hours, which is 54 times of the runtime of FV on the same GPU. For the SPH simulations, it is interesting to note that runtime increases significantly as the particle distance reduces from 10m to 5m in this case. The runtime increases about 40 times, which is much bigger than the expected 8 times as a result of using 4 times more particles and half of the time step. A possible explanation is that the smoothing length of certain particles becomes large and constrains the minimum grid size for searching neighbours. This may demand unnecessary extra effort when searching and sorting neighbour particles.

This test case also shows that SPH may need to use an initial particle distance comparable with the spatial resolution of FV to accurately capture the effects of highly irregular topography like buildings in urban areas. But the high resolution simulation by SPH is significantly slower than FV and becomes unacceptable in practical simulations.


SPH simulation with initial dx=20m
SPH simulation with initial dx=10m
SPH simulation with initial dx=5m
FVM simulation with dx=2m
h (m)
0
0.5
1
1.5
2
2.5
3
3.5
4
500
1000
1500
2000
2500
3000
3500
1500
2000
2500
3000
3500
4000
4500
5000
5500

(a)

(b)

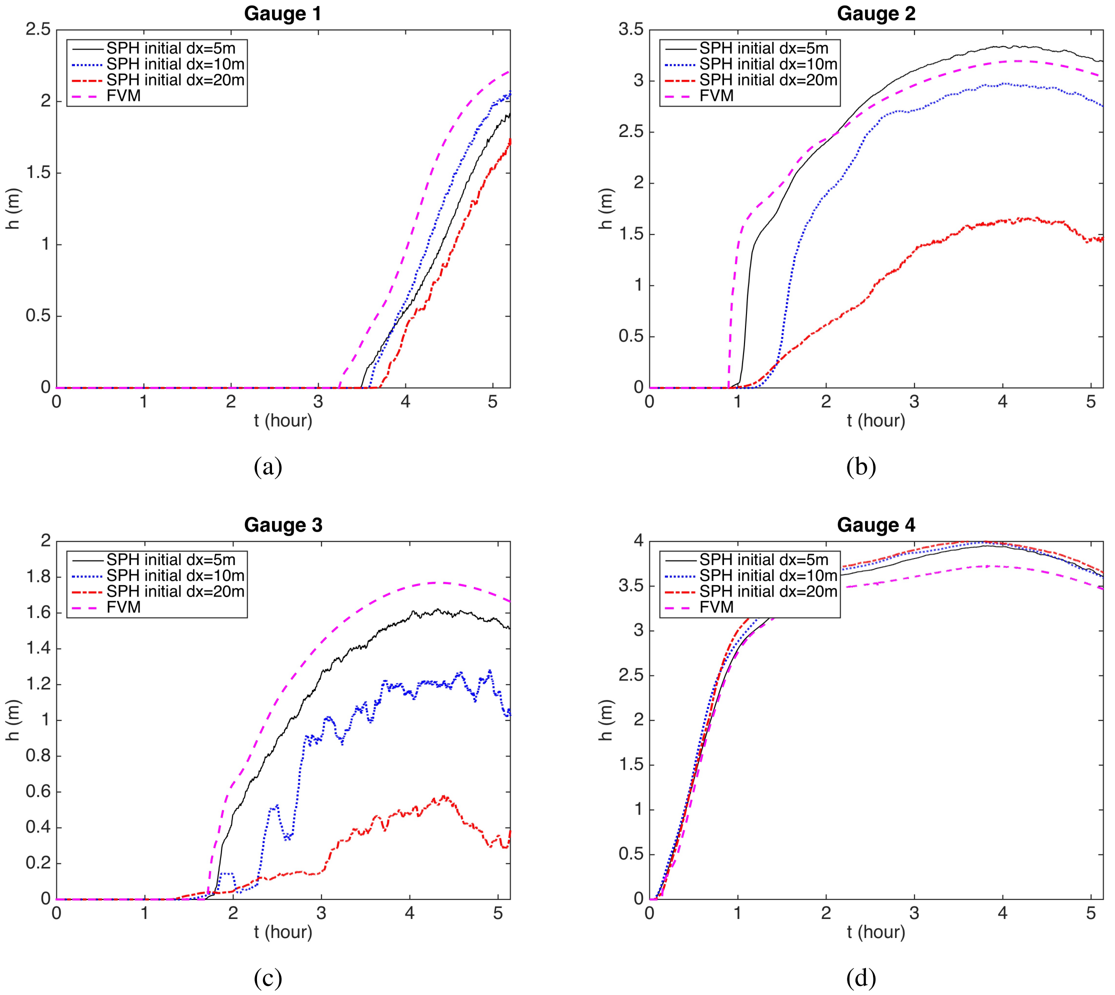
(c)

(d)

Figure 3.24: Thamesmead hypothetic flood: inundation maps at $t = 5.2$ hours: (a) SPH simulation with $\Delta x = 20$m; (b) SPH simulation with $\Delta x = 10$m; (c) SPH simulation with $\Delta x = 5$m; (d) FV simulation with $\Delta x = 2$m.

Table 3.3: Runtime for the Thamesmead hypothetic flood on different devices by different models

Devices	SPH (20 m)	SPH (10 m)	SPH (5 m)	FV
Xeon E5620	4.5 h	32 h	≥ 200 h	N/A
Tesla M2075	0.5 h	3.5 h	135 h	2.5 h



(a)

(b)

(c)

(d)

Figure 3.25: Thamesmead hypothetic flood: time histories of water depth at: (a) gauge 1; (b) gauge 2; (c) gauge 3; (d) gauge 4.

3.8 Conclusions

In this chapter, a new SPH model for solving the SWEs has been developed and compared with a Godunov-type finite volume model. In order to improve the accuracy of the standard formulation of SPH for solving the SWEs, the well-balanced issue has been identified and resolved. At the computational level, the steady-state solution of lake at rest is violated by the discretisation error introduced by the SPH formula for approximating the continuity equation. In this work, an analytical expression of the error has been derived, which contains the particle approximation error, surface variation error and bed variation error. The bed variation error leads to the violation of the well-balanced solution. A new corrected SPH model is then derived by subtracting the bed variation error from the standard SPH continuity formula. The SPH discretised momentum equation has also been modified to balance the source term and internal force. However, because of the truncation error induced by particle approximation, the modified SPH momentum equation may still not guarantee absolute well-balanced solution and the updated water particles may be slightly perturbed even though the initial conditions ensure horizontal water surface elevation and zero velocity. Fortunately, numerical tests show that the influence of the truncation error is not significant. The corrected continuity and momentum equations lead to a new SPH-SWEs model that conserves the lake at rest solution. Due to its well-balanced and shock-capturing capabilities, the new SPH-SWEs model is capable of simulating different types of practical shallow flow problems involving complex domain topography and moving wet-dry interface. The model performance has been demonstrated by applying it to simulate a series of idealised benchmark tests. The new SPH-SWEs model is then accelerated using GPUs.

Finally, the GPU accelerated SPH-SWEs model is applied to simulate two flood events over real world topography and compared with a Godunov-type FV model. If an initial particle distance comparable with the spatial resolution of FV is used, the numerical results is comparable with those of the FV model. However, the simulation is significantly slower (about 50 times) than the FV model. Because SPH does not show obvious superiority in terms of accuracy and stability and is much slower, it is less favourable to use an SPH model for large-scale simulations. Therefore, the traditional Godunov-type finite volume method will be chosen as the numerical method for the physically-based catchment-scale multi-hazard model.

Chapter 4. An efficient and stable hydrodynamic model with novel source term discretisation schemes for overland flow simulations

4.1 Introduction

Since rainfall-runoff and overland flows play a central role in the formation of surface flooding and initiation of landslides and debris flows, the proposed multi-hazard modelling system must be able to simulate overland flows robustly. However, as is already stated in chapter 2, existing popular schemes for solving the shallow water equations suffer from some problems that undermines the accuracy, efficiency and stability for overland flows simulations. These problems are:

1. Prevailing schemes may calculate bed slope terms incorrectly.

2. Prevailing schemes may lead to unrealistic large velocities if the topography is discontinuous.

3. Prevailing schemes cannot relax the velocity to equilibrium in a single time step when the relaxation time scale is much smaller than the the hydrodynamic timescale.

4. Prevailing schemes may give wrong discharges for flows on slopes when the friction is dominant.

This chapter aims to resolve these problems and develop a shallow flow model with novel source term discretisation. The numerical techniques presented in this chapter will be used in the proposed multi-hazard model while the whole shallow flow model can be used as a standalone tool to simulate flash floods.

4.2 Governing equations

The matrix form of the 2D SWEs with source terms may be written as

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \mathbf{R} + \mathbf{S}_b + \mathbf{S}_f \tag{4.1}$$

where **q** contains the flow variables, **f** and **g** are the flux vector terms in the x and y-directions, **R**, $\mathbf{S}_b$ and $\mathbf{S}_f$ are the the different source terms representing respectively the rainfall rate, bed slope and frictional effect. The vector terms are given by

$$\mathbf{q} = \begin{bmatrix} h \\ uh \\ vh \end{bmatrix} \mathbf{f} = \begin{bmatrix} uh \\ u^2h + \frac{1}{2}gh^2 \\ uvh \end{bmatrix} \mathbf{g} = \begin{bmatrix} vh \\ uvh \\ v^2h + \frac{1}{2}gh^2 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} R \\ 0 \\ 0 \end{bmatrix} \mathbf{S}_b = \begin{bmatrix} 0 \\ -gh\frac{\partial b}{\partial x} \\ -gh\frac{\partial b}{\partial y} \end{bmatrix} \mathbf{S}_f = \begin{bmatrix} 0 \\ -\frac{\tau_{bx}}{\rho} \\ -\frac{\tau_{by}}{\rho} \end{bmatrix} \tag{4.2}$$

where g is the gravitational acceleration, h is the water depth, b is the bed elevation, u and v are the depth-averaged velocities along the x and y directions, R is the rainfall rate, ρ is the water density, and τ_{bx} and τ_{by} are the frictional stresses estimated using the Manning fomula:

$$\tau_{bx} = \rho C_f u\sqrt{u^2 + v^2} \quad \tau_{by} = \rho C_f v\sqrt{u^2 + v^2} \tag{4.3}$$

in which C_f is the roughness coefficient evaluated using

$$C_f = gn^2/h^{1/3} \tag{4.4}$$

where n is the Manning's roughness number. If water depth is very small ($\sim 10^{-3}$ m), surface tension and capillary waves also become important. But they are currently not explicitly considered in this work. Instead, their effects are all packed into the Manning's friction term.

4.3 A brief review of the existing schemes

In this section, we give a brief review of certain existing numerical schemes solving the SWEs for overland flow simulations. However, there is no intention to provide an exhausted review herein and we will limit our scope to those prevailing and representative schemes and further explain the challenges as mentioned in the introduction.

4.3.1 *Bed slope discretisation*

For solving the SWEs for overland flow modelling, the first effective numerical scheme to be mentioned herein is the hydrostatic reconstruction introduced in Audusse et al. [2004]. Along the x-direction, let i and $i+1$ denote the two adjacent cells under consideration. The first step of hydrostatic reconstruction is to define a single bed elevation at the cell interface as

$$b_{i+1/2} = \max(b_i, b_{i+1}) \tag{4.5}$$

based on which the Riemann states of water depth across the cell interface are defined as

$$h_{i+1/2-} = \max(h_i + b_i - b_{i+1/2}, 0)\ , h_{i+1/2+} = \max(h_{i+1} + b_{i+1} - b_{i+1/2}, 0) \tag{4.6}$$

The corresponding bed slope source term is subsequently derived and discretised to become

$$S_{bx} = (\frac{1}{2}gh^2_{i+1/2-} - \frac{1}{2}gh^2_{i-1/2+})/\Delta x \tag{4.7}$$

where Δx is the cell size. This scheme preserves exactly the lake at rest solution and the positivity of water depth if an appropriate Riemann solver is adopted. (4.7) may be reformulated as follows

$$S_{bx} = -\frac{1}{2}g(h_{i+1/2-} + h_{i-1/2+})\frac{h_{i-1/2+} - h_{i+1/2-}}{\Delta x} \tag{4.8}$$

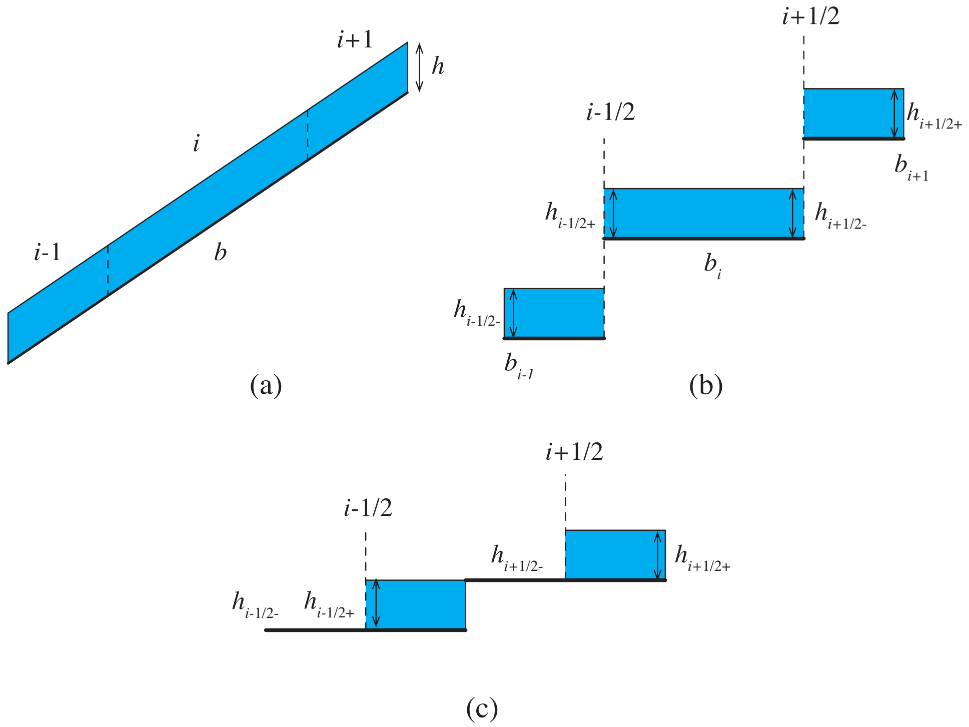
where $\frac{h_{i-1/2+} - h_{i+1/2-}}{\Delta x}$ may be viewed as the discretisation of the term related to bed slope, i.e. $\frac{\partial b}{\partial x}$. Apparently, no matter how steep the actual bed slope is, the computed value for $\frac{\partial b}{\partial x}$ is $\frac{h_{i-1/2+} - h_{i+1/2-}}{\Delta x}$, which essentially imposes a constrained condition on the bed slope gradient. Although this effectively avoids the large velocities that would otherwise occur when the bed slope is excessively steep, it unfortunately also introduces certain unwanted side effects for overland flow simulation which will be explained as below.

Consider a slope with a constant gradient and uniform water depth, as illustrated in Figure 4.1 (a). The water depth is smaller than the difference of bed elevation between the two neighbouring cells, i.e. $h < b_{i+1} - b_i$. In such a case, the computed bed slope from (4.6) is clearly milder than the actual slope, as shown in Figure 4.1 (b-c). Following this constrained slope reconstruction, the water depth defined from (4.6) is always zero at the left hand side of a cell interface for this case, as illustrated in Figure 4.1 (c). The local Riemann problem essentially defines a dry-bed dam break and the localised flow condition has been artificially reconstructed to become a waterfall rather than a uniform flow, which is physically incorrect and may lead to misleading prediction. The immediate effects are 1) water can only move in one direction; and 2) the forward-moving signal speed becomes associated with a dry-bed dam break, which is much larger than the flow velocity and will cause overestimation of the discharge. Numerically, it may somehow compensate the underestimation of discharge due to the constricted slope gradient, but the exact amount is impossible to quantify. In the rest of this text, this numerical phenomenon is referred to as 'waterfall effect'. Research efforts (e.g. Morales De Luna et al. [2013], Hou et al. [2014]) have been devoted to resolve this issue of incorrect discretisation of bed slope terms in the original hydrostatic reconstruction method. But all of the reported modified hydrostatic reconstruction methods still suffer from reconstructing water depth to zero at one side of a cell interface for the case as illustrated in Figure 4.1, and the problem of 'waterfall effect' persists.

The idea of constraining slope gradients is also used in other numerical schemes (e.g. Liang and Borthwick [2009], Bouchut and de Luna [2010], Song et al. [2011b]). Although the specific formulations may be different, these approaches tend to encounter the same problem as previously mentioned (Delestre et al. [2012]). The issue related to the incorrect discretisation of bed slope terms is particularly problematic for the first-order schemes. However, the higher order schemes may also be affected because it is a common practice to reduce a scheme to be first-order accurate when the predicted water depth becomes smaller than certain thresholds, in order to stabilise a simulation (e.g. Hou et al. [2013b], Song et al. [2011a], Murillo et al. [2007], Liang [2010]). There are some numerical schemes (e.g. George [2008], Murillo and García-Navarro [2010]) that do not impose a constraint on the slope source terms except at the wet-dry front. While these schemes do not underestimate the slope source terms as the hydrostatic reconstruction does, they also lose the advantage provided by constrained slope gradients to stabilise a simulation over irregular topography with abrupt bed change.

4.3.2 *Friction term discretisation*

In order to remove the stiffness of the friction source terms, a conventional way is to discretise the friction source terms implicitly. In practice, the implicit discretisation is often reformulated



(a)

(b)

(c)

Figure 4.1: The bed reconstruction method of the typical well-balanced schemes where h and b respectively denote the water depth and bed elevation: (a) idealised overland flow over a constant slope; (b) the same case at the discretised level; (c) face values of the water depth and bed elevation after reconstruction.

to derive an effective explicit scheme. Considering a 1D problem, many schemes (e.g. Song et al. [2011b], Busaman et al. [2015], Cea and Blade [2015], Cea and Vazquez-Cendon [2010], Burguete et al. [2008], Liang et al. [2006], Costabile et al. [2013], Singh et al. [2015], Rousseau et al. [2015]) adopt the following discretised equations

$$\hat{q}_x^{n+1} = q_x^n - \frac{\Delta t}{\Delta x}(F_{i+1/2}^n - F_{i-1/2}^n) + \Delta t S_{bx}^n \tag{4.9}$$

$$q_x^{n+1} = \frac{\hat{q}_x^{n+1}}{1 + \Delta t g n^2 (h^n)^{-4/3} |u^n|} \tag{4.10}$$

where $q_x = hu$ is the unit discharge at the x-direction, F is the momentum flux term, S_{bx} is the bed slope term and Δt is the time step.

Assuming a positive velocity, the equilibrium velocity as a result of balanced bed slope and friction terms can be obtained as

$$u_\infty = \sqrt{\frac{S_{bx}}{g n^2 h^{-1/3}}} \tag{4.11}$$

In order to correctly describe an overland flow when the water depth becomes small and the friction effect becomes predominant, it is important to ensure that a numerical scheme can relax the flow velocity to the correct equilibrium state in a single time step. For the implicit scheme as described in (4.9)-(4.10), the local acceleration (flux) term becomes small and negligible when the friction term is predominant. Rescaling (4.10) by taking $\Delta t \rightarrow \infty$, the velocity resulting from the scheme is

$$\hat{u}_\infty = \frac{S_{bx}}{gn^2 h^{n+1}(h^n)^{-4/3}|u^n|} \tag{4.12}$$

Apparently $\hat{u}_\infty$ and u_∞ are not equal, and therefore (4.9-4.10) cannot relax the velocity to the correct equilibrium state in a single time step.

Fractional splitting method has also been used for computing friction source terms. For example, Liang and Marche [2009] proposed the following scheme

$$q_x^{n+1} = q_x^* + \frac{\Delta t S_{fx}^*}{1 + 2\Delta t gn^2 (h^*)^{-4/3}|u^*|} \tag{4.13}$$

where h^*, q_x^* and u^* are updated from the previous time step by solving the frictionless SWEs, S_{fx}^* is the friction term calculated from h^*, q_x^* and u^*. The velocity for $\Delta t \rightarrow \infty$ can be obtained as

$$\hat{u}_\infty = \frac{1}{2}u^* \tag{4.14}$$

which again does not give the correct equilibrium state.

A few numerical schemes also impose an upper bound on the friction terms (e.g. Burguete et al. [2008], Liang and Marche [2009]) so that the friction does not reverse the flow direction, which leads to $\hat{u}_\infty = 0$ and again is different from the correct equilibrium state. The scheme reported by Yu and Duan [2014] manually set the velocity to the equilibrium state as long as the kinematic wave number is larger than a threshold, but the criteria to determine the threshold may be case-dependent, restricting its robustness for applications.

The existence of predominant friction terms may also affect the validity of existing Riemann solvers. In an extreme case, the discharge on slope can be significantly overestimated as already briefly mentioned in section 4.1. Herein the augmented Riemann solver introduced by George [2008] is taken as an example. This Riemann solver is implemented in the framework of the wave propagation algorithm proposed by LeVeque [1997] and has become part of the open-source GeoClaw software (Berger et al. [2011]). The time-marching formula in the wave propagation algorithm is given as

$$\mathbf{q}_i^{n+1} = \mathbf{q}_i^n - \frac{\Delta t}{\Delta x}(\mathcal{A}^+\Delta Q_{i-1/2} + \mathcal{A}^-\Delta Q_{i+1/2}) \tag{4.15}$$

where $\mathcal{A}^{\pm}\Delta Q$ is defined as the fluctuations at the cell interface that are determined by solving a general Riemann problem including initial bed elevation discontinuity. The mass flux, i.e. unit discharge across the cell interface $i + 1/2$ can be expressed by the fluctuations (see (6) in LeVeque [1997]) as

$$F^h_{i+1/2} = \frac{1}{2}(\mathcal{A}^-\Delta H_{i+1/2} - \mathcal{A}^+\Delta H_{i+1/2}) + \frac{1}{2}(hu_i + hu_{i+1}) \tag{4.16}$$

where $\mathcal{A}^{\pm}\Delta H$ is defined as the mass fluctuations, which also satisfy the following relationship (see (5) in LeVeque [1997])

$$\mathcal{A}^-\Delta H_{i+1/2} + \mathcal{A}^+\Delta H_{i+1/2} = hu_{i+1} - hu_i \tag{4.17}$$

In the simple case of steady and uniform flow on a slope with a unit discharge hu_∞, with (4.16) and (4.17) it is not difficult to obtain that

$$F^h_{i+1/2} = \mathcal{A}^-\Delta H_{i+1/2} + hu_\infty \tag{4.18}$$

The readers may refer to the full details of this scheme in George [2008] to verify that $\mathcal{A}^-\Delta H_{i+1/2} > 0$ if the flow is subcritical and $b_i > b_{i+1}$. In such a case, it can be obviously seen from (4.16) that $F^h_{i+1/2} > hu_i$, i.e. the mass flux computed at the cell interface by the Riemann solver is larger than the cell centred discharge. Although the increase of velocity due to the slope source term can be balanced by the friction effect during the time integration, the overestimated mass flux across cell interface cannot be eliminated by the friction effect, which leads to more water flowing across the cell interface than the reality. For Riemann solvers coupled together with the bed slope source terms, the bed discontinuity at the cell interface needs to generate a mass flux to balance the mass flux due the discontinuity of water depth to maintain the lake at rest solutions. But for steady flow featured by the balance between the bed slope and the friction slope, such a mass flux generated by the bed discontinuity becomes a source of error. It is very likely that some other schemes such as the upwind treatment of bed slope (Bermúdez et al. [1998]) and the lateralised HLL Riemann solver (Fraccarollo et al. [2003]) can encounter the same problem of overestimating the mass flux.

Other Riemann solvers that have been widely applied to solve the homogeneous SWEs such as the HLL or HLLC solver, do not seem to suffer from this problem. Because the HLL or HLLC Riemann solver is not coupled with the slope source terms, the calculated mass flux $F^h_{i+1/2}$ is equivalent to the cell centred discharge hu_i. The mass flux remains to be the original discharge as long as the increase of velocity induced by the bed slope is correctly balanced by the friction effect during time integration. But a numerical scheme implemented with the HLL or HLLC Riemann solver may suffer from the aforementioned inaccurate calculation of slope source terms or water fall effect, which must be overcame by incorporating with a proper slope source term discretisation scheme for applications in overland flow modelling.

4.4 Numerical scheme

In this section, a novel first-order Godunov-type finite volume scheme is presented to address the aforementioned challenges in the context of overland flow simulations.

4.4.1 *Finite volume discretisation*

A finite volume scheme solves the integrated form of governing equations and the resulting semi-discretised equation is given by

$$\frac{\partial \mathbf{q}_i}{\partial t} + \frac{1}{\Omega_i}\sum_{k=1}^{N} \mathbf{F}_k(\mathbf{q}) l_k = \mathbf{R}_i + \mathbf{S}_{bi} + \mathbf{S}_{fi} \tag{4.19}$$

where 'i' is the cell index, 'k' is the index of the cell edges ($N = 4$ for Cartesian grids adopted in this work), l_k is the length of cell edge 'k' and Ω_i is the cell area, $\mathbf{F}_k(\mathbf{q}) = \mathbf{f}_k(\mathbf{q})n_x + \mathbf{g}_k(\mathbf{q})n_y$ contains the fluxes normal to the cell boundary with $\mathbf{n} = (n_x, n_y)$ defining the outward normal direction at the cell boundary. Herein, the fluxes $\mathbf{F}$ and slope source term $\mathbf{S}_b$ are computed explicitly, but the friction source term $\mathbf{S}_f$ is treated implicitly. The final time-marching scheme is given as

$$\mathbf{q}^{n+1} = \mathbf{q}^n - \frac{\Delta t}{\Omega_i}\sum_{k=1}^{N} \mathbf{F}_k(\mathbf{q^n}) l_k + \Delta t(\mathbf{R}^n_i + \mathbf{S}^n_{bi} + \mathbf{S}^{n+1}_{fi}) \tag{4.20}$$

in which n denotes the time level and Δt is the time step. The asymptotic behaviour of the discretised equation may be analysed as follows: as $h \to 0$, it is straightforward to infer from non-dimensionalisation analysis that the convective flux term $\mathbf{F}(\mathbf{q})$ is much smaller than other terms and therefore negligible; the corresponding relaxation time is typically very small compared with the time step determined by the CFL condition and therefore $\Delta t \rightarrow \infty$, which is equivalent to rescaling the equations; subsequently, (4.20) reduces to the following equilibrium equation

$$\mathbf{S}_{bi}^{n} + \mathbf{S}_{fi}^{n+1} = 0 \tag{4.21}$$

where the rainfall rate term has been omitted because only the momentum equations are relevant to the analysis. The flow velocities at the new time step can then be recovered from (4.21) as

$$\mathbf{u}^{n+1} = \sqrt{\frac{||\mathbf{S}_{bi}^{n}||}{C_f}} \cdot \frac{\mathbf{S}_{bi}^{n}}{||\mathbf{S}_{bi}^{n}||} \tag{4.22}$$

This is basically the 2D form of the equilibrium state as defined in (4.11). (4.22) is derived without assuming a steady flow and therefore it is able to relax the velocity to the equilibrium state in a single time step when $h \rightarrow 0$ as long as the implicit part of (4.20) is solved exactly. Note that the use of implicit schemes as (4.9-4.10) and (4.13) does not ensure the recovery of the correct equilibrium state because the implicit schemes have been reformulated into explicit forms and the implicit part is no longer solved exactly.

4.4.2 *Interface fluxes and slope source term discretisation*

As discussed in the previous section, the augmented Riemann solvers (e.g. George [2008]) may lead to overestimated discharge on slopes. The classic Riemann solvers that solves the homogeneous SWEs appear to be a more suitable choice and an HLLC approximate Riemann solver is adopted to calculate the interface fluxes in (4.20) (see Liang and Borthwick [2009], Toro [2001] for detailed implementation). The HLLC Riemann solver does not automatically preserve the lake at rest solution and it is necessary to implement the hydrostatic reconstruction scheme (Audusse et al. [2004]) or other effective approaches to modify the Riemann states before solving the local Riemann problems. In this section, a novel surface reconstruction method (SRM) is proposed to integrate with the hydrostatic reconstruction scheme to obtain the Riemann states to define the local Riemann problems across the cell interfaces and overcome the limitations of the existing hydrostatic reconstruction implementations. Specifically, SRM can effectively 1) avoid the 'waterfall effect' when evaluating the fluxes; 2) preserve the original bed variation when necessary for correct prediction of flow velocities; and 3) constrain the bed slope for the case where the topography is 'genuinely' discontinuous to maintain stability.

To implement the proposed SRM, the water surface elevation (defined in an arbitrary cell 'i' as $\eta_i = h_i + b_i$) at cell interfaces is firstly reconstructed to support the derivation of Riemann states. Considering two adjacent cells 'i' and '$i+1$', the SRM reconstructed water surface elevations at left and right hand sides of their common interface are given by

$$\begin{cases} \eta_L = \eta_i + \max[0, \min(b_{i+1} - b_i - \delta b, \eta_{i+1} - \eta_i)] \\ \eta_R = \eta_{i+1} + \max[0, \min(b_i - b_{i+1} + \delta b, \eta_i - \eta_{i+1})] \end{cases} \tag{4.23}$$

with

$$\delta b = b_{i+1/2+} - b_{i+1/2-} \tag{4.24}$$

where $b_{i+1/2+}$ and $b_{i+1/2-}$ are respectively the bed elevation at the right and left hand sides of the cell interface, obtained through slope limited interpolation of the related cell centre values

$$b_{i+1/2-} = b_i + \mathbf{r}_i \Psi(r_i) \nabla b_i \ , b_{i+1/2+} = b_{i+1} + \mathbf{r}_{i+1} \Psi(r_{i+1}) \nabla b_{i+1} \tag{4.25}$$

where $\mathbf{r}$ is the distance vector from the cell centre to central point of the cell interface, $\Psi(r)\nabla b_i$ is the bed gradient restricted by a minmod slope limiter that is defined on a rectangular Cartesian grid for cell 'i' as

$$\Psi(r_i) = \max[0, \min(r_i, 1)] \text{ and } r_i = \frac{f_+ - f_i}{f_i - f_-} \tag{4.26}$$

in which the subscripts $+$ and $-$ denote the upstream and downstream cells, respectively. The interpolated bed elevations are only calculated at the beginning of a simulation to reduce computational cost. The use of a slope limiter herein is crucial for distinguishing the discontinuous and smooth topographies and applying the constraint to the slope gradient at the right place.

From the reconstructed water levels in (4.23), the corresponding states of bed elevation at left and right hand sides of the cell interface can be obtained as

$$\begin{cases} b_L = \eta_L - h_i \\ b_R = \eta_R - h_{i+1} \end{cases} \tag{4.27}$$

The Riemann states can then be deduced by firstly defining a single bed elevation at the cell interface, as suggested by Audusse et al. [2004]

$$b_f = \max(b_L, b_R) \equiv \max(b_i, b_{i+1}) \tag{4.28}$$

based on which the Riemann states of water depth are defined as

$$\begin{cases} h_L = \max(0, \eta_L - b_f) \\ h_R = \max(0, \eta_R - b_f) \end{cases} \tag{4.29}$$

which reinforces the non-negativity of water depth. The Riemann states of the other flow variables (i.e. unit-width discharges) are subsequently obtained

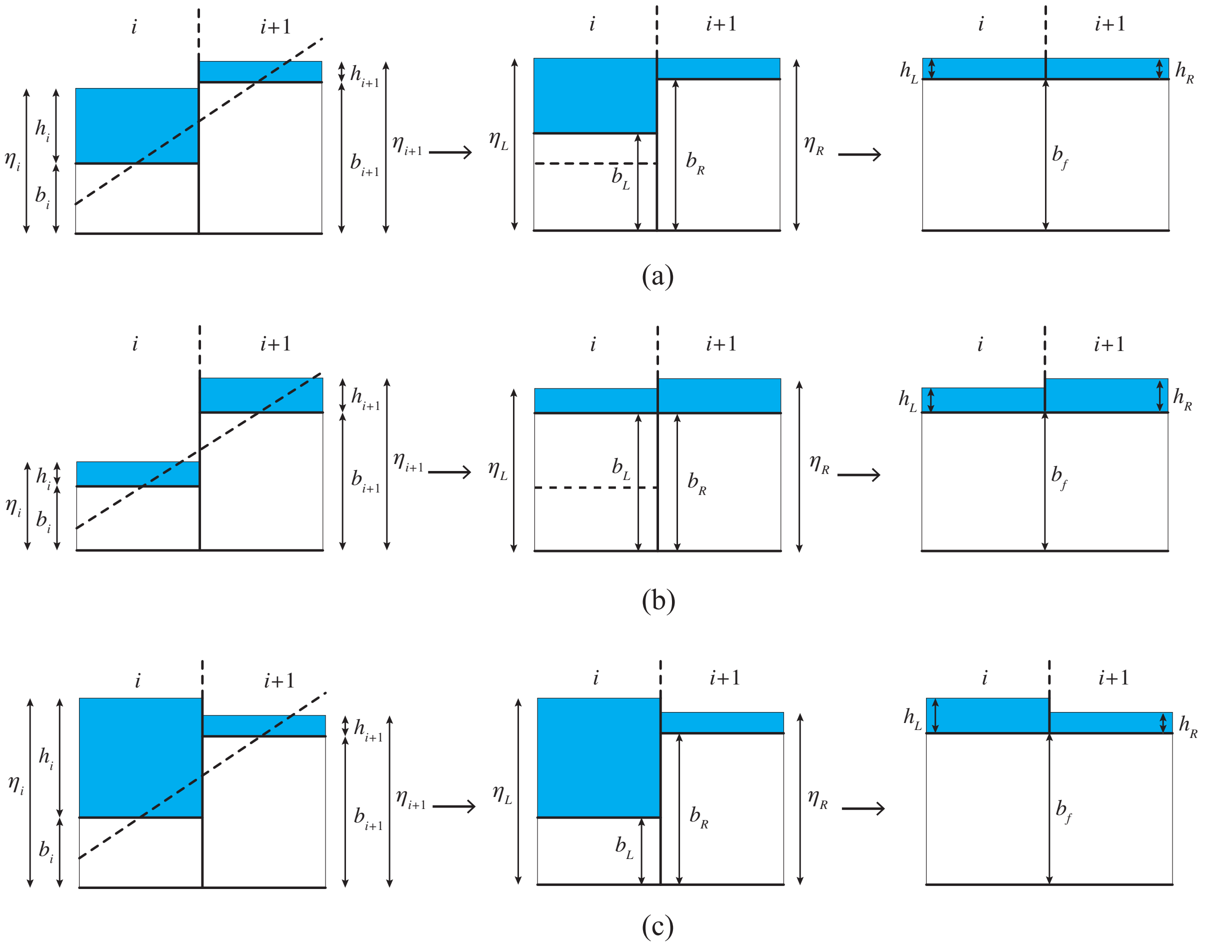
$$\begin{cases} [hu]_L = h_L u_i, & [hv]_L = h_L v_i \\ [hu]_R = h_R u_{i+1}, & [hv]_R = h_R v_{i+1} \end{cases} \tag{4.30}$$

where $u_i = [hu]_i / h_i$ and $u_{i+1} = [hu]_{i+1} / h_{i+1}$, similarly v_i and v_{i+1}, are the velocities defined at the cell centre.

It should be noted that (4.23) has an asymmetric form so that it only modifies the lower water surface elevation between the two. It effectively takes into account all different water surface and bed configurations that are possibly encountered in surface flow simulations. Without losing generality, we assume that $b_i < b_{i+1}$. Because the slope gradient is limited by a minmod limiter, the monotonicity of the bed elevations is preserved and we always have $\delta b \geq 0$.

When $\delta b = 0$, i.e. the slope is constant across the four consecutive cells involved in the slope limited reconstruction in (4.25), there are three possible configurations to consider. Figure 4.2 (a) illustrates the first case where $b_{i+1} - b_i > \eta_{i+1} - \eta_i > 0$. From (4.23), we have $\eta_L = \eta_i + (\eta_{i+1} - \eta_i)$ and so the water surface at the left hand side will be lifted to the same level as that at the right hand side. Figure 4.2 (b) presents the second case where $b_{i+1} - b_i < \eta_{i+1} - \eta_i$. From (4.23), it gives $\eta_L = \eta_i + (b_{i+1} - b_i)$, indicating that the water surface at the left hand side will be raised to eliminate the difference between the bed elevations. Finally, the third case involves $\eta_{i+1} - \eta_i \leq 0$, as shown in Figure 4.2 (c), where (4.23) imposes no modification. With the reconstructed water surface elevations, the Riemann states of water depth are obtained through (4.27-4.29), which are also shown in Figure 4.2. In all of the three cases, the reconstructed water depths at both sides of the cell interface are non-zero; the direction of fluxes depends on the local flow and topographic conditions. Therefore SRM effectively avoids the artificial 'waterfall effect' created by the specific numerical treatment as found in the prevailing hydrostatic reconstruction implementations (e.g. Audusse et al. [2004]).

When $\delta b > 0$, there are also different cases to be considered. When $b_{i+1} - b_i - \delta b >$



(a)

(b)

(c)

Figure 4.2: SRM implementation when $\delta b = 0$ and (a) $b_{i+1} - b_i > \eta_{i+1} - \eta_i > 0$; (b) $b_{i+1} - b_i < \eta_{i+1} - \eta_i$; (c) $\eta_{i+1} - \eta_i \leq 0$. The left, middle and right panels respectively illustrate the original discretisation with the dashed line representing the limited slope, the water surface elevation reconstructed according to (4.23) with the dashed line representing the original location of the bed, and the Riemann states of water depth defined following (4.28-4.29).

$\eta_{i+1}-\eta_i > 0$, (4.23) leads to $\eta_L = \eta_i+(\eta_{i+1}-\eta_i)$ and the resulting water surface reconstruction is identical to Figure 4.2 (a). Similarly, $\eta_{i+1} - \eta_i \leq 0$ defines the case identical to the one as illustrated in Figure 4.2 (c) for $\delta b = 0$. When $b_{i+1} - b_i - \delta b < \eta_{i+1} - \eta_i$, as illustrated in Figure 4.3, (4.23) gives $\eta_L = \eta_i + (b_{i+1} - b_i - \delta b)$, implying that the reconstructed bed elevation b_L on the left hand side is δb lower than b_R (i.e. b_{i+1}) which is untouched during the reconstruction. This can be also confirmed by identifying the reconstructed bed elevations following (4.27): $b_L = \eta_L-h_i = b_{i+1}-\delta b$, $b_R = \eta_R-h_{i+1} = b_{i+1}$, and therefore $b_R-b_L = \delta b$, confirming that the variation between the interpolated bed elevations δb is preserved during the reconstruction. With the reconstructed water surface elevation, the Riemann states of the water depth are then defined from (4.29): $h_L = \max(0, h_i - \delta b)$ and $h_R = h_{i+1}$ as $b_f = b_{i+1}$. Considering $h_L = \max(0, h_i - \delta b)$, it is clear that h_L depends on the magnitude of δb, which essentially quantifies the discontinuity of the bed profile. If the bed curvature is small, δb will be small and it is likely that $\delta b < h_i$, leading to a positive h_L, as shown in Figure 4.3 (a). If the bed curvature is large or there is an abrupt change of bed elevation (e.g. a wall or riverbank), δb may likely become larger than h_i, leading to $h_L = 0$, as shown in Figure 4.3 (b). In this case, the water can only flow downhill from the higher side, effectively representing the real-world situation.

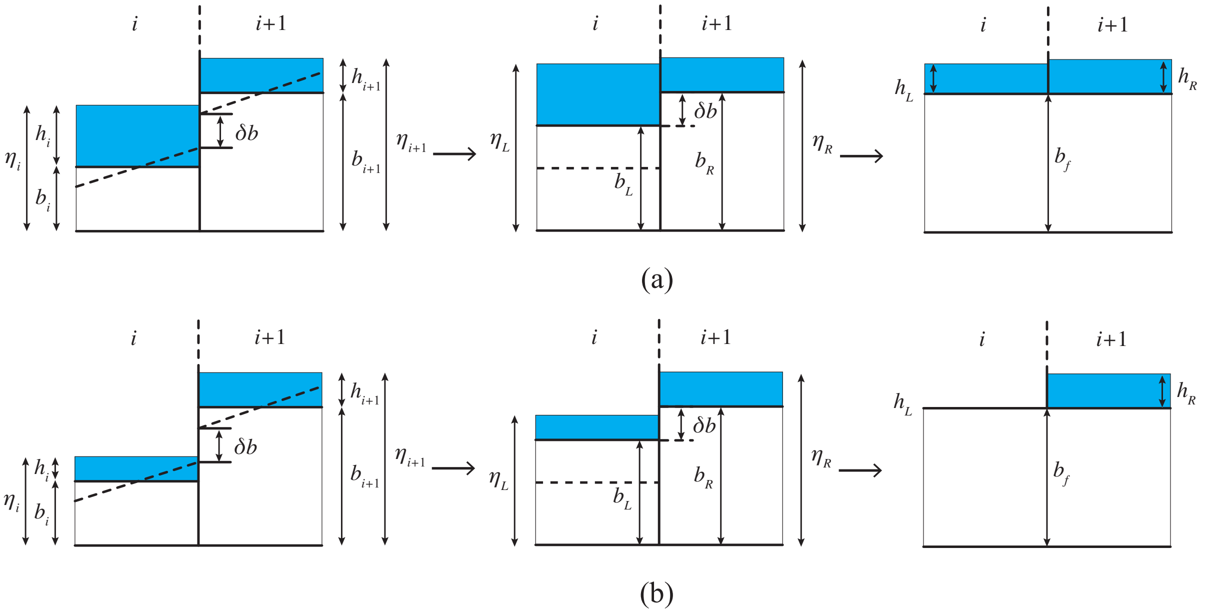


Figure 4.3: SRM implementation when $\delta b > 0$ and $b_{i+1}-b_i-\delta b < \eta_{i+1}-\eta_i$: (a) $0 < \delta b < h_i$; (b) $\delta b > h_i$. The left, middle and right panels respectively illustrate the original discretisation with the dashed line representing the limited slope, the water surface elevation reconstructed according to (4.23) with the dashed line representing the original location of the bed, and the Riemann states of water depth defined following (4.28-4.29).

Taking the x-direction flux $\mathbf{f}$ as an example, the HLLC Riemann solver is given as

$$\mathbf{f} = \begin{cases} \mathbf{f}_L & 0 \leq S_L \\ \mathbf{f}_{*L} & S_L \leq 0 \leq S_M \\ \mathbf{f}_{*R} & S_M \leq 0 \leq S_R \\ \mathbf{f}_R & S_R \leq 0 \end{cases} \tag{4.31}$$

in which $\mathbf{f}_L = \mathbf{f}(\mathbf{q}_L)$ and $\mathbf{f}_R = \mathbf{f}(\mathbf{q}_R)$ are calculated from the left and right Riemann states, S_L, S_R and S_M the are characteristic wave speeds. $\mathbf{f}_{*L}$ and $\mathbf{f}_{*R}$ are the fluxes in the left and right middle regions of the HLLC solution structure, calculated as

$$\mathbf{f}_{*L} = \begin{bmatrix} f_{*1} \\ f_{*2} \\ v_L f_{*1} \end{bmatrix} \qquad \mathbf{f}_{*R} = \begin{bmatrix} f_{*1} \\ f_{*2} \\ v_R f_{*1} \end{bmatrix} \tag{4.32}$$

with the HLL fluxes $\mathbf{f}_*$ provided by the following formula

$$\mathbf{f}_* = \frac{S_R \mathbf{f}_L - S_L \mathbf{f}_R + S_L S_R (\mathbf{q}_R - \mathbf{q}_L)}{S_R - S_L} \tag{4.33}$$

The formulae for the left and right characteristic wave speeds S_L and S_R are

$$S_L = \begin{cases} u_R - 2\sqrt{gh_R} & h_L = 0 \\ \min(u_L - \sqrt{gh_L}, u_* - \sqrt{gh_*}) & h_L > 0 \end{cases} \tag{4.34}$$

$$S_R = \begin{cases} u_L + 2\sqrt{gh_L} & h_R = 0 \\ \max(u_R + \sqrt{gh_R}, u_* + \sqrt{gh_*}) & h_R > 0 \end{cases} \tag{4.35}$$

in which

$$u_* = \frac{1}{2}(u_L + u_R) + \sqrt{gh_L} - \sqrt{gh_R} \tag{4.36}$$

$$h_* = \frac{1}{g}\left[\frac{1}{2}(\sqrt{gh_L} + \sqrt{gh_R}) + \frac{1}{4}(u_L - u_R)\right]^2 \tag{4.37}$$

The middle characteristic wave speed S_M is calculated as

$$S_M = \frac{S_L h_R (u_R - S_R) - S_R h_L (u_L - S_L)}{h_R (u_R - S_R) - h_L (u_L - S_L)} \tag{4.38}$$

Another advantage of SRM is that it does not affect the maximum value between b_i and b_{i+1}. Following the definition of the single face value of bed elevation in (4.27), the genuine bed variation within a cell is effectively preserved during the reconstruction. The bed slope source terms can be simply discretised using the following equation:

$$\mathbf{S}_{bi} = \begin{pmatrix} 0 \\ \frac{1}{\Omega_i}\sum \frac{1}{2}g(h_i + h_{L,k})(b_i - \bar{b}_{f,k})\mathbf{n}_k l_k \end{pmatrix} \tag{4.39}$$

where $h_{L,k}$ is the left Riemann state of water depth at cell edge 'k', and $\bar{b}_{f,k}$ is modified face value of the bed elevation that will be introduced below.

Considering an arbitrary cell 'i', violation of the lake at rest solution and numerical instability may occur during a simulation when the final reconstructed water surface elevation at one of its cell interface is lower than the bed elevation at another cell interface along the same direction. As illustrated in Figure 4.4, this basically reflects the cases when the flow hits a 'wall' or when cell 'i' has an excessive slope caused by abrupt change of bed elevation. In order to preserve the lake at rest solutions in the vicinity of wet-dry interfaces and avoid numerical instability caused by abrupt change of bed elevation, the following local bed modification scheme is proposed and applied to all of the computational cells

$$\bar{b}_f = b_f - \Delta b \tag{4.40}$$

$$\begin{cases} \Delta b = \max(0, b_f - \eta_i) & \text{if } h_{i+1} < \varepsilon_h \\ \Delta b = \max(0, \min(\delta b, b_f - \eta_i)) & \text{if } h_{i+1} >= \varepsilon_h \end{cases} \tag{4.41}$$

where ε_h is an infinitesimal value to define dry cells, which is taken as 10^{-10} in this work. It should be particularly noted that the original water surface elevation η_i at the cell centre rather than the reconstructed value η_L is used in (4.41) to calculate Δb. Without losing generality, it is assumed that the reconstructed water surface elevation at the left hand side cell interface of cell 'i', i.e. '$i - 1/2$', is lower than the bed elevation at the right hand side cell interface '$i + 1/2$'. Figure 4.4 (a) represents the case when the flow hits a 'wall', i.e. when cell 'i' is wet but cell '$i + 1$' is dry with a ground level higher than the water surface. With $\Delta b = b_f - \eta_i$, the local bed modification (4.40-4.41) essentially lowers the bed at cell interface '$i + 1/2$' until it becomes horizontal to the reconstructed water surface at '$i - 1/2$'. This effectively preserves the lake at rest solutions. When both cell 'i' and cell '$i + 1$' are wet, (4.41) gives

$\Delta b = \max(0, \min(\delta b, b_f - \eta_i))$. In a normal condition when the domain topography is smooth or has a constant slope, δb will be small/zero, so does Δb. The proposed local bed modification scheme does not take effects and the computation is undertaken with the bed slope inside cell 'i' correctly represented. Figure 4.4 (b) illustrates a case when bed level at cell '$i+1$' changes abruptly and is much higher than the water surface at cell 'i' (i.e. $\delta b > b_f - \eta_i$), as a result of excessive bed curvature or discontinuous topography. This may lead to numerical instability during a simulation and should be avoided. From (4.41), we have $\Delta b = b_f - \eta_i$ and then from (4.41) $\bar{b}_f = b_f - \Delta b = \eta_i$. This effectively removes the excessive slope in cell 'i' and hence ensures a stable simulation; otherwise, time steps may become prohibitively small due to the large velocities calculated from large slope source terms.

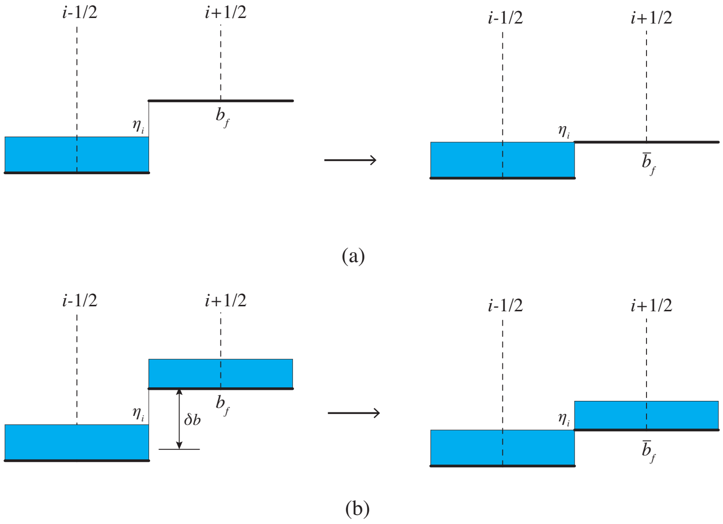


Figure 4.4: Local modification of interface bed elevation after water surface reconstruction. (a) wet-dry cells. (b) wet-wet cells.

The current numerical method, featured with the aforementioned SRM and local bed modification schemes, strictly preserves the lake at rest solution even when the numerical solution involves wetting and drying over irregular bed topography. This may be proved as follows: for a lake at rest problem with a stationary water surface $\bar{\eta}$, we have $\bar{b}_{f,k} = \bar{\eta} - h_{L,k}$; the momentum components of $\mathbf{S}_{bi}$ in (4.39) can be rewritten as $\frac{1}{\Omega_i}\sum[\frac{1}{2}g(h_i b_i - h_i\bar{\eta} + h_{k,L}^2)]\mathbf{n}_k l_k$; the first two terms vanish following simple algebra manipulation and the expression becomes $\frac{1}{\Omega_i}\sum\frac{1}{2}g h_{k,L}^2 \mathbf{n}_k l_k$; this effectively balances the momentum fluxes to preserve the lake at rest solution.

4.4.3 *Friction source term discretisation*

The implicit part of (4.20) is solved directly rather than reformulated into an explicit form as in (4.13) to ensure that the velocities can be relaxed to the correct equilibrium state in a single time step when the friction become dominant. Since the continuity equation does not involve a non-zero friction term, the implicit scheme only applies to the momentum equations. The implicit terms in the x and y-direction momentum equations are interdependent, and an iteration method must be used. Denoting the unit-width discharge $\mathbf{Q}_i^{n+1} = ([uh]_i^{n+1}, [vh]_i^{n+1})^T$, the momentum components of (4.20) may be rewritten as

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n + \Delta t(\mathbf{A}_i^n + \mathbf{S}_i^{n+1}) \tag{4.42}$$

where $\mathbf{A}_i^n = \frac{1}{\Omega_i}\sum_{k=1}^{N}\mathbf{F}_k(\mathbf{q^n})l_k + \mathbf{S}_{bi}^n$ represents the momentum components of the sum of convective fluxes and slope gradient source terms, and $\mathbf{S}_i^{n+1}$ contains the friction source terms for the momentum equations given by

$$\mathbf{S}_i^{n+1} = -gn^2(h_i^n)^{-7/3}\mathbf{Q}_i^{n+1}|\mathbf{Q}_i^{n+1}| \tag{4.43}$$

When the water depth is infinitesimally small, the friction source terms calculated from (4.43) may become excessively large ($\sim 10^{20}$ depending on ε_h), which may exceed the maximum floating number precision of a machine and generates unexpected machine error that prohibits a converged solution. To effectively resolve this (e.g. to rescale the equation to $\sim 10^{10}$ and avoid any unexpected machine error), a simple technique is implemented herein. The necessary auxiliary variables are defined as

$$\mathbf{U}_i^{n+1} = \mathbf{Q}_i^{n+1}/h_i^{n+1}, \mathbf{U}_i^n = \mathbf{Q}_i^n/h_i^{n+1}, \bar{\mathbf{A}}_i^n = \mathbf{A}_i^n/h_i^{n+1} \tag{4.44}$$

$$\bar{\mathbf{S}}_i^{n+1} = -gn^2(h_i^n)^{-4/3}\mathbf{U}_i^{n+1}|\mathbf{U}_i^{n+1}| \tag{4.45}$$

A new time-marching equation for the auxiliary variables can be obtained by dividing both sides of (4.42) by h_i^{n+1}

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n + \Delta t(\bar{\mathbf{A}}_i^n + \bar{\mathbf{S}}_i^{n+1}) \tag{4.46}$$

This is then solved using the Newton-Raphson method and $\mathbf{Q}_i^{n+1}$ can be easily recovered from $\mathbf{U}_i^{n+1}$ using the relationship as defined in (4.44).

The Newton-Raphson iteration procedure implemented for solving (4.46) is given as

$$\mathbf{U}^{p+1} = \mathbf{U}^{p} + [\mathbf{I} - \Delta t \mathbf{J}(\mathbf{U}^{p})]^{-1}[\Delta t \bar{\mathbf{S}}(\mathbf{U}^{p}) + \Delta t \bar{\mathbf{A}}_{i}^{n} - (\mathbf{U}^{p} - \mathbf{U}_{i}^{n})] \tag{4.47}$$

with a convergence criteria defined as

$$|\mathbf{U}^{p+1} - \mathbf{U}^{p}| \leq 0.001|\mathbf{U}^{p}| \tag{4.48}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{J}$ is the Jacobian matrix of $\bar{\mathbf{S}}$, p and $p+1$ denotes the iteration steps and $\mathbf{U}_{i}^{n}$ provides the initial values to start the iterations. The solution procedure usually converges in $2 \sim 3$ steps, or less if the flow is near to a steady state or the friction is small. Numerical experiments show that the iteration procedure requires less than 10% of the runtime needed for calculating interface fluxes. Therefore, the current implicit friction term discretisation method does not affect the computational efficiency of the overall numerical scheme.

It should be noted that no special treatment is needed for wet-dry front except for setting the velocities to zero when water depth is smaller than a small threshold (e.g. 10^{-10} m used in this work).

4.4.4 *Stability criteria*

The proposed fully implicit method for discretising the friction source terms does not impose any constraint on time steps. The stability criterion for the overall finite volume Godunov-type scheme is therefore the CFL condition, which is given as

$$\Delta t = \mathrm{CFL} \min_{i}\left(\frac{d_i}{|\mathbf{u}_i| + \sqrt{g h_i}}\right) \tag{4.49}$$

where d_i is the minimum distance from cell centre to cell edges and the CFL number may take any value between 0 and 1.

4.5 Results and validation

In this section, several test cases are simulated to validate the new overland flow model implemented with the proposed novel SRM and implicit friction term discretisation scheme. Results obtained from a selection of alternative models introduced in section 4.3 are compared to demonstrate the improved capability of the current model. Table 4.1 provides a list of the

Table 4.1: Summary of models used for comparison

Model name	Slope source term	Friction source term
George	George's Riemann solver	New implicit
HR	Hydrostatic reconstruction method	New implicit
SRM	New surface reconstruction method	New implicit
SRMEXP	New surface reconstruction method	Old implicit
SRMSPLIT	New surface reconstruction method	Fractional splitting

models used in the simulations with different combinations of slope source terms and friction source terms discretisation. In Table 4.1, the old implicit scheme refer to the 2D form of the explicit reformulation presented in (4.9-4.10) (e.g. Song et al. [2011b]) and the fractional splitting scheme refer to the implicit method used in Liang and Marche [2009] for which the 1D form is presented in (4.13). In all of the simulations, CFL = 1.0 and $g = 9.81$ m/s^2 are used. The boundary conditions are imposed using the ghost cell method described in Liang [2010].

4.5.1 *1D steady and uniform flow on slopes*

One dimensional steady and uniform flow on inclined slopes is a simple test case. But it can serve as a basic benchmark test to verify whether the slope source term and friction source term are correctly discretised and computed. In this test case, water depth $h = 0.01$ m, the Manning's coefficient $n = 0.033$ sm$^{-1/3}$ and five different slopes (0.02, 0.04, 0.06, 0.08 and 0.1) are used. Simulations are carried out at the same spatial resolution ($\Delta x = 10$ m) using models with different schemes for slope term discretisation. As the slope increases, the steady uniform flow changes from subcritical to supercritical. The steady state velocities computed by different models are plotted against the analytical solution in Figure 4.5. Both of the 'SRM' and 'George' models predict correct steady state velocities while the 'HR' model fails to give correct results for the flows over all five slopes. The velocities predicted by 'HR' do not vary with the change of the slopes, indicating that the slope gradient approximated in 'HR' is effectively bounded. The mass flux at a sample cell interface, i.e. discharge, computed by different models are also compared in Figure 4.5. It should be noted that the discharge here is not necessarily the same as hu defined at the cell centres. Only the 'SRM' model produces correct interface mass flux for all five slopes. The 'HR' model completely fails and predicts constant discharge for all five simulations. The predicted discharge is larger than the analytical discharge at one simulation in the subcritical region but smaller than the correct solution in all other four simulations including one for subcritical flow. This is clearly caused by the combination of the constrained slope gradient and the 'waterfall effect' as implemented in the numerical scheme.

George's augmented Riemann solver overestimates the discharge in the two simulations in the subcritical region, which is consistent with the analysis provided in section 4.3.

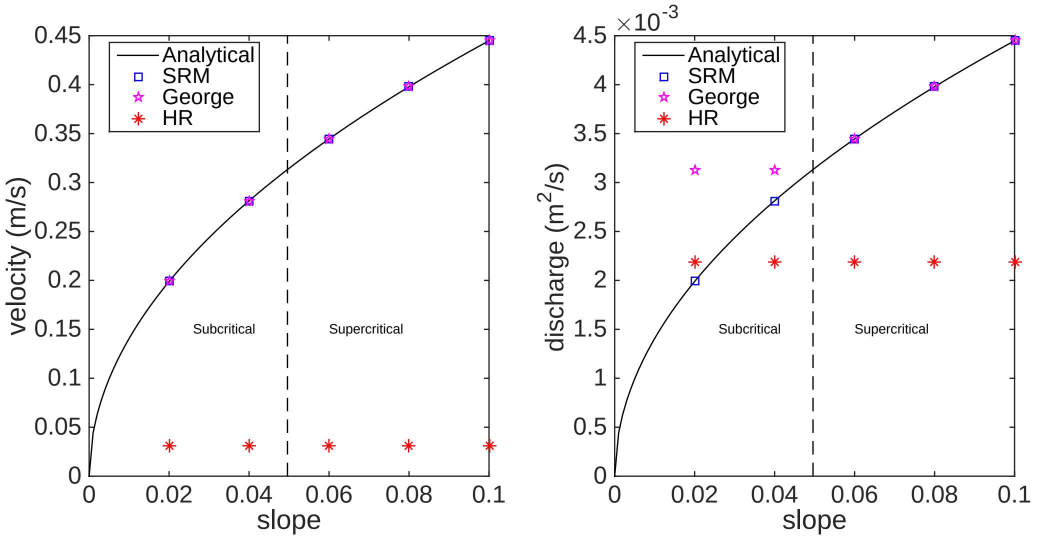


Figure 4.5: 1D steady and uniform flow on slopes: computed steady state velocities and discharges on 5 different slopes. Left: Velocity; Right: Discharge.

Further simulations are also carried out using the models with different discretisation schemes for the friction source terms to test their capability in terms of relaxing the flow to the equilibrium state. While the time steps predicted by the CFL condition increases with the cell size, the relaxation time towards equilibrium does not depend on the cell size. Therefore it can be expected that the flow is more likely to be relaxed to the equilibrium state in a single time step when using coarser grids. Therefore, the simulations are performed at three spatial resolution ($\Delta x = 1$ m, $\Delta x = 10$ m, $\Delta x = 100$ m) and the same slope (0.1). The water is set as motionless in the beginning. The velocities are normalised by the steady state velocity and plotted against the time step count in Figure 4.6. Clearly, the 'SRMSPLIT' model is not able to relax to the correct equilibrium state at all three spatial resolutions and the deviation increases as the resolution decreases. 'SRMEXP' can relax the flow to the correct equilibrium state but strong oscillations are observed for coarse grids ($\Delta x = 10$ m and $\Delta x = 100$ m). Furthermore, excessively large velocities are at the first time step. The reason is that these two schemes are essentially explicit, which effectively results in zero friction at the first time step. On the contrary, the 'SRM' model with the new friction discretisation scheme is able to quickly and monotonically relax the flow to the correct equilibrium state, achieved in only a single time step even when a coarse enough grid is used ($\Delta x = 100$ m).

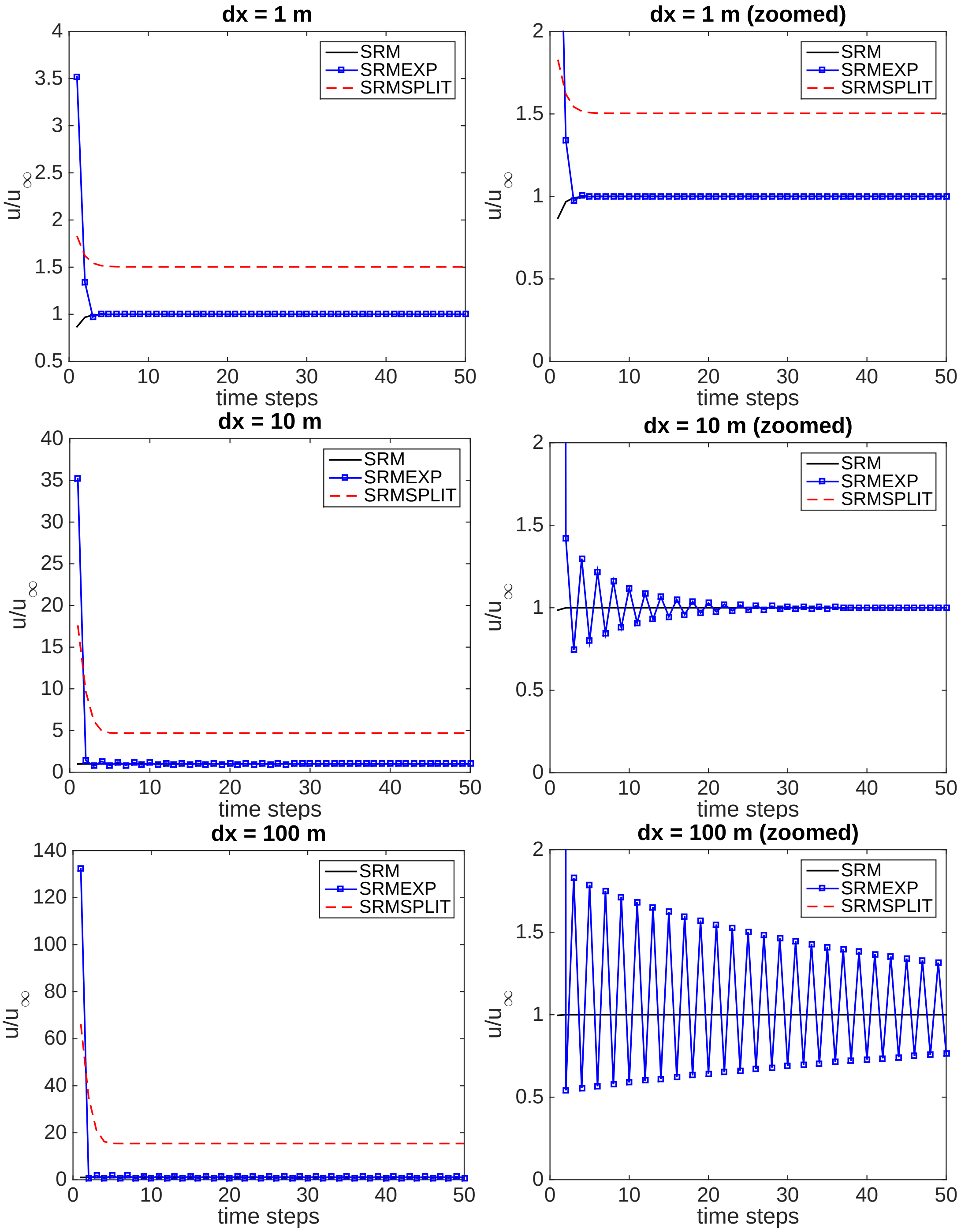


Figure 4.6: 1D steady and uniform flow on slopes: relaxation towards the equilibrium state with three different spatial resolutions. Left column: normal view. Right column: zoomed view for clear illustration of SRM and SRMEXP.

4.5.2 *Rainfall on a V-shaped catchment*

As shown in Figure 4.7, the V-shaped catchment consists of two hillsides with a 0.05 slope and, in between, a channel with a 0.02 slope. Constant and uniform rainfall with an intensity of 10.8 mm/h falls on the whole catchment for 1.5 hours. The Manning coefficient is set to 0.015 sm$^{-1/3}$ on the hillsides and 0.15 sm$^{-1/3}$ on the channel. Compared with the 1D steady and uniform flow on slopes, this test case is more realistic in the sense that the V-shaped catchment can be interpreted as an idealisation of real world catchments. During the simulations, the catchment is discretised using a grid of 10 m resolution. All of the domain boundaries are closed except for the channel outlet where open boundary conditions are imposed.

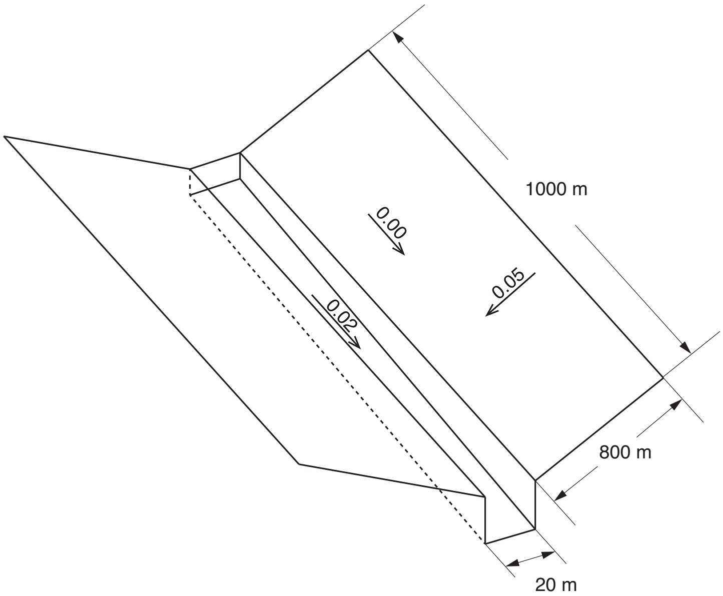
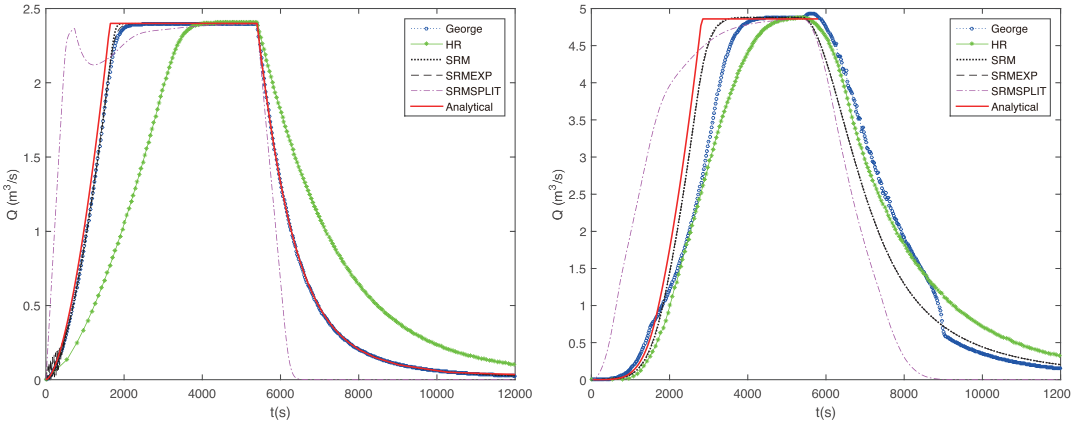


Figure 4.7: Rainfall on a V-shaped catchment: geometry of the catchment.

Figure 4.8 plots the hydrographs at the hillsides and channel outlet predicted by the three different models, comparing with the analytical solutions derived based on the kinematic wave assumption Di Giammarco et al. [1996]. The time histories of water depth at the middle of hillside end and channel outlet are compared in Figure 4.10. Closest agreement with the analytical solutions is achieved by the 'SRM' model. The hydrographs predicted by 'HR' has slower rising limbs, which implies that the discharge has been underestimated. The maximum water depth computed by the model 'HR' model is about 0.01 m, which is much smaller than the bed elevation change across two cells (0.5 m). Therefore we can conclude that the error is caused by the incorrect approximation of the slope gradient. George's Riemann solver gives a correct hillside hydrograph because the flow is on the hillside supercritical and George's scheme does not lead to overestimated discharge in supercritical flow conditions. But clear simulation error and small unphysical oscillations can be found in the hydrograph at the channel outlet.

Comparing the friction discretisation schemes, 'SRMSPLIT' gives the worst results. 'SRMEXP' produces better results; however, unphysical oscillations can be found at the initial stage of the hydrograph at the hillside and larger deviation from the analytical solution is predicted at the rising limb close to the steady phase, as shown in Figure 4.9. The different results predicted by the models implemented with different friction discretisation schemes are consistent with the analysis as provided in section 4.3 and the results for the 1D steady uniform flow test in the previous section. The results clearly confirm that the new friction discretisation scheme is indeed essential for accurate and stable simulation of overland flows with reasonable large time steps (CFL = 1.0).



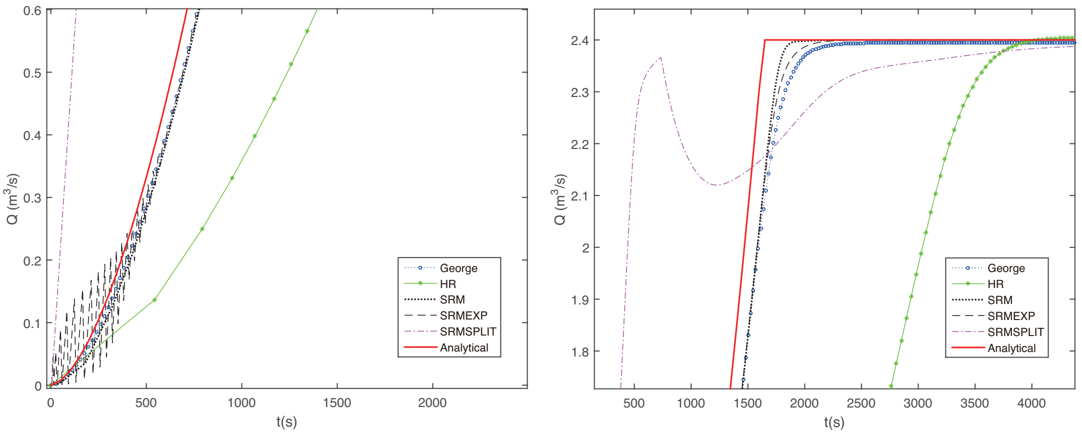
(a)

(b)

Figure 4.8: Rainfall on a V-shaped catchment: comparison between the kinematic wave analytical solutions and simulated hydrographs at the: (a) hillside; (b) channel outlet. Note that the kinematic wave solution for the falling limb of the channel outlet hydrograph is not available.

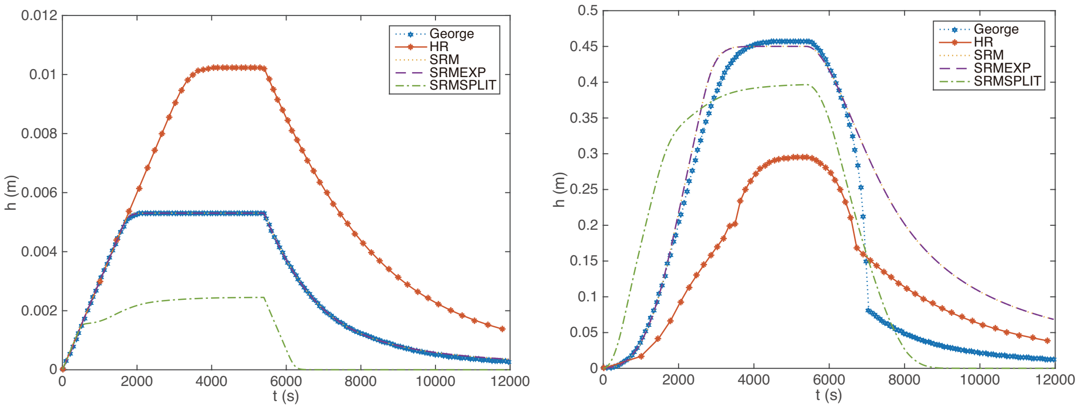
4.5.3 *Still water test*

In this test case, a quiescent steady flow over uneven topography (Hou et al. [2013a]) with wet-dry interface to demonstrate that the new 'SRM' model is able to preserve the lake at rest solution exactly. The 8000 m × 8000 m domain is discretised into 160 × 160 cells. The topography featured with two overlapped bumps is defined as $b(x, y) = \max(0, B_1, B_2)$ with $B_1 = 2000 - 0.00032[(x-3000)^2 + (y-5000)^2]$ and $B_2 = 900 - 0.000144[(x-5000)^2 + (y-3000)^2]$. The initial water level is 1000 m. The results at $t = 5000$ s is illustrated in Figure 4.11 and clearly demonstrate that the lake at rest solution is exactly preserved by the new model.



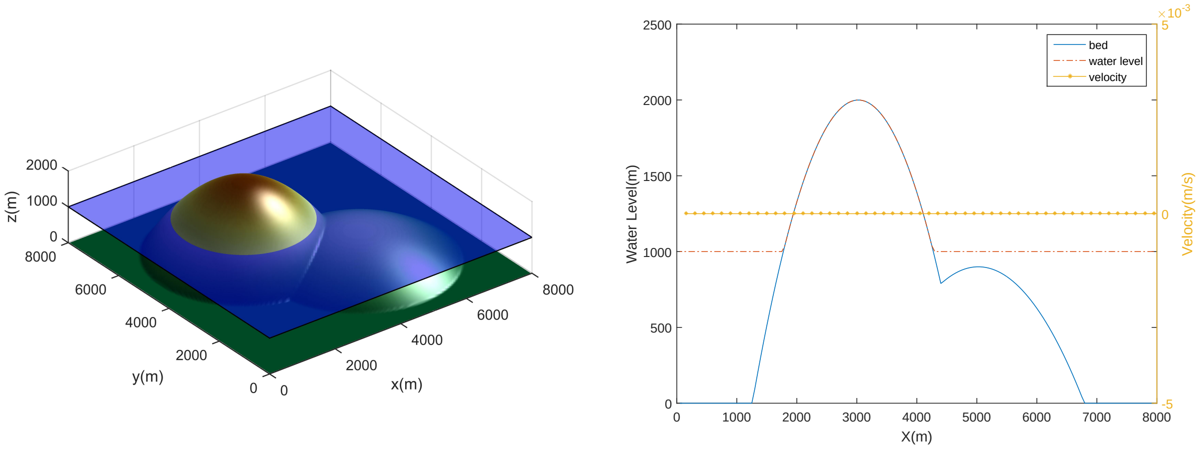
(a) (b)

Figure 4.9: Rainfall on a V-shaped catchment: zoomed view of the discharge at the hill side. (a) initial stage; (b) transition between increasing phase and steady phase.



(a) (b)

Figure 4.10: Rainfall on a V-shaped catchment: comparison of time histories of water depth at: (a) middle of the hillside end; (b) channel outlet.



(a) (b)

Figure 4.11: Results for still water test after $t = 5000$ s. (a) 3D view; (b) Water level and velocity at the diagonal cross section.

4.5.4 *Rainfall-runoff experiment in a simplified urban area*

The rainfall-runoff experiment carried out by Cea et al. [2010] is considered to further validate the current SWEs model for overland flow simulation in urban areas. The experimental urban catchment is represented by a steel basin decorated with buildings made from wood blocks, with the topography and the layout of buildings shown in Figure 4.12. The basin features a valley in the middle with the bottom edge designed to be the outlet. The horizontal dimensions of the buildings are 20 cm × 30 cm and the vertical outer walls are 20 cm in height. All of the buildings also have a roof with two sloping sides from the centre to the edge. Rainfall events with a constant intensity of 300 mm/h but three different durations, i.e. 20 s, 40 s and 60 s, is imposed on the basin. Following Cea et al. [2010], the three rainfall events are referred to as Q25T20, Q25T40 and Q25T60.

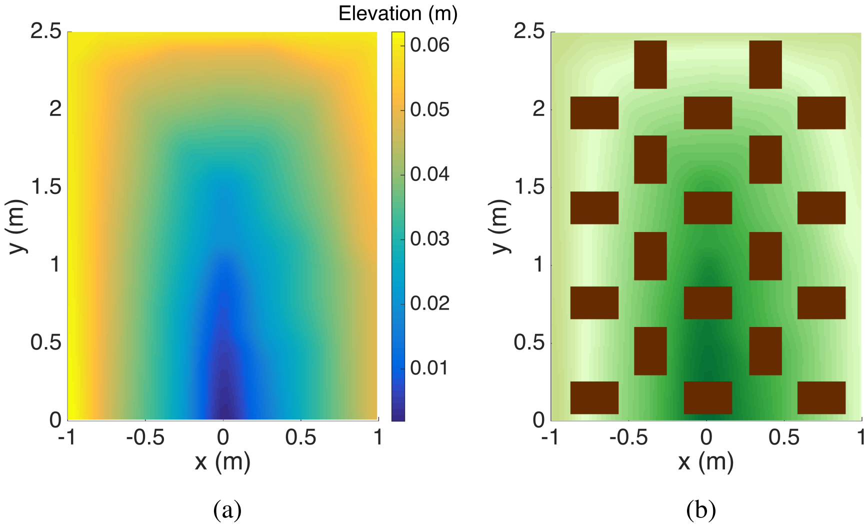


Figure 4.12: Rainfall-runoff experiment: (a) domain topography; (b) buildings layout.

During the simulation, the building blocks are represented in the model as part of the topography without any further special treatment. The Manning coefficient is set to 0.016 sm$^{-1/3}$ over the entire domain as suggested by Cea et al. [2010]. The uniform computational grid consists of 50000 cells with a 0.01 m resolution. Zero flow depth is assumed in the boundary ghost cells at the bottom edge of the domain to impose the outlet boundary conditions while the other three edges are assumed to be closed.

Figure 4.13 presents the 'SRM' predicted water depths and velocities induced by the three rainfall events at the time when the rainfall stops. It is evident that rainfall-runoff process and the resulting flow patterns are closely related to domain topography. The rainwater flows from sloping roofs of the buildings to the ground, which is then directed to valley at the centre of the basin and travels towards the bottom outlet. Although the water depths on the building roofs are very small, the velocities are significant and diverge from the central line of the roofs due to the slopes, indicating correct representation of the slope gradients and runoff with small water depth. Because the proposed numerical scheme effectively handles excessive slopes, no unrealistically large velocities are predicted at the edges of the buildings where abrupt change of the topography occurs. Figure 4.14 compares the simulated discharges produced by all three different models with the experiment measurements at the bottom outlet for the three rainfall events. Excellent agreement is achieved by 'SRM' for all three cases, confirming the model's superior simulation accuracy. The 'HR' model also gives good results because the resolution is high for this small-scale test case and the inadequacy of the hydrostatic reconstruction method in handling small water depth is therefore not pronounced. The peak discharge is slightly overestimated, which may be due to the fact that the flow in valley centre is mostly subcritical, as illustrated in Figure 4.15 (b), and the George's Riemann solver typically overestimates discharge on slopes for subcritical flows as discussed in section 4.3.

In this test, the entire domain is wet due to uniform rainfall events. Because George's Riemann solver does not involve any special treatment to enhance numerical stability, the predicted velocities near to building walls where abrupt change of bed elevation occurs are significantly higher (5 - 10 times) than those in other areas with smooth topography. These 'big' velocities effectively control the model stability, leading to the use of much smaller and unstable time steps, as shown in Figure 4.16. This inevitably affects the numerical stability and computational efficiency of the model. This test case therefore also demonstrates the importance of implementing an appropriate technique (e.g. the SRM proposed in this work) to automatically constrain bed slope gradients over the entire whole domain.

4.5.5 *Laboratory-scale flash flood onto a simplified urban district*

This urban flash flood experiment was carried out by Testa et al. [2007] as part of the joint European IMPACT (Investigation of extreMe flood Processes And unCerTainty) project. The physical model is a 1:100 representation of the Toce river valley built in the CESI facility in Milan, Italy. Cubic concrete blocks with a side length of 15 cm were placed in a staggered layout to represent buildings, as shown in Figure 4.17 (a). The inflow hydrograph (total discharge at the inlet), as shown in Figure 4.17(b), was controlled by a head tank located at the left-hand

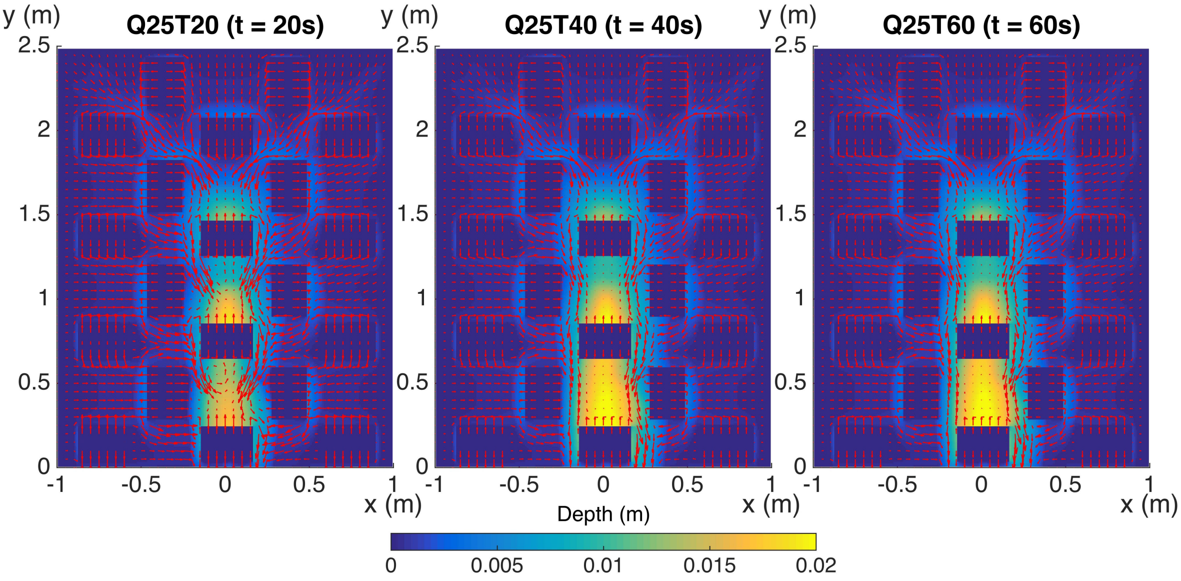


Figure 4.13: Rainfall-runoff experiment: 'SRM' predicted water depths and velocities at the end time of rainfall for the three rainfall events.

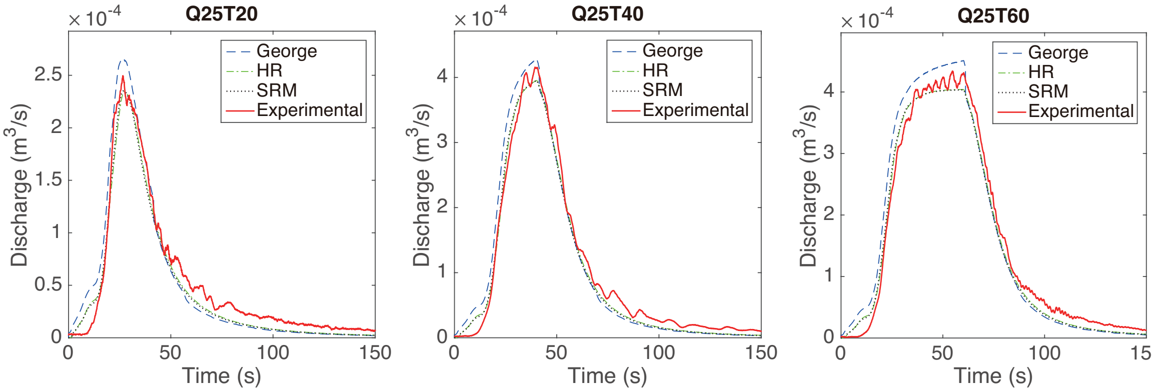


Figure 4.14: Rainfall-runoff experiment: comparison between the numerical and experimental hydrographs at the domain outlet for the three rainfall events.

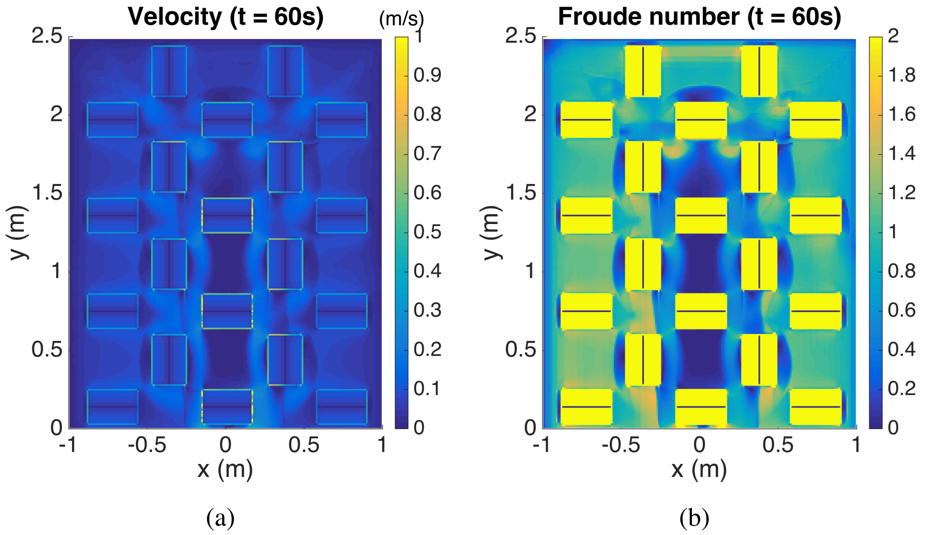


Figure 4.15: Rainfall-runoff experiment: simulation results predicted by the model implemented with the George's Riemann solver for rainfall event Q25T60 at t = 60 s. (a) velocity. (b) Froude number.

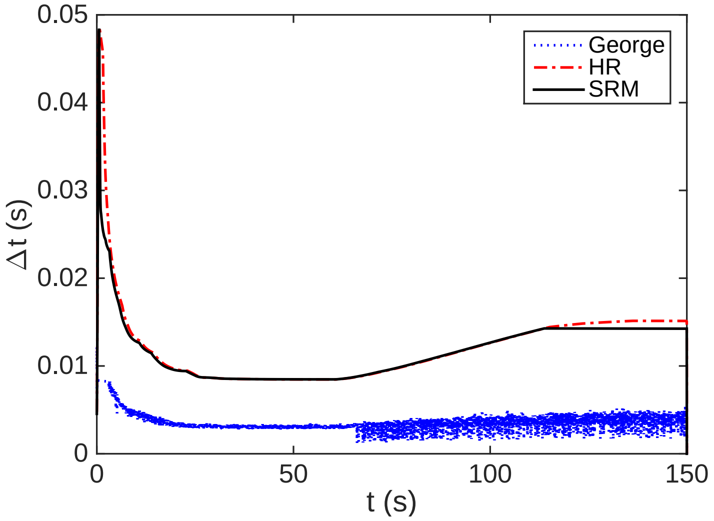
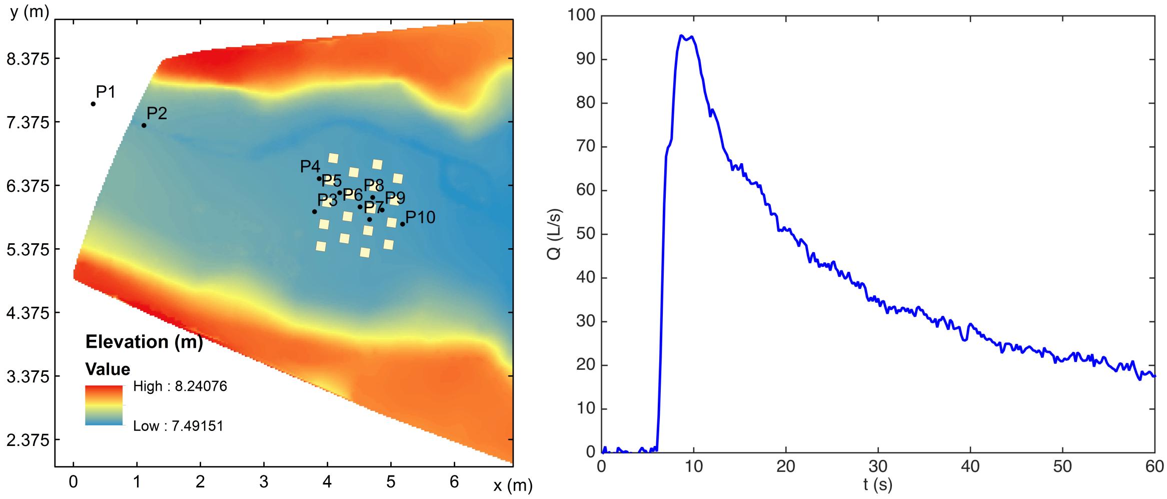


Figure 4.16: Rainfall-runoff experiment: temporal change of time steps produced by the three different models for the Q25T60 simulations.

side of the valley (Testa et al. [2007]), which was referred to as the ‘low’ discharge in Testa et al. [2007]. The water depth was continuously measured at 10 gauge points during the 60 s of the experiment; the locations of gauges are also illustrated in Figure 4.17 (a). Open boundary conditions are imposed at the downstream (right) end of the valley.



(a)

(b)

Figure 4.17: Flash flood onto an urban area: (a) layout of the physical model; (b) inflow hydrograph.

The Manning coefficient is set to 0.0162 sm$^{-1/3}$ as suggested by Testa et al. [2007]. A uniform computational grid at a 0.025 m × 0.025 m resolution is used for the simulation. The buildings are represented in the models by simply raising the bed elevation 0.15 m above the ground. Figure 4.18 presents the ‘SRM’ predicted water depths and velocities different output times, which clearly shows the model’s capability in capturing the transient features of the flash flood. The velocity of the flood front is very high before it reaches the buildings; the flow is then slowed down by the buildings and a reflected shock is developed. Meanwhile, a significant amount of floodwater bypasses the built area through the two sides of the valley. Flood depth starts to decrease after the peak flow has passed. Figure 4.19 further verifies the current simulation results by comparing the predicted time histories of water depth with the measured data and those predicted by alternative models (‘George’ and ‘HR’). Overall, the current numerical results agree well with the measurements and correctly predict both of the arriving time and peak water depth except for Gauges P6 and P8, where the numerical simulation respectively underestimates the flood peak by 1 cm and 2 cm, respectively. This may be attributed to the three-dimensional flow phenomena associated with this highly transient and turbulent flow processes that are not able to be captured by the shallow water equations. This test case does not typically involve those challenging perspectives for overland flow simulations, and so that most Godunov-type schemes can produce satisfactory results. The results predicted by 'SRM' are similar to the alternative simulations produced by the other two models. This test case demonstrates that the new model can successfully simulate those highly transient flows with wetting and drying.

4.5.6 *Hypothetic rainfall event on the Haltwhistle Burn catchment*

The Haltwhistle Burn catchment, covering an area of about 42km^2, is a small catchment in Northumberland, England. It is one of the Rapid Response Catchments recognised by the UK Environment Agency. The topographic map of the catchment is shown in Figure 4.20.

In order to verify the new model's capability in simulating overland flows over real world topography, a hypothetic rainfall event is assumed and the resulting rainfall-runoff and overland flow processes are simulated using the new SRM model. Constant and uniform rainfall with an intensity of 45 mm/h is assumed to fall on the whole catchment for 2 hours. The simulation is carried out for a total of 10 hours starting from beginning of the rainfall. Most of the catchment is covered by pastures and so a constant Manning coefficient of 0.05 sm$^{-1/3}$ is used. Zero infiltration is assumed and open boundary conditions are imposed. The rectangular computational domain embracing the catchment is discretised by a uniform grid at a 5 m resolution, containing 4 million cells. During the simulation, water level is monitored at four gauges located along the Haltwhistle Burn.

The water depth simulated by the new 'SRM' model is shown in Figure 4.21. The rainfall induced overland flow converges into the streams, river channels and lower parts of the catchment, which is as expected. Figure 4.22 presents the time histories of water depth recorded at the four gauges. In order to compare numerical results, simulations are also carried out using alternative models (i.e. 'George' and 'HR'). From the results, it is observed that 'George' and 'HR' perform more consistently than the SRM model and both predict earlier arrival of the flood peaks. To better explain this, the Froude number of the flow predicted the 'George' model is plotted in Figure 4.23 for $t = 2$ h. The Froude number is relatively small in the almost the entire domain and the flow is predominantly subcritical. It has been concluded that George's Riemann solver tends to overestimate the discharge on slopes when the flow is subcritical and this obviously explains the earlier arrival of flood peaks. The early arrival of flood peaks predicted by the 'HR' model may be explained by the overestimated discharge caused by the 'waterfall effect' in this case. The cell size used in the simulation is 5 m and the water

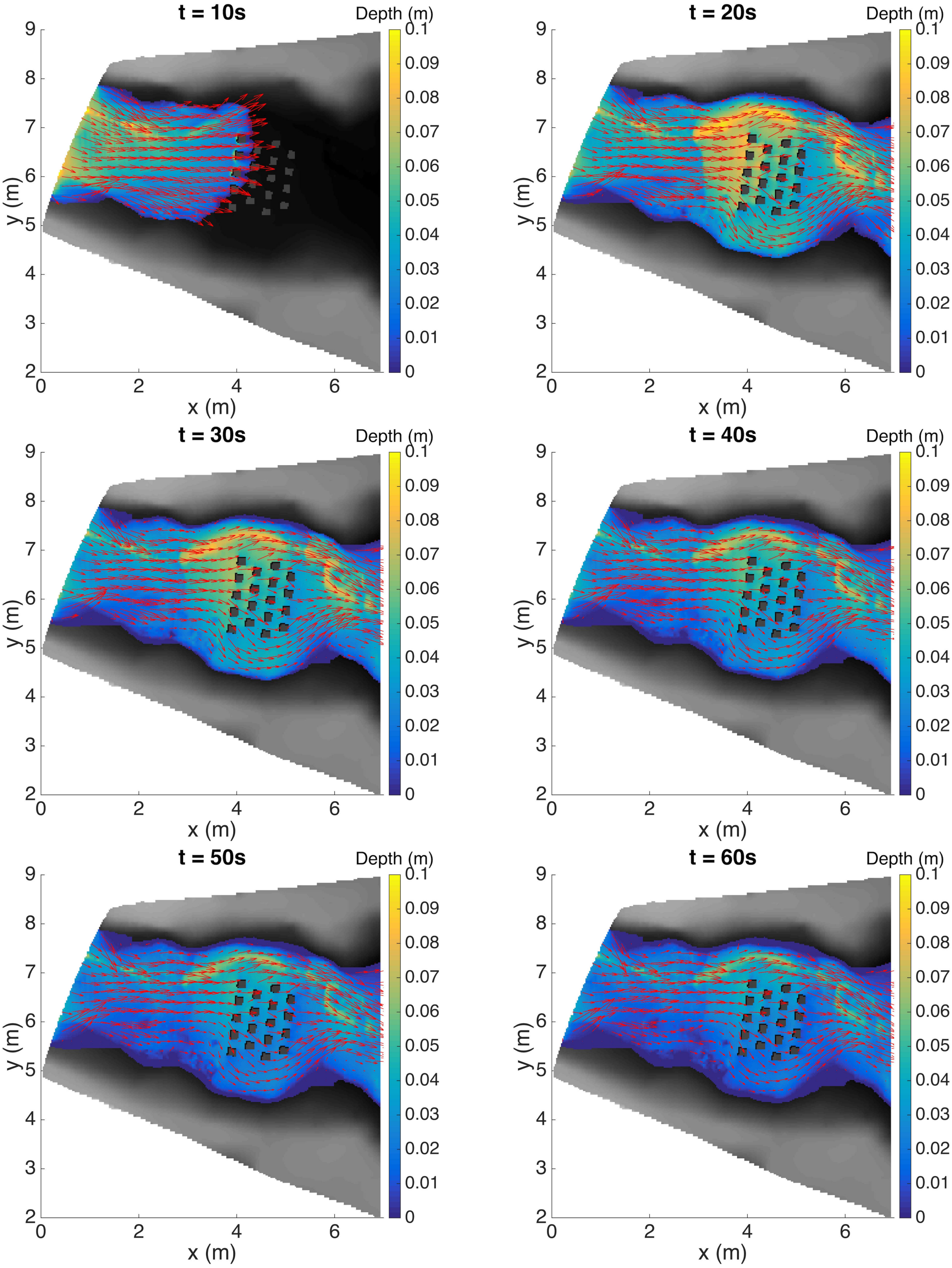


Figure 4.18: Flash flood onto an urban area: simulated water depths and velocities.

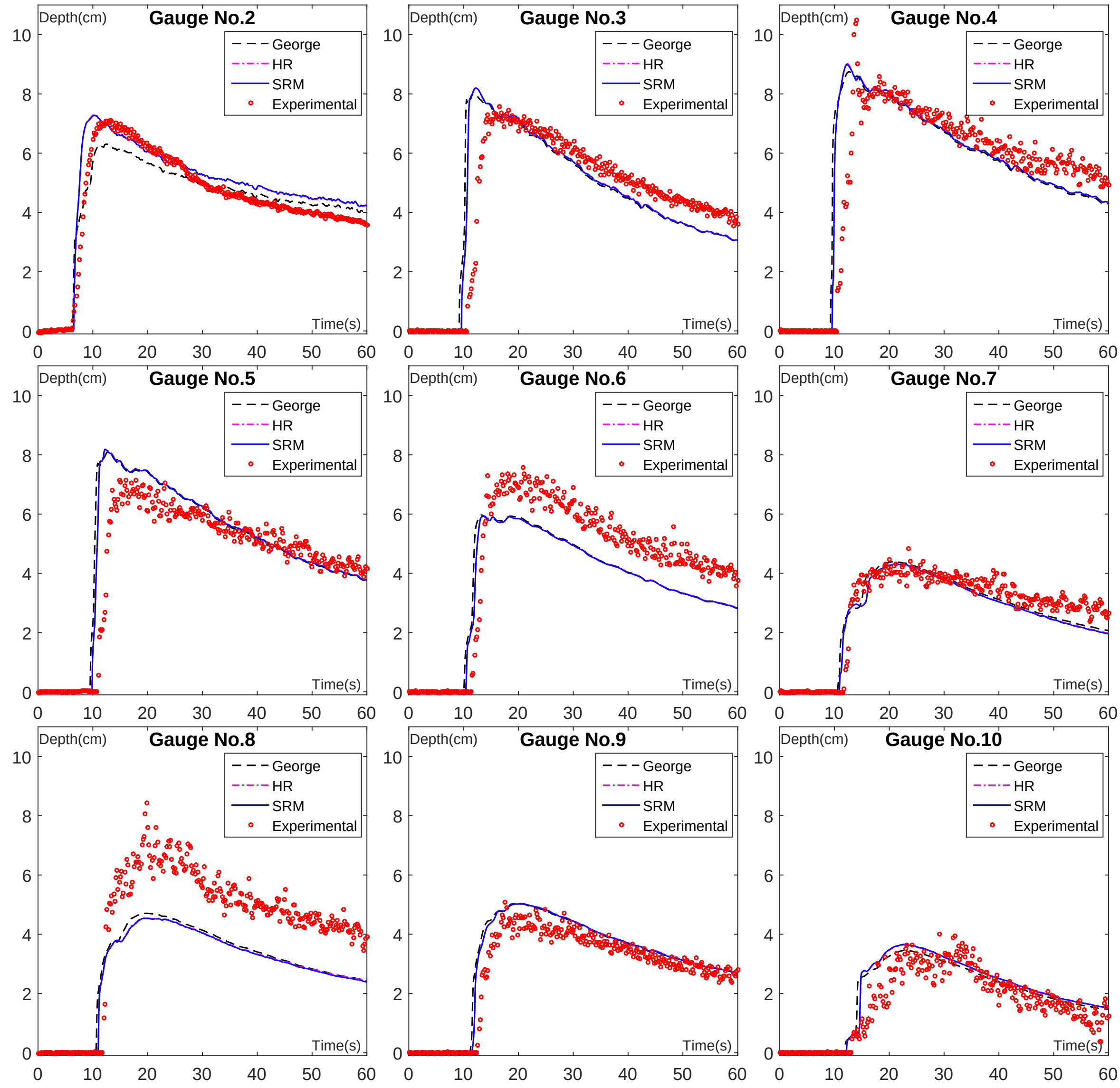


Figure 4.19: Flash flood onto an urban area: time histories of water depth at nine sample gauges.

depth on the slopes is typically at an order of 0.01 m. As a result, even a gentle slope can lead to variation of bed elevation between neighbouring cells larger than the water depth, effectively causing the ‘waterfall effect’. The ‘SRM’ model as presented does not suffer from either of these limitations and hence predicts a late arrival of the flood peaks.

In order to compare their computational efficiency, the three different models are implemented in the same GPU-accelerated codebase and simulations are all carried out on a NVIDIA Tesla K40 GPU. Table 4.2 provides the quantitative comparison of the time step count and runtime required by the three models. The three different simulations need similar number of time steps to finish the whole simulation. Unlike the urban rainfall experiment test previously considered, the ‘George’ model does not require much more time steps than the other two models

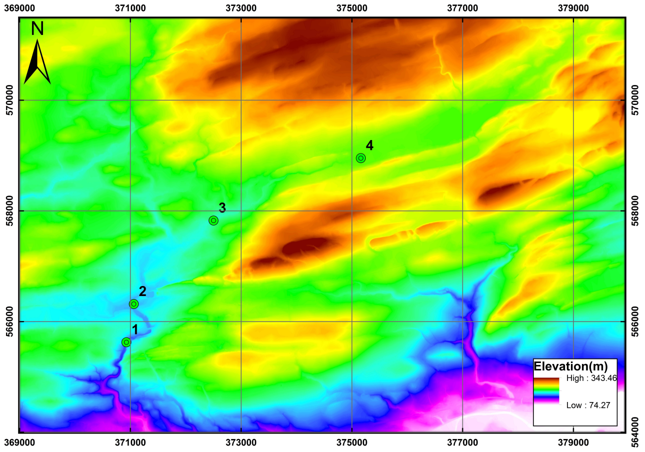


Figure 4.20: The topographic map of the Haltwhistle Burn catchment and the locations of the gauges.

because the time steps are more controlled by the largest water depth in the river channels rather than the largest velocity for this specific case. In terms of run time, the new 'SRM' model consumes 199 minutes to finish the 10-hour simulation, about 40% more than the 137 minutes as required by the 'HR' model. The computational efficiency between the two models is therefore comparable and the new SRM implementation does not necessarily lead to substantial increase of computational cost. On the other hand, the 'George' model is found to be nearly three times slower than the current SRM model due to the complexity of the Riemann solver.

Table 4.2: Hypothetic rainfall event on the Haltwhistle Burn catchment: time steps count and total run time for three simulations.

Model name	Time steps count	Total run time (minutes)
George	154744	595
HR	151034	137
SRM	151089	199

4.6 Conclusion

This chapter discusses in detail the existing challenges encountered in overland flow simulation, proposes new discretisation schemes to deal with these challenges, and subsequently implement these new schemes to develop a new hydrodynamic overland flow model with much improved computational efficiency and numerical stability. The new model has the following key features:

1. Through implementation of a novel surface reconstruction method (SRM) proposed in this work, the current hydrodynamic overland flow model is able to correctly compute

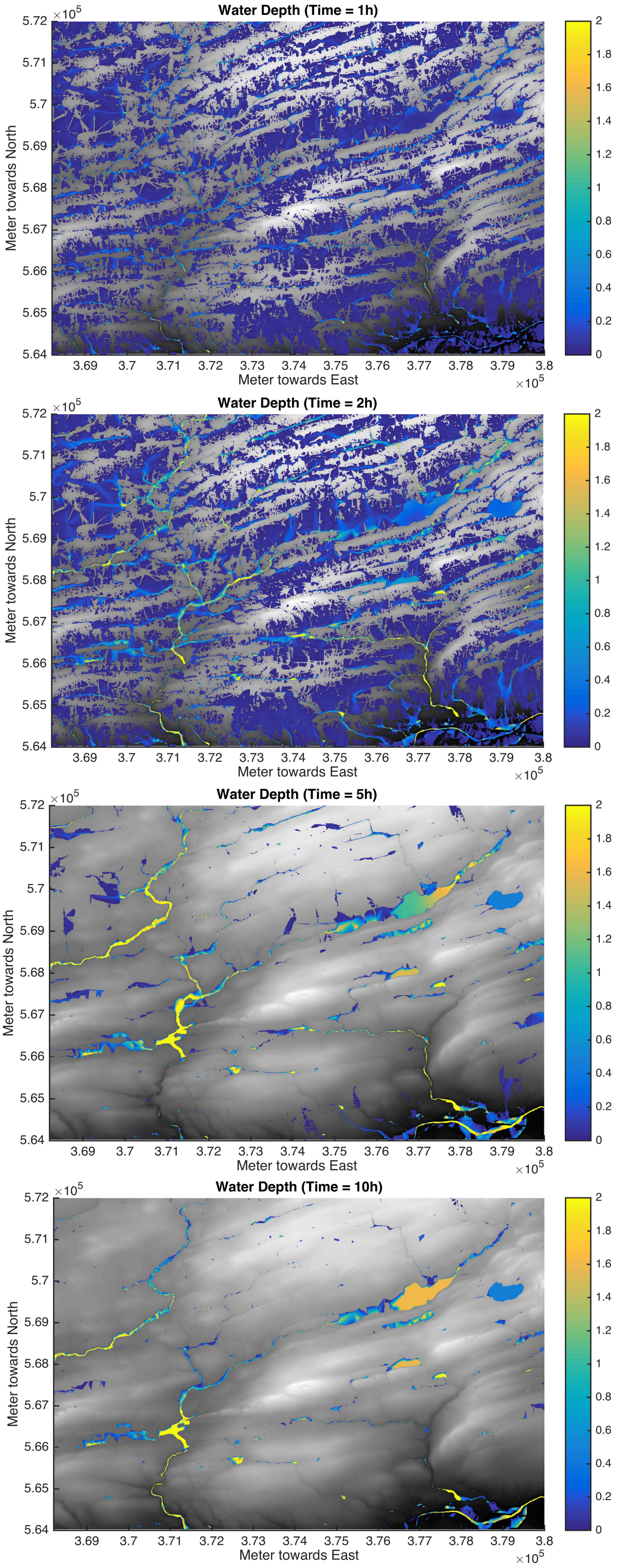


Figure 4.21: Hypothetic rainfall event on the Haltwhistle Burn catchment: water depth (in metres) predicted by the new model at t = 1 hours, t = 2 hour, t = 5 hours and t = 10 hours.

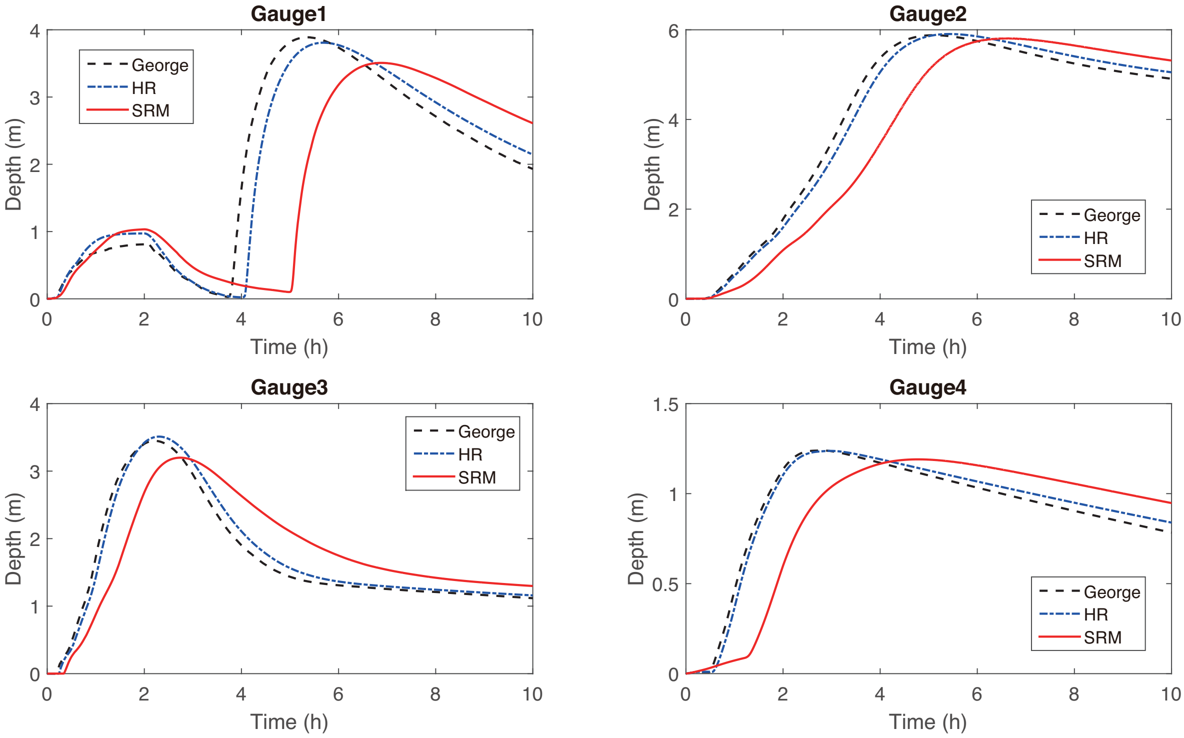


Figure 4.22: Hypothetic rainfall event on the Haltwhistle Burn catchment: time histories of water depth at the four gauges.

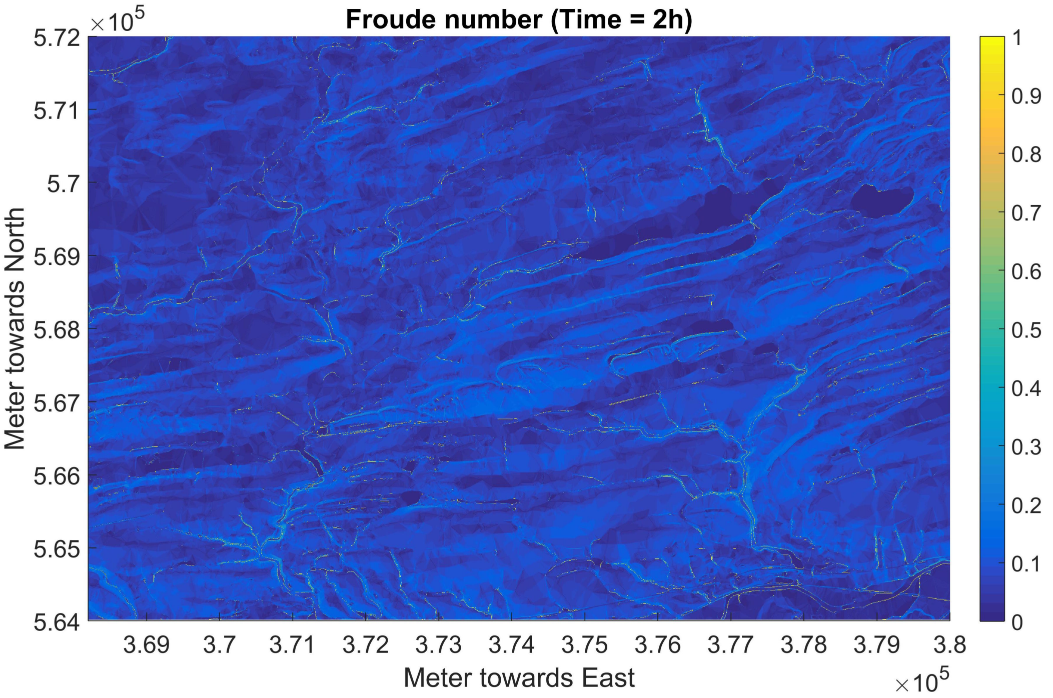


Figure 4.23: Haltwhistle hypothetic flood: Froude number predicted by the model with George's Riemann solver at t = 2 hours.

the bed slope source terms in the limit of disappearing water depth, maintain numerical stability for simulations over rough terrains with abrupt change of bed profiles, and preserve the lake at rest solution with wetting and drying.

2. An implicit scheme is proposed to discretise the strongly nonlinear friction source terms, allowing correct recovery of the theoretical equilibrium state when water depth is small and friction becomes dominant. This new implicit friction discretisation scheme allows accurate and stable simulations using ‘normal’ time steps controlled by the CFL condition, effectively improving computational efficiency.

3. When incorporating with a classic Riemann solver that solves the homogeneous SWEs (e.g. the HLLC approximate Riemann solver as adopted in this work), the new model is able to give correct prediction of mass flux, i.e. unit discharge, on slopes in the limit of disappearing water depth.

These features are essential for accurate, stable and efficient simulation of overland flows. Theoretical analysis and numerical simulations show that the prevailing shallow flow models do not normally implement with all these features simultaneously. The new overland flow model developed in this work resolves all the aforementioned numerical challenges and has all of the above positive features in a single code base. The new model is validated against several test cases and produced numerical results that agree closely with analytical solutions or experimental measurements. The new model also clearly outperforms, in terms of numerical accuracy, stability and efficiency, several other alternative models implemented with different source term discretisation schemes as summarised in Table 4.1.

The numerical technologies featured by the model, i.e. the SRM and the implicit discretisation of the friction source terms will be used for developing the proposed integrated multi-hazard modelling system. As a standalone tool, the model developed in this work can be used as a reliable tool for catchment-scale overland flow simulations.

Chapter 5. A new non-hydrostatic depth-averaged model for flow-like landslides over complex terrain

5.1 Introduction

To facilitate the development of a unified mathematical framework for simulating both water and mass flow, this chapter presents a new depth-averaged model based on a global Cartesian coordinate system with the following highlights:

1. New governing equations have been derived through depth-integration and asymptotic analysis, taking into account the non-hydrostatic vertical pressure and centrifugal force; the resulting equations are hyperbolic and rotational invariant, and mathematically preserve the lake at rest solution.

2. A modified hydrostatic reconstruction method is proposed to numerically handle wetting and drying and preserve the lake at rest solution in the context of a second-order Godunov-type finite volume scheme.

3. A new splitting method is proposed to discretise the friction source terms.

5.2 Governing equations

5.2.1 *Depth-averaging*

On a rectangular Cartesian coordinate system with the vertical axis parallel to the direction of gravity, the three-dimensional governing equations for the single-phase granular flows are given by

$$\nabla \cdot \mathbf{u} = 0 \tag{5.1}$$

$$\rho\{\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u})\} = \nabla \cdot \mathbf{T} + \rho \mathbf{g} \tag{5.2}$$

where ∇ is the gradient operator, $\mathbf{u} = (u, v, w)^T$ defines the velocity field, ρ is the bulk density, t denotes the time, $\otimes$ is the dyadic product, $\mathbf{T}$ is the Cauchy stress tensor, $\mathbf{g} = (0, 0, -g)^T$ gives the gravity field. According to Savage and Hutter [1989], the moving mass may be assumed to satisfy an internal Mohr-Coulomb frictional rheology. The Cauchy stress tensor may be decomposed into an isentropic pressure p and a deviatoric stress $\boldsymbol{\sigma}$:

$$\mathbf{T} = -p\mathbf{I} + \boldsymbol{\sigma} \tag{5.3}$$

where $\mathbf{I}$ is the identity tensor and $\boldsymbol{\sigma} = (\sigma_{xx}, \sigma_{xy}, \sigma_{xz}, \sigma_{yy}, \sigma_{yz}, \sigma_{zz})$ contains stress components in different directions.

For the above governing equations (5.1-5.2), the kinematic boundary conditions are applied at the free surface and the bed as

$$w^s = \frac{\partial s}{\partial t} + u^s \frac{\partial s}{\partial x} + v^s \frac{\partial s}{\partial y} \text{ and } w^b = \frac{\partial b}{\partial t} + u^b \frac{\partial b}{\partial x} + v^b \frac{\partial b}{\partial y} \tag{5.4}$$

where s and b respectively denote the free surface and bed elevations, as illustrated in Figure 5.1.

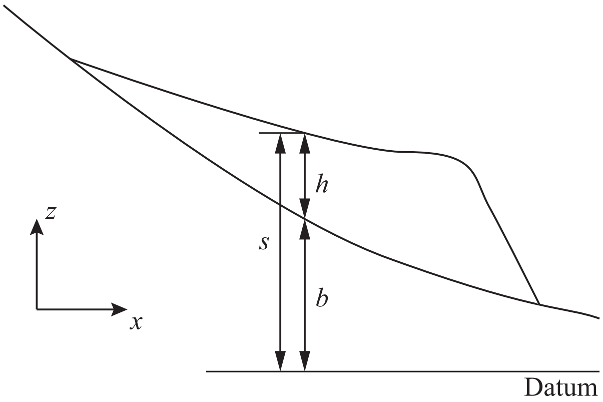


Figure 5.1: The depth-related variables used in the new depth-integration model.

To compare the magnitude of different terms and make appropriate approximations, the above governing equations will be non-dimensionalised through the introduction of the following dimensionless variables:

$$(x, y, z, b, s, h) = L(\hat{x}, \hat{y}, \hat{z}, \hat{b}, \hat{s}, \varepsilon\hat{h}) \text{ , } t = L/\sqrt{gH}\hat{t} \tag{5.5}$$

$$(u, v, w) = \sqrt{gH}(\hat{u}, \hat{v}, \hat{w}) \tag{5.6}$$

in which (x, y, z) define the the three-dimensional Cartesian coordinates, h is the depth of the mass flow ($s = h + b$, as illustrated in Figure 5.1), L and H are the characteristic length and thickness, and the aspect ratio must satisfy $\varepsilon = H/L \ll 1$ to reinforce the shallow flow assumption, and quantities with hats represent the scaled variables. Herein, the downslope flow speed is assumed to be of the same order as the gravity wave celerity $U = \sqrt{gH}$ which is different from $U = \sqrt{gL}$ as used in Savage and Hutter [1989]. The difference between the gravity force and the resistance is of a lower order compared with the gravity force itself; thus the flow velocity can only be accelerated to the order of $\sqrt{gH}$ but not $\sqrt{gL}$. The same scalings have also been used in other literature (e.g. Gray and Edwards [2014]).

The scaled stress tensor components may be subsequently given as

$$
\begin{aligned}
&(p, \sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{xy}, \sigma_{yz}, \sigma_{xz}) = \\
&\rho g H(\hat{p}, \mu\hat{\sigma}_{xx}, \mu\hat{\sigma}_{yy}, \mu\hat{\sigma}_{zz}, \mu\hat{\sigma}_{xy}, \mu\hat{\sigma}_{yz}, \mu\hat{\sigma}_{xz})
\end{aligned}
\tag{5.7}
$$

According to the Mohr-Coulomb assumption, the magnitude of all the deviatoric tensor components are $O(\mu\rho g H)$, where μ is the friction coefficient.

In order to estimate the influence of bed curvature on the movement of flow-like landslides, it is also useful to scale the second-order derivatives of the bed elevation

$$
(\frac{\partial^2 b}{\partial x^2}, \frac{\partial^2 b}{\partial y^2}, \frac{\partial b}{\partial x \partial y}) = \varepsilon^{\gamma} L^{-1} (\frac{\partial^2 \hat{b}}{\partial \hat{x}^2}, \frac{\partial^2 \hat{b}}{\partial \hat{y}^2}, \frac{\partial \hat{b}}{\partial \hat{x} \partial \hat{y}}) \tag{5.8}
$$

Herein we introduced a free parameter γ to characterise the curvature of the topography because the magnitude of L/L^2 alone is not adequate to quantify the curvature. For example, a flat slope and a strongly curved slope both have the same L/L^2. For irregular topography, coefficient γ may usually take a value between 1 and 2, except for cases where the slope direction changes abruptly such as near a wall or cliff.

Finally the expanded non-dimensionalised governing equations are given as follows

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\varepsilon^{-1}\frac{\partial w}{\partial z}=0 \tag{5.9}$$

$$\begin{aligned}&\rho\varepsilon(\frac{\partial u}{\partial t}+\frac{\partial u^2}{\partial x}+\frac{\partial uv}{\partial y}+\varepsilon^{-1}\frac{\partial uw}{\partial z})=\\&-\varepsilon\frac{\partial p}{\partial x}+\varepsilon\mu\frac{\partial \sigma_{xx}}{\partial x}+\varepsilon\mu\frac{\partial \sigma_{xy}}{\partial y}+\mu\frac{\partial \sigma_{xz}}{\partial z}\end{aligned} \tag{5.10}$$

$$\begin{aligned}&\rho\varepsilon(\frac{\partial v}{\partial t}+\frac{\partial uv}{\partial x}+\frac{\partial v^2}{\partial y}+\varepsilon^{-1}\frac{\partial vw}{\partial z})=\\&-\varepsilon\frac{\partial p}{\partial y}+\varepsilon\mu\frac{\partial \sigma_{xy}}{\partial x}+\varepsilon\mu\frac{\partial \sigma_{yy}}{\partial y}+\mu\frac{\partial \sigma_{yz}}{\partial z}\end{aligned} \tag{5.11}$$

$$\begin{aligned}&\rho\varepsilon(\frac{\partial w}{\partial t}+\frac{\partial uw}{\partial x}+\frac{\partial vw}{\partial y}+\varepsilon^{-1}\frac{\partial w^2}{\partial z})=\\&-\frac{\partial p}{\partial z}+\varepsilon\mu\frac{\partial \sigma_{xz}}{\partial x}+\varepsilon\mu\frac{\partial \sigma_{yz}}{\partial y}+\mu\frac{\partial \sigma_{zz}}{\partial z}-\rho\end{aligned} \tag{5.12}$$

in which the hats have all been dropped for simplicity.

Defining the depth-averaged variables as

$$h=s-b,\ \overline{()}=\frac{1}{h}\int_b^s()dz \tag{5.13}$$

and incorporating the boundary conditions in (5.4), the three-dimensional non-dimensionalised governing equations (5.9-5.12) may be integrated along the vertical direction through the application of the Leibniz integral rule, leading to the following depth-integrated equations:

$$\frac{\partial h}{\partial t}+\frac{\partial h\bar{u}}{\partial x}+\frac{\partial h\bar{v}}{\partial y}=0 \tag{5.14}$$

$$\begin{aligned}\varepsilon(\frac{\partial h\bar{u}}{\partial t}+\frac{\partial h\overline{u^2}}{\partial x}+\frac{\partial h\overline{uv}}{\partial y})&=\varepsilon(-\frac{\partial h\bar{p}}{\partial x}+\mu\frac{\partial h\bar{\sigma}_{xx}}{\partial x}+\mu\frac{\partial h\bar{\sigma}_{xy}}{\partial y})\\&-p^b\frac{\partial b}{\partial x}+\mu\sigma^b_{xx}\frac{\partial b}{\partial x}+\mu\sigma^b_{xy}\frac{\partial b}{\partial y}-\mu\sigma^b_{xz}\end{aligned} \tag{5.15}$$

$$\begin{aligned}\varepsilon(\frac{\partial h\bar{v}}{\partial t}+\frac{\partial h\overline{uv}}{\partial x}+\frac{\partial h\overline{v^2}}{\partial y})&=\varepsilon(-\frac{\partial h\bar{p}}{\partial y}+\mu\frac{\partial h\bar{\sigma}_{xy}}{\partial x}+\mu\frac{\partial h\bar{\sigma}_{yy}}{\partial y})\\&-p^b\frac{\partial b}{\partial y}+\mu\sigma^b_{xy}\frac{\partial b}{\partial x}+\mu\sigma^b_{yy}\frac{\partial b}{\partial y}-\mu\sigma^b_{yz}\end{aligned} \tag{5.16}$$

$$\begin{aligned}\varepsilon(\frac{\partial h\bar{w}}{\partial t}+\frac{\partial h\overline{uw}}{\partial x}+\frac{\partial h\overline{vw}}{\partial y})&=\varepsilon\mu(\frac{\partial h\bar{\sigma}_{xz}}{\partial x}+\frac{\partial h\bar{\sigma}_{yz}}{\partial y})\\&+p^b+\mu\sigma^b_{xz}\frac{\partial b}{\partial x}+\mu\sigma^b_{yz}\frac{\partial b}{\partial y}-\mu\sigma^b_{zz}-h\end{aligned} \tag{5.17}$$

It is noted that the leading order balance in (5.17) (i.e. the same equation without all $O(\varepsilon)$ or smaller terms) is between the pressure and gravity plus the shear stresses. At the initial stage of the slope collapse, the left hand side (LHS) of (5.17), i.e. vertical acceleration, may be significant as well. This indicates that the hydrostatic assumption is no longer valid in a rectangular Cartesian coordinate system where the vertical axis is aligned with the direction of gravity. This brings in a major difficulty for deriving a closed depth-averaged equations.

The right hand side (RHS) terms in (5.15-5.17) may still be evaluated under the shallow "water" assumption, involving firstly the determination of the force normal to bed surface by projecting the momentum equations (5.15-5.17) onto the axis normal to be bed, i.e. basal normal direction. In order to simplify the expression of projected equations, it is useful to rewrite (5.15-5.17) into a single vectorised equation as

$$\mathbf{L} = \overline{\mathbf{T}} + \mathbf{T}^b + \mathbf{f} \tag{5.18}$$

where $\mathbf{L}$ is LHS of (5.15-5.17) and may be referred to as the depth integrated inertial force (acceleration times volume), $\overline{\mathbf{T}}$ is the depth-averaged force vector defined as

$$\overline{\mathbf{T}} = \begin{pmatrix} \varepsilon(-\frac{\partial h\bar{p}}{\partial x} + \mu\frac{\partial h\bar{\sigma}_{xx}}{\partial x} + \mu\frac{\partial h\bar{\sigma}_{xy}}{\partial y}), \\ \varepsilon(-\frac{\partial h\bar{p}}{\partial y} + \mu\frac{\partial h\bar{\sigma}_{xy}}{\partial x} + \mu\frac{\partial h\bar{\sigma}_{yy}}{\partial y}), \\ \varepsilon\mu(\frac{\partial h\bar{\sigma}_{xz}}{\partial x} + \frac{\partial h\bar{\sigma}_{yz}}{\partial y}) \end{pmatrix} \tag{5.19}$$

and $\mathbf{T}^b$ is the basal force vector

$$\mathbf{T}^b = \begin{pmatrix} -p^b\frac{\partial b}{\partial x} + \mu\sigma^b_{xx}\frac{\partial b}{\partial x} + \mu\sigma^b_{xy}\frac{\partial b}{\partial y} - \mu\sigma^b_{xz}, \\ -p^b\frac{\partial b}{\partial y} + \mu\sigma^b_{xy}\frac{\partial b}{\partial x} + \mu\sigma^b_{yy}\frac{\partial b}{\partial y} - \mu\sigma^b_{yz}, \\ p^b + \mu\sigma^b_{xz}\frac{\partial b}{\partial x} + \mu\sigma^b_{yz}\frac{\partial b}{\partial y} - \mu\sigma^b_{zz} \end{pmatrix} \tag{5.20}$$

and $\mathbf{f} = (0, 0, -h)^T$ is the gravity-related body force.

The projection of (5.15-5.17) onto the basal normal direction can be therefore written as

$$\mathbf{L}\cdot\mathbf{n}^b = \overline{\mathbf{T}}\cdot\mathbf{n}^b + \mathbf{T}\cdot\mathbf{n}^b - \frac{h}{\phi} \tag{5.21}$$

where $\mathbf{n}_b = \frac{1}{\phi}(-\frac{\partial b}{\partial x}, -\frac{\partial b}{\partial y}, 1)$ is the basal inward normal with $\phi = \{(\frac{\partial b}{\partial x})^2 + (\frac{\partial b}{\partial y})^2 + 1\}^{1/2}$.

For shallow flows, it is generally assumed that the velocity is (nearly) parallel to the bed surface, the relationship between the three components of the velocity $\mathbf{u}$ can be written in the following formal form:

$$w = u\frac{\partial b}{\partial x} + v\frac{\partial b}{\partial y} + O(\varepsilon) \tag{5.22}$$

After using (5.22) to replace the vertical velocity component w in $\mathbf{L}$, i.e. LHS of (5.15-5.17) which is inside the integral of the definition (5.13), and following certain algebraic manipulations, LHS of (5.21) becomes

$$\mathbf{L} \cdot \mathbf{n}^b = \frac{\varepsilon^{1+\gamma}}{\phi} h\overline{\mathbf{v}}^T \mathbf{H}\overline{\mathbf{v}} + O(\varepsilon^2) \tag{5.23}$$

where $\overline{\mathbf{v}} = (\overline{u}, \overline{v})^T$ and $\mathbf{H}$ is the Hessian matrix defined as

$$\mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 b}{\partial x^2} & \dfrac{\partial b}{\partial x \partial y} \\ \dfrac{\partial b}{\partial x \partial y} & \dfrac{\partial^2 b}{\partial y^2} \end{bmatrix} \tag{5.24}$$

Herein, it is also assumed

$$\overline{u^2} = \overline{u}^2, \overline{v^2} = \overline{v}^2, \overline{uv} = \overline{u}\overline{v} \tag{5.25}$$

In (5.23), the first term at the right hand side (RHS) indeed contains the centrifugal forces, which have been derived naturally from the above depth-integrating procedure without making any ad-hoc assumptions. Now the basal normal force can be easily evaluated from (5.21) by substituting (5.23) into (5.21)

$$T_n^b = \mathbf{T}^b \cdot \mathbf{n}^b = \frac{h}{\phi}(1 + \varepsilon^{1+\gamma}\overline{\mathbf{v}}^T \mathbf{H}\overline{\mathbf{v}}) + O(\varepsilon) \tag{5.26}$$

in which the term $\overline{\mathbf{T}} \cdot \mathbf{n}^b$ in (5.21) has become part of the residual term $O(\varepsilon)$ as evidenced from (5.19).

With T_n^b, it is possible to obtain RHS terms in (5.15-5.17) or the vectorised equation (5.18). The first step is to make an approximation of $\mathbf{T}^b$. According to the Mohr-Coulomb rheology assumption, the basal tangential stresses acting on the bed is given as

$$\boldsymbol{\tau}^b = -(\frac{h}{\phi}(1+\varepsilon^{1+\gamma}\overline{\mathbf{v}}^T\mathbf{H}\overline{\mathbf{v}}) + O(\varepsilon)) \cdot \mu\frac{\mathbf{u}^b}{|\mathbf{u}^b|} \tag{5.27}$$

The vertical velocity component w^b in $\mathbf{u}^b$ is related to the horizontal velocity components u^b and v^b through

$$w^b = u^b\frac{\partial b}{\partial x} + v^b\frac{\partial b}{\partial y} \tag{5.28}$$

Substituting (5.28) into (5.27) leads to

$$\boldsymbol{\tau}^b = -(\frac{h}{\phi}(1+\varepsilon^{1+\gamma}\overline{\mathbf{v}}^T\mathbf{H}\overline{\mathbf{v}}) + O(\varepsilon)) \cdot \frac{\mu\mathbf{u}^b}{\sqrt{{u^b}^2 + {v^b}^2 + (u^b\frac{\partial b}{\partial x} + v^b\frac{\partial b}{\partial y})^2}} \tag{5.29}$$

The basal force (i.e. $\mathbf{T}^b$) can be expressed as the combination of the basal normal force and tangential force, i.e.

$$\begin{aligned}\mathbf{T}^b &= T_n^b\mathbf{n}^b + \boldsymbol{\tau}^b \\ &= (\frac{h}{\phi^2}(1+\varepsilon^{1+\gamma}\overline{\mathbf{v}}^T\mathbf{H}\overline{\mathbf{v}}) + O(\varepsilon)) \\ &\quad \cdot(\phi\mathbf{n}^b - \frac{\mu\mathbf{u}^b\phi}{\sqrt{{u^b}^2 + {v^b}^2 + (u^b\frac{\partial b}{\partial x} + v^b\frac{\partial b}{\partial y})^2}})\end{aligned} \tag{5.30}$$

It is straightforward to infer from (5.15-5.16) that $\mathbf{T}^b$ is $O(\varepsilon)$ in the x and y directions because all other terms are $O(\varepsilon)$. Therefore $\phi\mathbf{n}^b - \frac{\mu\mathbf{u}^b\phi}{\sqrt{{u^b}^2+{v^b}^2+(u^b\frac{\partial b}{\partial x}+v^b\frac{\partial b}{\partial y})^2}}$ in (5.30) must be $O(\varepsilon)$ otherwise $\mathbf{T}^b$ will become $O(1)$. Assuming that the velocity is linearly distributed along the depth, the basal velocities in (5.30) can also be replaced by the depth averaged velocities. Then (5.30) can be rearranged to become

$$\mathbf{T}^b = \frac{h}{\phi^2}(1+\varepsilon^{1+\gamma}\overline{\mathbf{v}}^T\mathbf{H}\overline{\mathbf{v}}) \cdot (\phi\mathbf{n}^b - \frac{\mu\mathbf{u}^b\phi}{\sqrt{\overline{u}^2 + \overline{v}^2 + (\overline{u}\frac{\partial b}{\partial x} + \overline{v}\frac{\partial b}{\partial y})^2}}) + O(\varepsilon^2) \tag{5.31}$$

An approximation of $\mathbf{T}^b$ can now be obtained by omitting the higher order $O(\varepsilon^2)$ term in (5.31). The next step is make an approximation of the depth-averaged force $\overline{\mathbf{T}}$. Firstly, p^b

must also be estimated and may be obtained from the following relationship

$$T_n^b = \mathbf{T}^b \cdot \mathbf{n}^b = (p^b - \mu \mathbf{n}^b \cdot \sigma^b \mathbf{n}^b)\phi \tag{5.32}$$

where the second equality results from some simple algebraic manipulations. Combining it with (5.26) gives the following explicit expression for

$$p^b = \frac{h}{\phi^2} + O(\varepsilon) + O(\varepsilon^{1+\gamma}) + O(\mu) \tag{5.33}$$

Assuming linear pressure distribution along the vertical direction, the depth averaged pressure can be obtained

$$\overline{p} = \frac{1}{2}p^b = \frac{h}{2\phi^2} + O(\varepsilon^{1+\gamma}) + O(\varepsilon) + O(\mu) \tag{5.34}$$

Subsequently, we have

$$\frac{\partial h\bar{p}}{\partial x} = \frac{h}{\phi^2}\frac{\partial h}{\partial x} + \frac{1}{2}h^2\frac{\partial(1/\phi^2)}{\partial x} + O(\varepsilon^{1+\gamma}) + O(\varepsilon) + O(\mu) \tag{5.35}$$

The corresponding expression in the y-direction can be derived in a similar way.

Substituting (5.35) into (5.19) gives

$$\overline{\mathbf{T}} = -\varepsilon(\frac{h}{\phi^2}\nabla h, 0)^T + \varepsilon(\frac{1}{2}h^2\nabla(1/\phi^2), 0)^T + O(\varepsilon^{2+\gamma}) + O(\varepsilon^2) + O(\varepsilon\mu) \tag{5.36}$$

Again, the higher order term $O(\varepsilon^2)$ is small thus can be omitted, similarly the $O(\varepsilon^{2+\gamma})$ term representing the variation of depth-averaged centrifugal force. The second term in the RHS of (5.36) is also dropped due to the following reason: 1) $\partial(1/\phi^2)/\partial x$ inside this term is indeed a function of the second-order derivatives of b with a magnitude of $O(\varepsilon^\gamma)$, the entire second term is thus $O(\varepsilon^{1+\gamma})$, which is small enough in most places except for where slope direction changes abruptly; 2) omission of this term ensures the preservation of lake at rest solution, which will be explained in more details in section 2.2.1. The last $O(\varepsilon\mu)$ term is linked to the shear stresses determined by the Mohr-Coulumb rheology, which can be taken into account by introducing a lateral stress coefficient K. Then (5.36) becomes

$$\overline{\mathbf{T}} = -\frac{h}{\phi^2}Kgh(\nabla h, 0)^T \tag{5.37}$$

A popular method to determine the lateral stress coefficient was derived by Savage and Hutter [1989]. However the approach cannot be directly used here because it was originally derived on a local coordinate system with the vertical axis normal to the bed. A rotated stress tensor should be applied to obtain a new lateral stress coefficient, but the resulting formula is highly complicated. Actually the validity of the lateral pressure coefficient introduced by Savage and Hutter [1989] is still in debate and the more recent molecular dynamic simulations (Silbert et al. [2003], GDR Midi [2004]) suggested that the lateral pressure coefficient is actually closer to unity. In this work, the lateral stress coefficient is taken as unity (i.e. $K = 1$) for simplicity but more complicated formula can be adopted in the future. This value has been also adopted by many other researchers to develop their models (e.g. Bouchut and Westdickenberg [2004], Pudasaini and Hutter [2003], Gray et al. [1999]). Substituting the above forces terms, i.e. (5.25), (5.31) and (5.37) into (5.21), the depth-averaged momentum equations in the x and y-direction can be finally obtained

$$\begin{aligned}&\varepsilon(\frac{\partial h\bar{u}}{\partial t} + \frac{\partial h\bar{u}^2}{\partial x} + \frac{\partial h\bar{u}\bar{v}}{\partial y})\\ &= \varepsilon(-\frac{h}{\phi^2}\frac{\partial h}{\partial x}) - ah(\frac{\partial b}{\partial x} + \frac{\mu\bar{u}\phi}{\sqrt{\bar{u}^2+\bar{v}^2+(\bar{u}\frac{\partial b}{\partial x}+\bar{v}\frac{\partial b}{\partial y})^2}})\end{aligned} \tag{5.38}$$

$$\begin{aligned}&\varepsilon(\frac{\partial h\bar{v}}{\partial t} + \frac{\partial h\bar{u}\bar{v}}{\partial x} + \frac{\partial h\bar{v}^2}{\partial y})\\ &= \varepsilon(-\frac{h}{\phi^2}\frac{\partial h}{\partial y}) - ah(\frac{\partial b}{\partial y} + \frac{\mu\bar{v}\phi}{\sqrt{\bar{u}^2+\bar{v}^2+(\bar{u}\frac{\partial b}{\partial x}+\bar{v}\frac{\partial b}{\partial y})^2}})\end{aligned} \tag{5.39}$$

$$a = \frac{1}{\phi^2}(1 + \varepsilon^{1+\gamma}\overline{\mathbf{v}}^T\mathbf{H}\overline{\mathbf{v}}) \tag{5.40}$$

The final depth-averaged governing equations for mass movement are obtained after reformulating (5.14) and (5.38-5.39) with the dimensional variables and dropping the bars above the depth-averaged variables for simplified expression:

$$\frac{\partial h}{\partial t} + \frac{\partial hu}{\partial x} + \frac{\partial hv}{\partial y} = 0 \tag{5.41}$$

$$\frac{\partial hu}{\partial t} + \frac{\partial hu^2}{\partial x} + \frac{\partial huv}{\partial y} = -\frac{gh}{\phi^2}\frac{\partial h}{\partial x} - ah(\frac{\partial b}{\partial x} + \frac{\mu u \phi}{\sqrt{u^2 + v^2 + (u\frac{\partial b}{\partial x} + v\frac{\partial b}{\partial y})^2}}) \tag{5.42}$$

$$\frac{\partial hv}{\partial t} + \frac{\partial huv}{\partial x} + \frac{\partial hv^2}{\partial y} = -\frac{gh}{\phi^2}\frac{\partial h}{\partial y} - ah(\frac{\partial b}{\partial y} + \frac{\mu v \phi}{\sqrt{u^2 + v^2 + (u\frac{\partial b}{\partial x} + v\frac{\partial b}{\partial y})^2}}) \tag{5.43}$$

$$a = \frac{1}{\phi^2}(g + \mathbf{v}^T\mathbf{H}\mathbf{v}) \text{ with } \mathbf{H} = \begin{bmatrix} \dfrac{\partial^2 b}{\partial x^2} & \dfrac{\partial b}{\partial x \partial y} \\ \dfrac{\partial b}{\partial x \partial y} & \dfrac{\partial^2 b}{\partial y^2} \end{bmatrix} \tag{5.44}$$

$$\phi = \{(\frac{\partial b}{\partial x})^2 + (\frac{\partial b}{\partial y})^2 + 1\}^{1/2} \tag{5.45}$$

The above equations appear to be similar to the shallow water equations, which may be beneficial in terms of adopting existing numerical methods originally developed for the shallow water equations. In addition to the friction and centrifugal force terms ($\mathbf{v}^T\mathbf{H}\mathbf{v}$), the major difference is that the terms at the RHS of (5.42) and (5.43) as well as the gravity term are simply multiplied by a factor of $1/\phi^2$. This factor is only related to the bed topography and is independent of either the coordinate system or the velocity direction, which is an essential condition to ensure rotational invariance of the above depth-averaged equations. The inclusion of the factor is theoretically important in order for the governing equations to properly describe the effect of complex topography in a Cartesian coordinate system.

The above depth-averaged governing equations are subsequently rewritten in a conservative matrix form to facilitate the implementation of a finite volume Godunov-type scheme:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{q})}{\partial x} + \frac{\partial \mathbf{g}(\mathbf{q})}{\partial y} = \mathbf{S}_b + \mathbf{S}_f \tag{5.46}$$

where the vector terms are given by

$$\mathbf{q} = \begin{bmatrix} h \\ uh \\ vh \end{bmatrix} \mathbf{f}(\mathbf{q}) = \begin{bmatrix} uh \\ u^2h + \dfrac{1}{\phi^2}\dfrac{1}{2}gh^2 \\ uvh \end{bmatrix} \mathbf{g}(\mathbf{q}) = \begin{bmatrix} vh \\ uvh \\ v^2h + \dfrac{1}{\phi^2}\dfrac{1}{2}gh^2 \end{bmatrix} \tag{5.47}$$

$$\mathbf{S}_b = \begin{bmatrix} 0 \\ -ah\dfrac{\partial b}{\partial x} + \dfrac{1}{2}gh^2\dfrac{\partial(1/\phi^2)}{\partial x} \\ -ah\dfrac{\partial b}{\partial y} + \dfrac{1}{2}gh^2\dfrac{\partial(1/\phi^2)}{\partial y} \end{bmatrix} \mathbf{S}_f = \begin{bmatrix} 0 \\ \dfrac{a\mu u\phi}{\sqrt{u^2+v^2+(u\frac{\partial b}{\partial x}+v\frac{\partial b}{\partial y})^2}} \\ \dfrac{a\mu v\phi}{\sqrt{u^2+v^2+(u\frac{\partial b}{\partial x}+v\frac{\partial b}{\partial y})^2}} \end{bmatrix} \tag{5.48}$$

Compared with the relevant formulations reported in the literature (e.g. Juez et al. [2013], Hergarten and Robl [2015]), the new depth-averaged mass flow governing equations (5.46-5.48) have been rigorously derived and mathematically justified. More importantly, the current equations introduce new terms to better represent the physics behaviour of mass flows. Firstly, the new terms $\frac{1}{2}gh^2\frac{\partial(1/\phi^2)}{\partial x}$ and $\frac{1}{2}gh^2\frac{\partial(1/\phi^2)}{\partial y}$ in $\mathbf{S}_b$ which are directly resulted from the omission of the second term in (5.36) are essential for the mathematical preservation of the lake at rest condition as will be explained in the next sub-section; numerical experiments demonstrate that neglecting these terms will lead to spurious wave-like solution if the domain topography is highly irregular. Secondly, the inclusion of new centrifugal force term (i.e. $\mathbf{v}^T\mathbf{H}\mathbf{v}$ in (5.44)) ensures that the flow velocity is calculated correctly on curved topography that is often encountered in real-world simulations.

The most pronounced effect of the non-hydrostatic pressure is that the gravity g is reduced by a factor $1/\phi^2$, so that flow modelled by the current equations could move slower than that modelled by the hydrostatic shallow water equations. The effect of the centrifugal force is more complex. On the one hand, it increases the normal pressure and hence the friction force, so that the movement modelled by the new equations could be slower. On the other hand, the centrifugal force can also results in faster movement of flow in some certain situations. Considering flow moving from an inclined slope into a horizontal plane, the vertical velocity component will disappear when it moves into the horizontal plane if the centrifugal force is not considered, leading to the reduction of velocity on the horizontal plane. Adding the centrifugal force will retain the full magnitude of the velocity so that the the flow can actually moves faster.

Compared with the more complicated Boussinesq-like models (e.g. Denlinger and Iverson [2004] and Castro-Orgaz et al. [2014]), the models based on the current depth-averaged governing equations are computationally much less demanding due to the use of much simplified formulation and subsequently the less sophisticated numerical schemes.

5.2.2 *Compatibility with other types of rheology*

The derivation here relies on a the shallow water assumption and the scalings in (5.5-5.7). Sometimes the internal rheology is known or assumed. In such a case, changing internal rheology only leads to the difference of the stress scaling (5.7). In (5.7), all the stresses except for the normal pressure p leads to an $O(\varepsilon\mu)$ term that has been finally dropped in our derivation. For less viscous fluid or material with similar frictional rheology like μI rheology, we can still safely ignore these stresses. So all the derivation of the non-hydrostatic pressure and centrifugal force remains the same and we can still have a similar set of equations with only the exception of the basal resistance terms. But sometimes the basal resistance terms are empirical rather than derived formally by assuming an internal rheology. We may take the pragmatic approach, i.e, replacing the basal resistance terms without changing other terms. In other words, the current model can be used for simulating flows with other types of rheology simply by changing the basal resistance terms in both cases.

5.2.3 *Properties of the new depth-averaged equations*

In this section, the new depth-averaged governing equations will be proved to satisfy three important properties, i.e. mathematical preservation of the lake at rest solution, hyperbolicity and rotational invariance, which are essential to guarantee accurate numerical solutions in the context of implementing a Godunov-type numerical scheme.

1. Mathematical preservation of the lake at rest solutions

Preservation of the lake at rest solution is a necessary condition to ensure numerically stable and physically correct simulation results. The lake at rest conditions may be defined as

$$h + b = const \text{ and } u = 0, v = 0 \tag{5.49}$$

To prove the preservation of these conditions, we may substitute (5.49) into (5.46-5.48) and the continuity equation is automatically satisfied. Herein we focus on the x-direction momentum equation and the y-direction equation can be proved in a similar way. After eliminating all terms containing zero velocities, the x-direction momentum equation reduces to

$$\frac{\partial}{\partial x}(\frac{1}{\phi^2}\frac{1}{2}gh^2) = -\frac{gh}{\phi^2}\frac{\partial b}{\partial x} + \frac{1}{2}gh^2\frac{\partial}{\partial x}(\frac{1}{\phi^2}) \tag{5.50}$$

After applying the chain rule and simple manipulations, the equation can be further simplified to become

$$\frac{gh}{\phi^2}\frac{\partial h}{\partial x} = -\frac{gh}{\phi^2}\frac{\partial b}{\partial x} \tag{5.51}$$

which is equivalent to

$$\frac{gh}{\phi^2}\frac{\partial(h+b)}{\partial x} = 0 \tag{5.52}$$

Since we have $h + b = const$, the above equation is apparently satisfied. Therefore, the new governing equations mathematically preserve the lake at rest solution.

The above balancing property is a direct result of the inclusion the additional terms $\frac{1}{2}gh^2\frac{\partial(1/\phi^2)}{\partial x}$ in the x-direction momentum equation and $\frac{1}{2}gh^2\frac{\partial(1/\phi^2)}{\partial y}$ in the y-direction equation in (5.46-5.48). The governing equations presented by Juez et al. [2013] do not include these two terms and hence cannot mathematically preserve the lake at rest solution. A special numerical technique must be introduced to discretise the slope source terms in order to ensure the preservation of the lake at rest solution at the discrete level. As a result, their simulation results are highly dependent on the cell edge orientation and the common rectangular Cartesian grids may not even produce acceptable results.

2. Hyperbolicity

The Jacobian matrix corresponding to the flux terms is given by

$$\mathbf{J} = \frac{\partial \mathbf{F}(\mathbf{q})}{\partial \mathbf{q}} = \begin{bmatrix} 0 & n_x & n_y \\ (c^2 - u^2)n_x - uvn_y & 2un_x + vn_y & un_y \\ -uvn_x + (c^2 - u^2)n_y & vn_x & un_x + 2vn_y \end{bmatrix} \tag{5.53}$$

where the flux vector $\mathbf{F}(\mathbf{q}) = \mathbf{f}(\mathbf{q})n_x + \mathbf{g}(\mathbf{q})n_y$, with n_x and n_y being the two Cartesian components of the unit vector, $c = \frac{\sqrt{gh}}{\phi}$ is the wave speed, the eigenvalues of $\mathbf{J}$ are

$$\lambda_1 = un_x + vn_y + c,\ \lambda_2 = un_x + vn_y,\ \lambda_3 = un_x + vn_y - c \tag{5.54}$$

Obviously, these eigenvalues are distinct real values when $h \neq 0$; therefore the corresponding homogeneous equations of (5.46-5.48) are strictly hyperbolic when $h \neq 0$. This implies that the new depth-averaged mass flow governing equations can be numerically solved using a range of Godunov-type numerical schemes that have been established for solving the shallow water equations.

3. Rotational invariance

Rotationally invariant equations are independent of the choice of coordinate directions, therefore facilitating the implementation of a more robust numerical scheme and ensuring more reliable simulation results. Adopting the following rotational matrix

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \tag{5.55}$$

in which θ is an arbitrary angle. It is straightforward to verify that the fluxes of the depth-averaged equations satisfy

$$\cos\theta\mathbf{f}(\mathbf{q}) + \sin\theta\mathbf{g}(\mathbf{q}) = \mathbf{T}^{-1}\mathbf{f}(\mathbf{Tq}) \tag{5.56}$$

and therefore are rotational invariant. Similarly, the source terms can be also proved to be rotationally invariant and so the overall formulation strictly satisfies the rotational invariant requirement.

5.3 Numerical scheme

In this work, the hyperbolic conservation laws formed by the newly derived depth-averaged governing equations are solved using a Godunov-type finite volume scheme to produce automatic shock-capturing numerical solutions. In order to effectively preserve the lake at rest solution at the discrete level, and handle wetting and drying, particular attention is paid to implement a hydrostatic reconstruction approach (Audusse et al. [2004]) with effective improvements and discretise the source terms in a divergent form. A fractional splitting method is used to evaluate the friction source terms to reinforce the non-negative numerical solutions.

5.3.1 *Finite volume discretisation*

A finite volume numerical scheme solves the integrated form of governing equations and the resulting semi-discretized equation for an arbitrary cell '*i*' is given by

$$\frac{\partial \mathbf{q}_i}{\partial t} + \frac{1}{\Omega_i}\sum_{k=1}^{N}\mathbf{F}_k(\mathbf{q})l_k = \mathbf{S}_{bi} + \mathbf{S}_{fi} \tag{5.57}$$

where '*k*' is the index of the cell edges ($N = 4$ for Cartesian grids adopted in this work) of an arbitrary cell '*i*', l_k is the length of cell edge '*k*' and Ω_i is the cell area.

Herein, the overall solution procedure adopts a fractional splitting method to update the flow variables to a new time level, i.e. $\mathbf{q}_i^{n+1}$, where n denotes the time level. In the first step, the flow variables are updated against only the interface fluxes and slope source terms using a second-order Runger-Kutta time marching scheme as follows

$$\mathbf{q}_i^{**} = \mathbf{q}_i^n + \Delta t^n \mathbf{K}(\mathbf{q}_i^n) \tag{5.58}$$

$$\mathbf{q}_i^{*} = \mathbf{q}_i^n + \frac{\Delta t^n}{2}(\mathbf{K}(\mathbf{q}_i^n) + \mathbf{K}(\mathbf{q}_i^{**})) \tag{5.59}$$

where the Runge-Kutta coefficient is defined as

$$\mathbf{K}(\mathbf{q}_i) = \mathbf{S}_{bi} - \frac{1}{\Omega_i}\sum_{k=1}^{N}\mathbf{F}_k(\mathbf{q}_i)l_k \tag{5.60}$$

In the second step, the flow variables are fully updated by also taking into account the friction source terms using the following time-marching formula

$$\mathbf{q}_i^{n+1} = \mathbf{q}_i^{*} + \Delta t^n \mathbf{S}_{fi}^{*} \tag{5.61}$$

Details of the calculation for each term will be given in the following sub-sections.

5.3.2 *Interface flux calculation*

The interface fluxes in (5.57) are calculated by solving the Riemann problems defined locally at each cell interface, i.e. $\mathbf{F}(\mathbf{q}) = \mathcal{F}(\mathbf{q}_L, \mathbf{q}_R)$, where $\mathbf{q}_L$ and $\mathbf{q}_R$ are the Riemann states.

In order to obtain the Riemann states, firstly the linear reconstruction as originally proposed by van Leer [1979] is used to estimate the face values of flow variables from their cell-centred values:

$$f_L^* = f_i + \Psi(r_i)\mathbf{r_i}\nabla f_i \text{ and } f_R^* = f_j + \Psi(r_j)\mathbf{r_j}\nabla f_j \tag{5.62}$$

where f denotes a flow variable, 'j' represents a neighbour of cell 'i', the subscripts 'L' and 'R' represent 'left' and 'right', $\mathbf{r}$ is the distance vector from the cell centre to the face centre, $\Psi(r)$ is a limiter function to ensure monotonicity during the interpolation to suppress spurious oscillations in the numerical solutions. In this work, the minmod slope limiter is adopted, which is defined on a rectangular Cartesian grid for cell 'i' as

$$\Psi(r_i) = \max[0, \min(r_i, 1)] \text{ and } r_i = \frac{f_+ - f_i}{f_i - f_-} \tag{5.63}$$

in which the subscripts $+$ and $-$ denote the upstream and downstream cells, respectively.

With the face values, the final left and right Riemann states are obtained by implementing the hydrostatic reconstruction method as proposed by Audusse et al. [2004] to avoid negative water depth. The first step is to define a single bed elevation at the cell interface under consideration:

$$b_f = \max(b_L^*, b_R^*) \tag{5.64}$$

where b_L^* and b_R^* are calculated by finding the difference between the linearly reconstructed face values of water surface elevation (i.e. $\eta = h + b$) and water depth. Then the left and right Riemann states of the water depth are redefined as

$$\begin{cases} h_L = \max(0, h_L^* + b_L^* - b_f) \\ h_R = \max(0, h_R^* + b_R^* - b_f) \end{cases} \tag{5.65}$$

which are then used to reconstruct other Riemann states

$$\begin{cases} [hu]_L = h_L u_L^*, \quad [hv]_L = h_L v_L^* \\ [hu]_R = h_R u_R^*, \quad [hv]_R = h_R v_R^* \end{cases} \tag{5.66}$$

The Riemann states are then used to define local Riemann problems, which are numerically solved to evaluate the interface fluxes. Due to the similarity between the new mass flow governing equations (5.46-5.48) and the classic shallow water equations, a standard approach developed for the shallow water equations can be used with certain modifications. We assume that both sides of the cell interface share the same modified gravity acceleration, which may be simply determined by finding the average between the two neighbouring cells:

$$g_f = \frac{1}{2} g (\frac{1}{{\phi_i}^2} + \frac{1}{{\phi_j}^2}) \tag{5.67}$$

where $1/{\phi_i}^2$ is calculated in cell 'i' as

$$\phi_i = \{ (\frac{\partial b}{\partial x})_i^2 + (\frac{\partial b}{\partial y})_i^2 + 1 \}^{\frac{1}{2}} \tag{5.68}$$

where the gradient terms of b can be calculated simply using central difference method.

Therefore solving (5.46-5.48) essentially becomes seeking numerical solution to the shallow water equations with a modified but still constant gravity acceleration. An HLLC approximate Riemann solver (Toro [2001]) already introduced in Chapter 4, is implemented herein to evaluate the interface fluxes with the only exception that g is replaced by g_f.

5.3.3 *Slope source term discretisation and preservation of the lake at rest solutions*

Apart from the flux calculation, the slope source term $\mathbf{S}_b$ must also be discretised properly to preserve the lake at rest solutions. A new formulation different from that in the original hydrostatic reconstruction method (Audusse et al. [2004]) is derived to adapt to the new governing equations.

To discretise the slope source term $\mathbf{S}_b$, we first integrate $\mathbf{S}_b$ over the entire domain Ω of an arbitrary cell, i.e.

$$\begin{aligned} \int_\Omega \mathbf{S}_b d\Omega &= \int_\Omega \left[-ah\nabla b + \frac{1}{2} g h^2 \nabla (\frac{1}{\phi^2}) \right] d\Omega && (5.69) \\ &= \int_\Omega \left[\nabla (\frac{1}{2\phi^2} g h^2) + (a-g) \nabla (\frac{1}{2} h^2) - ah\nabla \eta \right] d\Omega && (5.70) \end{aligned}$$

In a spatially second-order scheme, the gradient term $\nabla\eta$ is constant within a cell, leading to the following relationship:

$$\int_{\Omega} h\nabla\eta d\Omega = \int_{\Omega} \frac{1}{2}\nabla(\eta - b_M)^2 d\Omega \tag{5.71}$$

where b_M is defined at the centre of the domain, essentially the cell centred value. After applying the Green-Gauss theorem, the discretised equation for the slope source term $\mathbf{S}_b$ can be obtained as

$$\mathbf{S}_{bi} = \frac{1}{\Omega_i}\sum_{k=1}^{N}[\frac{1}{2}(g_k + a_i - g_i){h_{Lk}}^2 - \frac{1}{2}a_i(\eta_k - b_i)^2]\mathbf{n}_k l_k \tag{5.72}$$

where g_k, calculated by (5.67), is the modified gravity acceleration at cell edge 'k'; $g_i = g/\phi_i^2$ is defined at the cell centre; h_{Lk} is the hydrostatic reconstructed water depth at cell edge 'k' obtained according to (5.65); a_i is calculated according to (5.44) with the second-order derivatives approximated using central differences. Unlike the original hydrostatic reconstruction method (Audusse et al. [2004]), the above slope source term discretisation scheme does not involve any additional cell centred source term to maintain the consistency of the overall numerical scheme, removing extra effort in numerical implementation.

The preservation of the lake at the rest solutions of the overall numerical scheme can be proved as follows. For the lake at rest condition: $\eta = const$ and $\mathbf{v} = 0$, the centrifugal force term $\mathbf{v}^T\mathbf{H}\mathbf{v}$ vanishes, thus $a_i = g_i$, and η_k is equal to η_i at all cell edges; hence $\sum_{k=1}^{N}\left[\frac{1}{2}a_i(\eta_k - b_i)^2\right]\mathbf{n}_k l_k$ will also disappear. As a result, the discretised $\mathbf{S}_{bi}$ becomes

$$\mathbf{S}_{bi} = \frac{1}{\Omega_i}\sum_{k=1}^{N}\frac{1}{2}g_k {h_{Lk}}^2\mathbf{n}_k l_k \tag{5.73}$$

It is trivial to verify that (5.73) exactly balances the summation of flux at all cell edges. Therefore the lake at rest solutions are maintained.

5.3.4 *Friction source term calculation*

The friction source terms in (5.48) represent the kinetic frictions. If the soil is static and the forces induced by the pressure gradient and slope gradient is smaller than the kinetic friction force, a static resistant force smaller than the kinetic friction will exist to balance exactly the other forces to maintain the static soil. Friction forces alone can only stop the flow but can never reverse the flow. In order to represent this physical reality in the numerical model, the friction source term must be restricted as follows:

$$\mathbf{S}_f = \begin{cases} \mathbf{S}_f^* & \text{if } |\mathbf{S}_f|\Delta t < |\mathbf{f}| \\ -\mathbf{f}/\Delta t & \text{if } |\mathbf{S}_f|\Delta t >= |\mathbf{f}| \end{cases} \tag{5.74}$$

where $\mathbf{f} = (0, hu, hv)^T$ contains the flow variables updated using the second-order Runge-Katta scheme as introduced previously; $\mathbf{S}_f^*$ is calculated from (5.48) with the updated water depth h and velocity vector $\mathbf{v}$ at the cell centre.

Due to the use of the above restriction measures, the current friction discretization scheme will never revert the flow. It is also straightforward to verify that the current scheme maintains the static soil condition. If kinetic friction is larger than other forces, the velocity will be slowed down to nil in the friction source term discretisation as provided in (5.61). Therefore, the physical nature of the friction forces is correctly interpreted in the numerical scheme.

5.4 Results and validation

In this section, the new mass flow model as presented in the previous sections is validated against several test cases, including an idealised uniform flow on an inclined frictional slope, two experimental granular flow tests and two real-world flow-like landslides.

5.4.1 *A uniform but unsteady flow on an inclined frictional slope*

A uniform but unsteady flow on an inclined frictional slope may test the effect of the non-hydrostatic pressure and is considered herein. Being a simple test, the analytical solution may be derived. This test has been considered by a few other researchers (e.g. Hergarten and Robl [2015]). The flow occurs on a slope with an angle of θ and friction coefficient of μ. The flow depth is h and the depth-averaged velocity $\tilde{u}$ is parallel to the slope surface. The flow acceleration can be then obtained by applying the Newton's second law

$$\frac{d\tilde{u}}{dt} = g \sin\theta - \mu g \cos\theta \tag{5.75}$$

which can be projected to the x-direction as

$$\frac{du}{dt} = g \cos\theta \sin\theta - \mu g \cos^2\theta \tag{5.76}$$

For a uniform but unsteady flow, $\frac{\partial u}{\partial x}$ and $\frac{\partial h}{\partial x}$ are both 0 and hence the flow acceleration given by our new depth-averaged equation is

$$\frac{du}{dt} = -\frac{g}{\phi^2}(\frac{\partial b}{\partial x} + \mu) \tag{5.77}$$

Considering that $\frac{\partial b}{\partial x} = -\tan\theta$ and $\frac{1}{\phi^2} = \cos^2\theta$, (5.77) and (5.76) become identical. The new mass flow model presented in this work can therefore exactly recover the uniform but unsteady flow on a frictional inclined slope.

However, if the conventional shallow water equations are used without the modified gravity acceleration, the flow acceleration will be predicted as

$$\frac{du}{dt} = -\frac{\partial b}{\partial x} - \mu g = g\tan\theta - \mu g \tag{5.78}$$

where a factor of $\cos^2\theta$ has been missed out compared with the correct solution. For real-world landslides, the slope can be as steep as $30° - 40°$. In such cases, the conventional shallow water equations will overestimate the flow acceleration up to 30%, which will lead to unacceptable errors in velocity and run-out distance predictions.

5.4.2 *USGS granular flow experiment*

An experimental granular flow reported by Denlinger and Iverson [2001] is used herein to test the current non-hydrostatic depth-averaged landslide model. The experiment was undertaken in USGS. In the experiment, 290 cm^3 dry quartz was placed behind a gate on an inclined flume. The flume consists of three parts: an inclined slope of $31.4°$, a curved section with a 10 cm radius and a horizontal run-out surface at the end. The width of the flume is 20 cm. A sketch of the experiment setup is shown in Figure 5.2. The friction angle between the granule and the bed is 29° as suggested by Denlinger and Iverson [2001]. During the experiment, the gate was suddenly opened to release the granule, which flowed rapidly downstream through the flume and finally deposited at the horizontal surface.

The simulated flow profile at different output times are shown in Figure 5.3. As the gate is opened, the mass front rapidly moves down the slope. When the mass front reaches the horizontal plane, it begins to deposit to form a pile, which is consistent with the observation during the experiment. The measured and predicted flow extents, defined by the front and

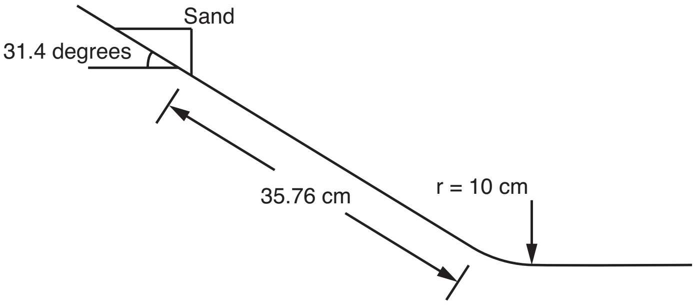


Figure 5.2: USGS granular flow experiment: experiment setup.

rear positions of flow at different times, are compared in Figure 5.4. Satisfactory agreement has been achieved, demonstrating the capability of our model in reproducing this laboratory test. In order to investigate the influence of non-hydrostatic pressure and curvature effects, we respectively turned off non-hydrostatic correction, i.e. set $1/\phi^2 = 1$ and centrifugal forces and plotted the corresponding front and rear positions in Figure 5.4. Clearly, both the non-hydrostatic pressure and curvature effects significantly influences the results. Negligence of non-hydrostatic pressure leads to faster mass movement and the deposition of flow front at a further position. On the contrary, the negligence of curvature effect causes slower mass motion and nearer deposition position of the flow material because the velocity component normal to the bed surface disappears when the bed slope direction changes if the curvature is not considered.

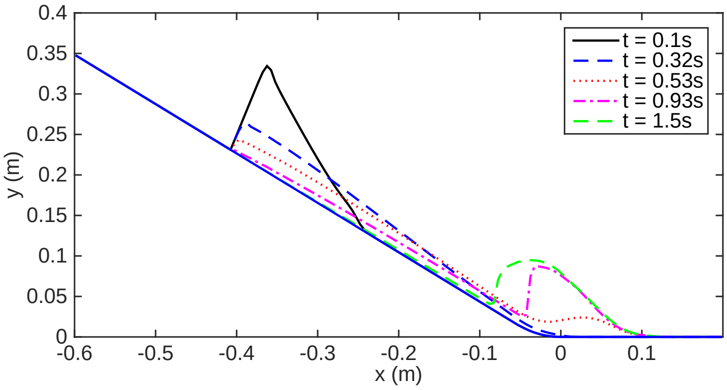


Figure 5.3: USGS granular flow experiment: simulated flow profile at different output times (the depths are amplified 5 times for clearer illustration).

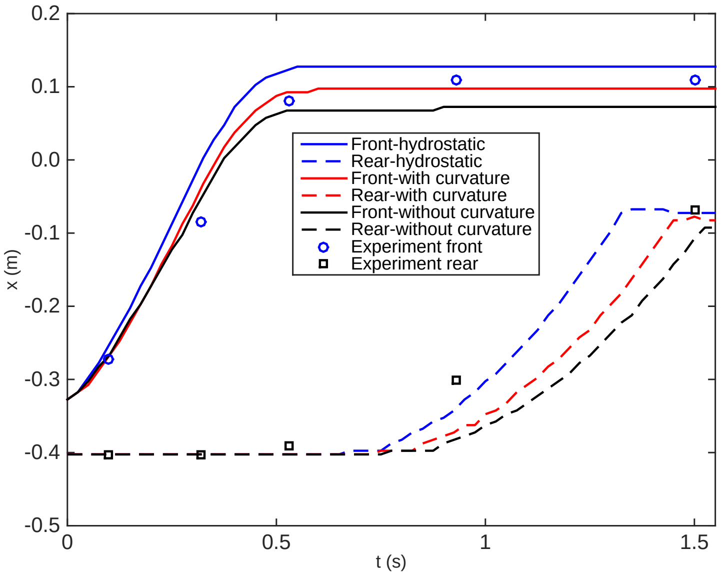


Figure 5.4: USGS granular flow experiment: measured and predicted front and rear positions of flow.

5.4.3 *Experiment granular flow with an obstacle*

In this section, the granular flow experiment reported by Gray et al. [2003] is simulated using the new model. The topography of the experiment is illustrated in Figure 5.5, featured with a pyramid shaped obstacle on an inclined flume. Starting from time zero, granule was fed from the upstream side of the flume with a constant velocity and depth.

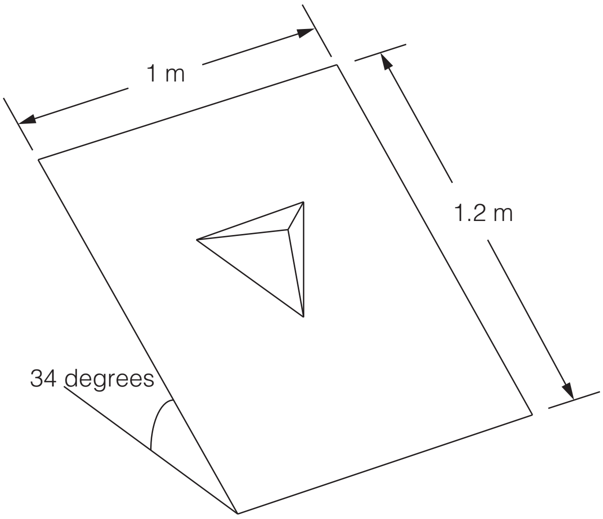


Figure 5.5: Experimental granular flow with an obstacle: the topography of the experiment.

During the simulation, the friction angle between the flume and the granule is chosen as 29°. The simulated flow patterns are shown in Figure 5.6 for different output times, compared with the snapshots taken at the same moments during the experiments. As the granule flow moves to downstream, the flow front hits and is blocked by the obstacle. The Coulomb basal friction balances the internal pressure gradients caused by the obstacle to create a stationary zone. At the meantime, a detached shock wave is formed upstream and separates the flow into two streams because the inflow velocity is greater than the wave speed $\sqrt{gh}$ in this case. Such a phenomenon is physically similar to a 'hydraulic jump' in hydraulics. The simulated results are observed to be consistent with the experimental records. Particularly, the location of the simulated shock wave front and the extent of the stationary zone both match well with those provided by the experiment, demonstrating that the new model is able to capture complex flow characteristics such as shock waves that form around complex topography.

5.4.4 *2015 Heifangtai landslide, China*

Herein a real-world landslide event featured with complex topography is further simulated to confirm the simulation capability of the new model. As shown by the satellite image in Figure 5.7, a landslide happened in 2015 at the Heifangtai loess platform located 42 km to the west of Lanzhou city, in Yongjing county, Gansu province, China. Heifangtai loess platform is prone to landslides as a result of intensive agricultural irrigation. Following irrigation, water seeps into the slope to form saturated bands with low shear strength and initiates slope failure. The platform can be as high as 100 m with steep slopes up to 30°, creating a large gravitational potential that may drive a landslide to travel rapidly for a long distance. For example, the particular landslide considered in this work travelled nearly 500 m. The depth of the landslides (~1 m) was much smaller than its horizontal span (~100 m). Therefore the depth-averaged model presented herein provides an appropriate tool to simulate this type of landslides. As can be seen from Figure 5.7, the topography of the study site is complex featured with two trenches developed along the slope direction. The landslide induced mass flow mainly travelled along the trenches and deposited along the whole flow paths.

During the simulation, a constant bed friction angle of 14° is used, which is chosen to be close to the typical residual shear strength ($12^\circ - 15^\circ$) of the loess in this area. The total volume of the landslide body was about 0.1 million m^3. Simulation results at $t = 15$ s, $t = 30$ s, $t = 45$ s and $t = 60$ s are plotted in Figure 5.8 to present the major dynamic characteristics of the landslide. At the beginning of the event, the soil moves down rapidly through the channels; after 30 seconds, the landslide reaches its maximum runout distance; then it begins to deposit and completely settled down after 60 seconds. Deposited material can be found along almost

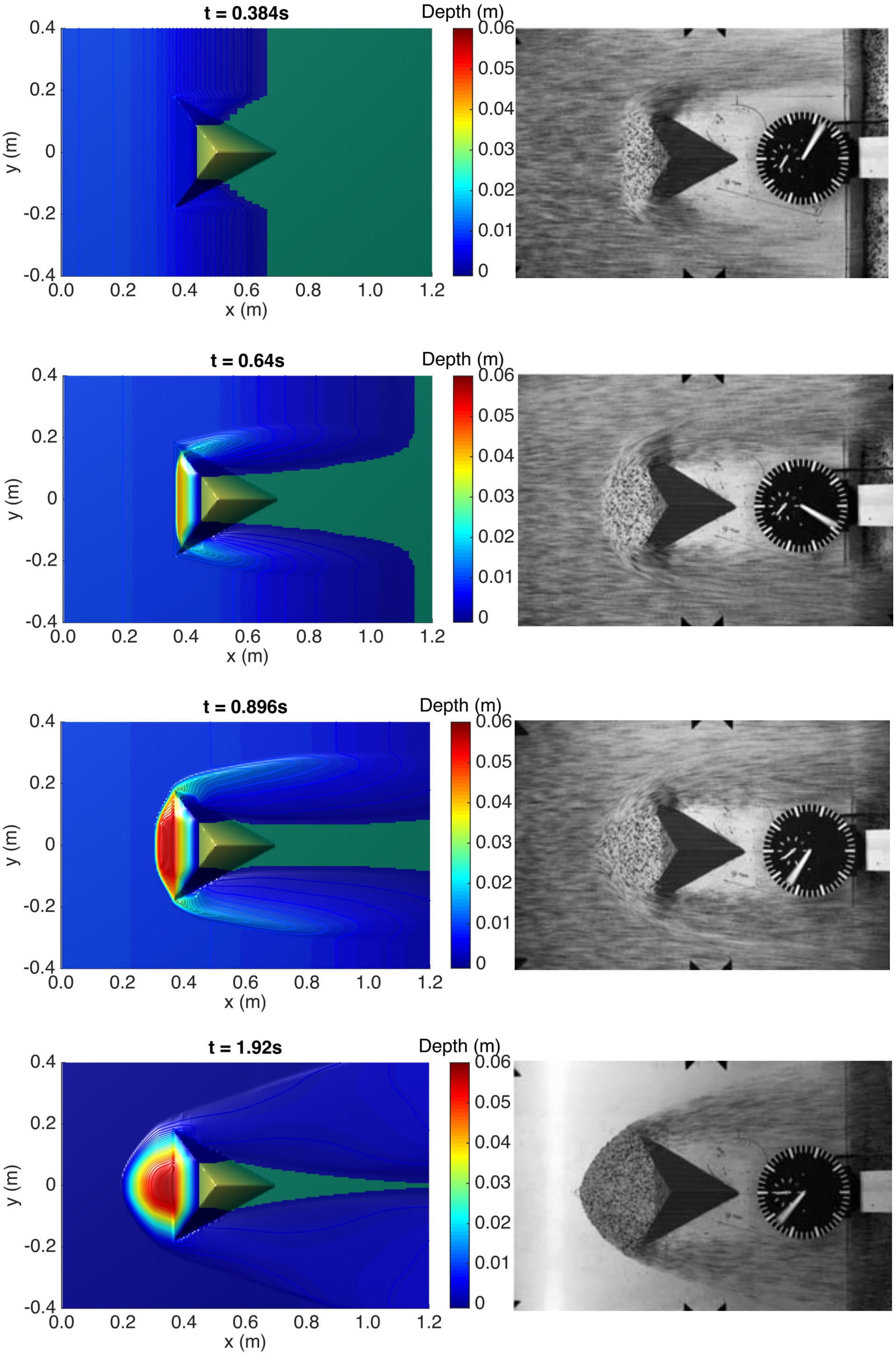


Figure 5.6: Experimental granular flow with an obstacle: simulated (left column) and measured (right column; from Gray et al. [2003]) flow patterns at different output times (right column).

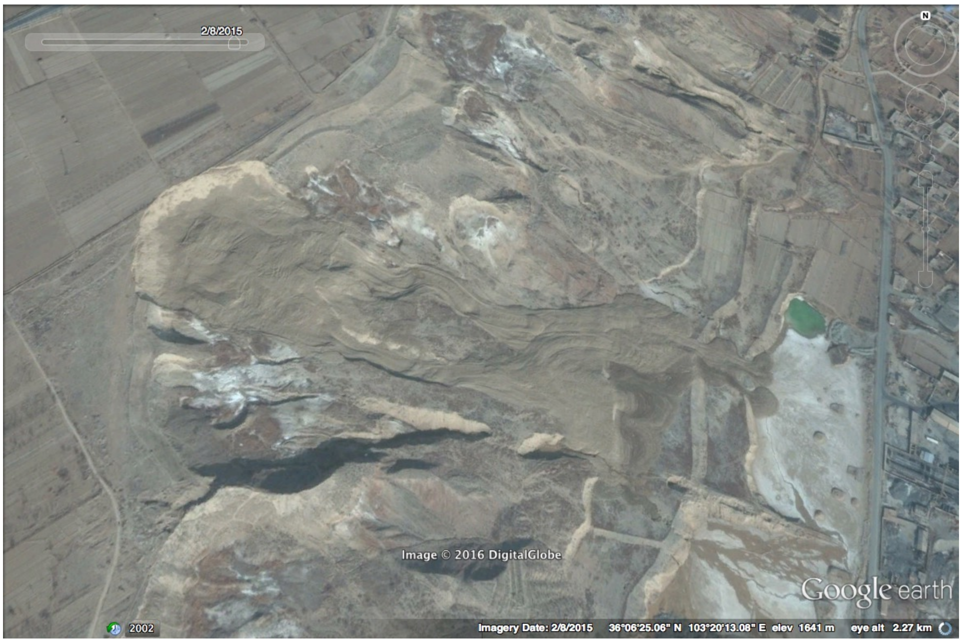

2/8/2015
Image © 2016 DigitalGlobe
Google earth
2002
Imagery Date: 2/8/2015 36°06'25.06" N 103°20'13.08" E elev 1641 m eye alt 2.27 km

Figure 5.7: Satellite image of the Heifangtai 2015 landslide. (Image from google.com and copyright of DigitalGlobe)

the whole slope, as also shown in the satellite image. The simulated deposition extent matches generally well with the observation from the satellite image, despite obvious discrepancy in the final flow front. In the satellite image, a small amount of soil flowed further and formed a tongue shape deposition in the northern part, which is not reproduced by the current simulation. A possible reason is that part of the soil may have higher water content thus been fluidised more thoroughly to travel for a longer distance. This effect however is not taken into account in the current model. Generally, the simulation successfully reproduces the overall flow process of the event and provides a deposited extent generally comparable with that shown in the field. This confirms the simulation capability of the present non-hydrostatic depth-average landslide model and demonstrates its potential for simulating flow-like landslides with complex topographic settings.

5.4.5 *Socompa rock avalanche, Chile*

The Socompa debris avalanche deposit is at the border of Argentina and Chile. Although this avalanche occurred about 7000 years ago, the hyperarid climate of northern Chile has well preserved the morphology of its deposit. Therefore, this avalanche has been intensively studied in the past few decades (e.g. Van Wyk De Vries et al. [2001], Kelfoun and Druitt [2005]). Kelfoun and Druitt [2005] reconstructed the topography before the collapse based on the description by Van Wyk De Vries et al. [2001] (Figure 5.9). The avalanche deposit was formed by a gravita-

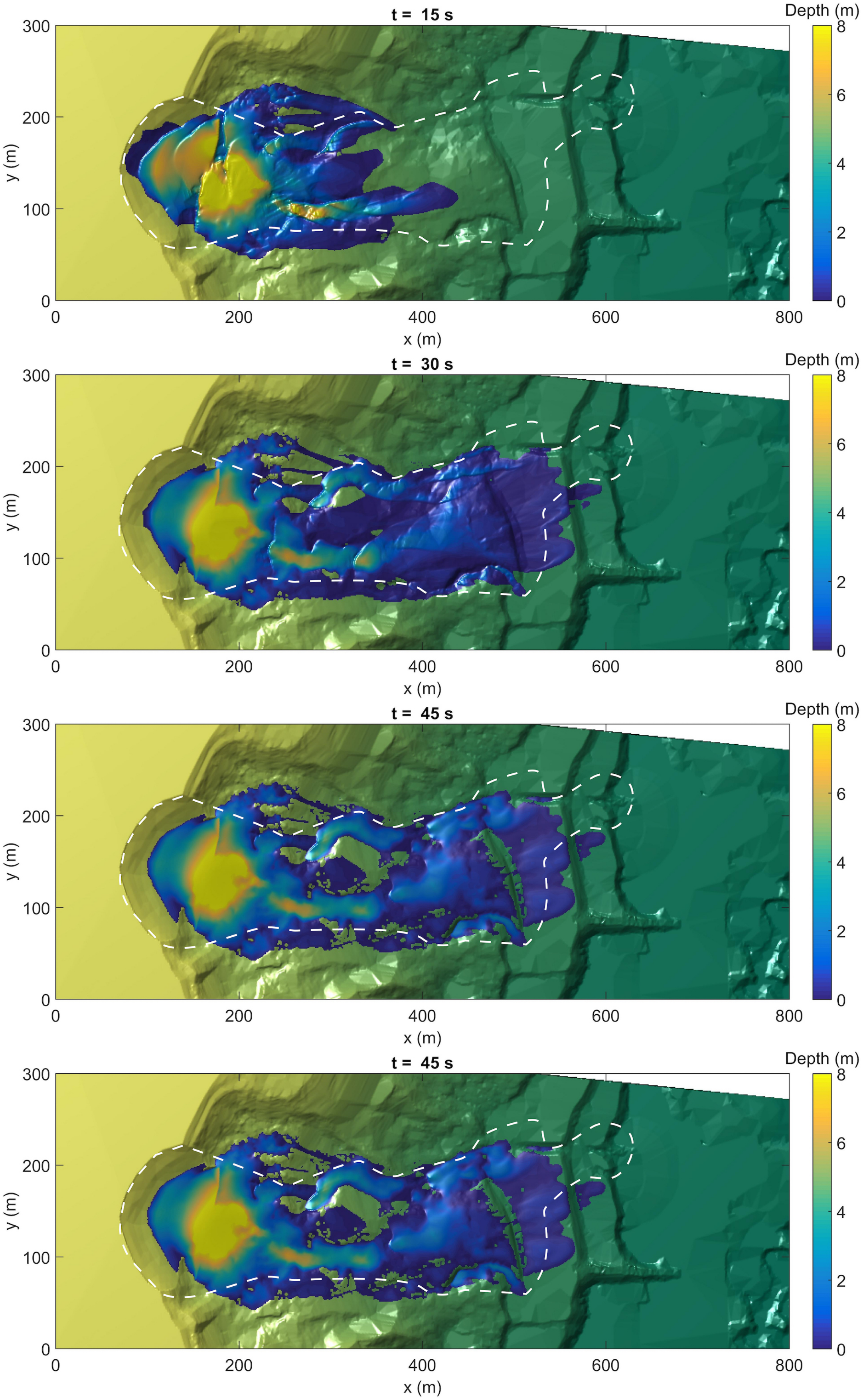


Figure 5.8: Simulated flow depths of 2015 Heifangtai landslide at different moments, from top to bottom: $t = 15$ s, $t = 30$ s, $t = 45$ s and $t = 60$ s.

tional failure of the northwestern flank of the 6000-m-high stratovolcano (Van Wyk De Vries et al. [2001]), leaving a 300 to 400-m-high cliff and a 12-km-wide amphitheater. The dropping distance from the summit of the volcano to the ground was about 3000 m. Kelfoun and Druitt [2005] simulated this avalanche with a depth-averaged model that is based on the curvilinear coordinate system. They found that using a constant retarding stress of 52 kPa reproduces the final extent and morphology of the avalanche deposit surprisingly well.

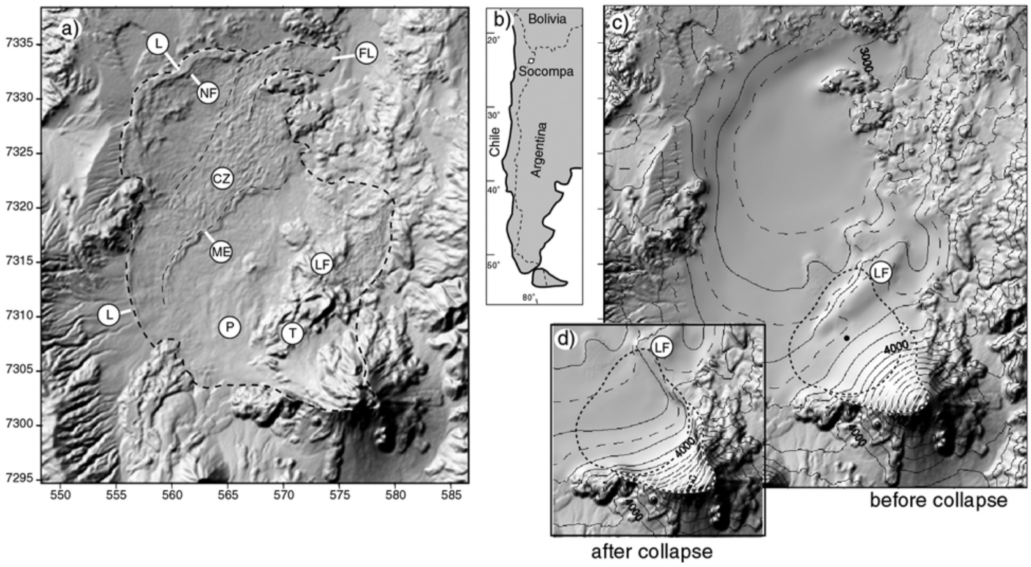


Figure 5.9: Socompa volcano: satellite image, location and reconstructed topography before and after the collapse (Image from Kelfoun and Druitt [2005]).

In this section, we further tested our model's capability by simulating this avalanche using the same type of basal resistance term. For the current non-hydrostatic depth-averaged formulation, the basal resistance term with a constant retarding stress is given as

$$\mathbf{S}_f = \begin{bmatrix} 0 \\ \dfrac{\tau_b u}{\rho\sqrt{u^2 + v^2 + (u\frac{\partial b}{\partial x} + v\frac{\partial b}{\partial y})^2}} \\ \dfrac{\tau_b v}{\rho\sqrt{u^2 + v^2 + (u\frac{\partial b}{\partial x} + v\frac{\partial b}{\partial y})^2}} \end{bmatrix} \tag{5.79}$$

where τ_b is the constant retarding stress and ρ is the density of the material. During this simulation, we used the same topography that was used by Kelfoun and Druitt [2005], which has a spatial resolution of 100 m. ρ is set as 2000 kg/m^3 following Kelfoun and Druitt [2005].

Flow depths at different moments are plotted together with the result by Kelfoun and Druitt

[2005] in Figure 5.10. After the collapse of the volcano, the majority of debris moves rapidly towards northwest. As the debris is being blocked by the mountains at the west, a reflected wave is moving towards the centre and forming the middle escarpment (ME in Figure 5.9). The debris moving towards north is diverted towards northeast to become the frontal lobe (FL in Figure 5.9). After 600 s, the avalanche have reached its maximum extent. The simulated extent and morphology of the avalanche debris generally matches well with the observation while an obvious difference can be found at the northwest front (L in Figure 5.9) where the simulated runout extent is further than the observation. Such a difference is possibly caused by the inadequacy of the basal resistance term or the uncertainty of initial conditions. These results are very similar to those reported in Kelfoun and Druitt [2005]; we use a constant stress of 55 kPa that is only slightly different from 52 kPa used by Kelfoun and Druitt [2005]. Considering that we used different coordinate systems and numerical methods, such a small difference can be regarded as acceptable.

5.5 Conclusions

This chapter presents a new non-hydrostatic depth-averaged model for simulating flow-like landslides. Based on the shallow flow assumption and the Mohr-Coulomb rheology, new depth-averaged governing equations are derived in a mathematical rigorous way through asymptotic analysis in a global Cartesian coordinate system. The new equations take into account the non-hydrostatic vertical pressure and centrifugal acceleration due to bed curvature. The final governing equations are rotational invariant, hyperbolic and mathematically well-balanced to preserve the lake at rest solution. A second-order Godunov-type finite volume numerical scheme is then implemented to solve the new governing equations. To preserve the lake at rest solutions and facilitate a non-negative approach for treating wetting and drying, the hydrostatic reconstruction approach is implemented with certain modifications proposed in the context of the new governing equations. A simple fractional scheme with a novel discretisation scheme for friction terms is proposed to properly simulate the static resistance and stopping condition. The new model is therefore mathematically and numerically rigorous to support landslide simulations over complex real-world terrains.

The new model has been validated against several test cases. The model is able to reproduce the exact solution to the uniform flow on a frictional slope and satisfactory results for two laboratory scale test cases, including the more complex case of granular flow interacting with

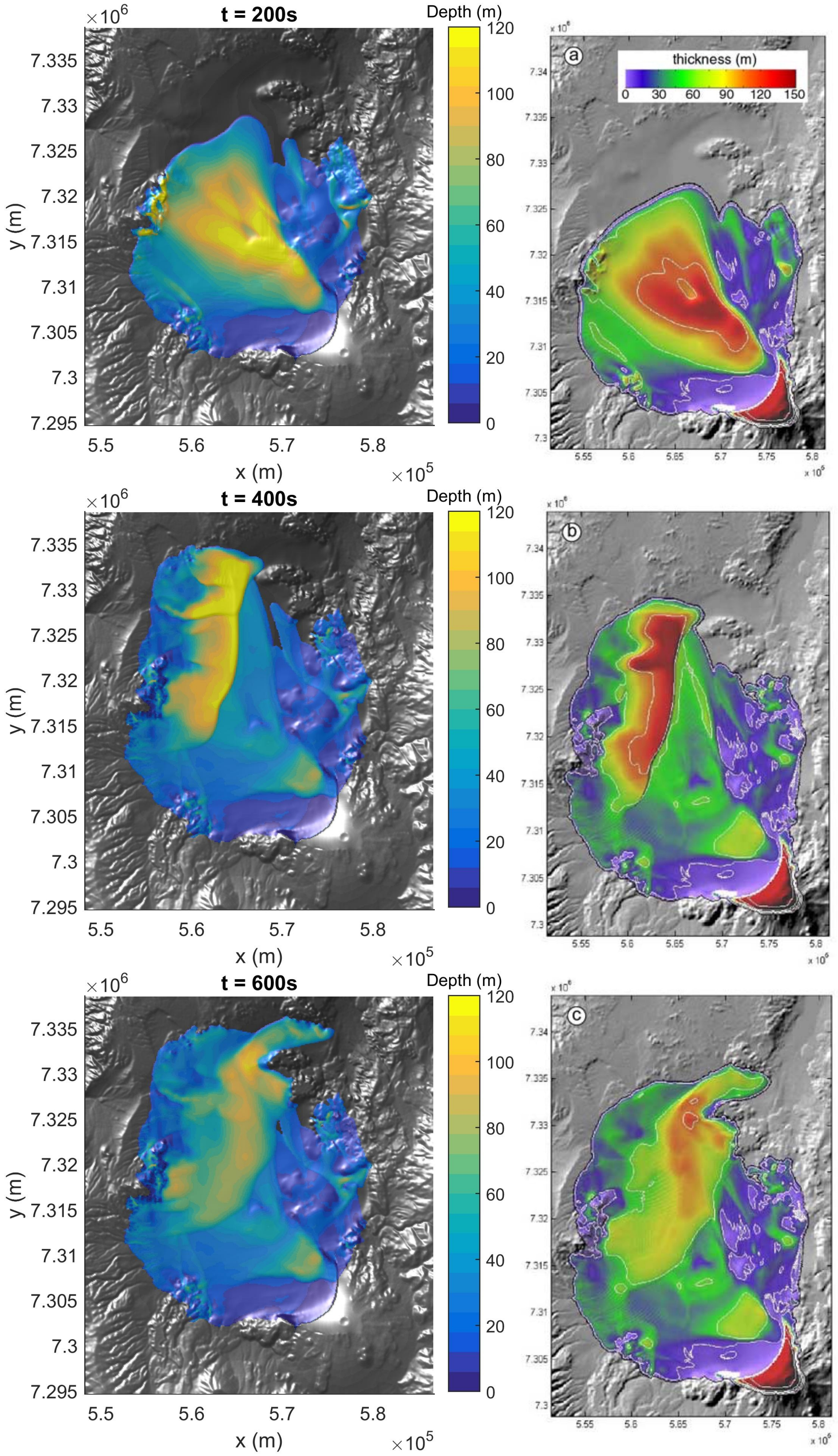


Figure 5.10: Snapshots of Socompa debris avalanche simulation. Left column: simulation by the current model. Right column: simulation by Kelfoun and Druitt [2005]. From top to bottom: $t = 200$ s, $t = 400$ s and $t = 600$s.

an obstacle. Then the model is applied to simulate two field-scale landslides with complex topographic features. The model is able to successfully capture the major characteristics of the landslide dynamics and predict a final deposited extent and morphology comparable to that shown in satellite images or simulation results reported in the literature. This confirms the simulation capability of the current model and its potential for a wider range of applications.

The new model can be used as a standalone tool for robust simulation of flow-like landslides for analysing the run-out distance and flow dynamics. More importantly, the new governing equations and numerical schemes provide a firm basis for developing the proposed integrated multi-hazard modelling system. Because the new model is formulated on a global Cartesian coordinate system and compatible with other types of rheology, it enables us to simulate different types of mass and water flows by varying the rheology in a single modelling framework. This will be further demonstrated in the next chapter by combining the current model with Manning's friction law to simulate flood waves.

Chapter 6. A high-performance modelling framework for multiple hazards

6.1 Introduction

The shallow water equations are based on global Cartesian coordinate system. On the contrary, most depth-averaged equations for flow-like landslides are based on surface fitted curvilinear coordinate system in order to properly take into account the non-hydrostatic vertical pressure and centrifugal force due steep slopes and large curvature. The use of different coordinate systems makes it difficult to simulate water flows and mass flows in a single codebase.

Chapter 5 presented a new non-hydrostatic depth-averaged model for simulating flow-like landslides on global Cartesian coordinate systems. We also indicated that the non-hydrostatic governing equations can also incorporate other types of basal resistance terms in addition to the Mohr-Coulomb friction terms. Therefore, the non-hydrostatic depth-averaged equations derived in Chapter 5 may become a unified formulation for simulating both water flows and mass flows. To demonstrate this capability, the depth-averaged equations presented in Chapter 5 is combined with Manning's friction law to simulate water flows, i.e. flood waves in this chapter.

In this chapter, we also present a flexible GPU-accelerated codebase that supports long term development of this model.

6.2 Governing equations

The governing equations used in this chapter only differs from the depth averaged equations derived in chapter 5, i.e. (5.46-5.48) in terms of the basal resistance terms. The Manning friction law is used herein as the basal resistance terms, which is the same as the formulae in chapter 4 (4.3-4.4). For completeness, the governing equations are summarised as follows

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}(\mathbf{q})}{\partial x} + \frac{\partial \mathbf{g}(\mathbf{q})}{\partial y} = \mathbf{S}_b + \mathbf{S}_f \tag{6.1}$$

with the vector terms given by

$$\mathbf{q} = \begin{bmatrix} h \\ uh \\ vh \end{bmatrix} \mathbf{f}(\mathbf{q}) = \begin{bmatrix} uh \\ u^2h + \frac{1}{\phi^2}\frac{1}{2}gh^2 \\ uvh \end{bmatrix} \mathbf{g}(\mathbf{q}) = \begin{bmatrix} vh \\ uvh \\ v^2h + \frac{1}{\phi^2}\frac{1}{2}gh^2 \end{bmatrix} \tag{6.2}$$

$$\mathbf{S}_b = \begin{bmatrix} 0 \\ -ah\frac{\partial b}{\partial x} + \frac{1}{2}gh^2\frac{\partial(1/\phi^2)}{\partial x} \\ -ah\frac{\partial b}{\partial y} + \frac{1}{2}gh^2\frac{\partial(1/\phi^2)}{\partial y} \end{bmatrix} \mathbf{S}_f = \begin{bmatrix} 0 \\ C_f u\sqrt{u^2+v^2} \\ C_f v\sqrt{u^2+v^2} \end{bmatrix} \tag{6.3}$$

where the notations are identical to those used in Chapter 4 and Chapter 5.

6.3 Numerical scheme

The modified hydrostatic reconstruction method is used herein to solve the governing equations with the exception that a second-order new Implicit-Explicit (IMEX) Runge-Kutta scheme is used as the time-marching scheme to handle the non-linearity of the friction term. Based on the work of Pareschi and Russo [2005], we devised the following second-order IMEX Runge-Kutta scheme:

Step 1:

$$\mathbf{q}_i^1 = \mathbf{q}_i^n + \Delta t[\mathbf{K}(\mathbf{q}_i^n) + \mathbf{S}_{fi}^1] \tag{6.4}$$

Step 2:

$$\mathbf{q}_i^2 = \mathbf{q}_i^n + \frac{1}{4}\Delta t[\mathbf{K}(\mathbf{q}_i^n) + \mathbf{K}(\mathbf{q}_i^1)] + \frac{1}{2}\Delta t\mathbf{S}_{fi}^2 \tag{6.5}$$

Step 3:

$$\mathbf{q}_i^3 = 2\mathbf{q}_i^n - \mathbf{q}_i^2 + \frac{1}{4}\Delta t[\mathbf{K}(\mathbf{q}_i^n) + \mathbf{K}(\mathbf{q}_i^1)] + \frac{1}{2}\Delta t\mathbf{S}_{fi}^3 \tag{6.6}$$

Step 4:

$$\mathbf{q}_i^{n+1} = \mathbf{q}_i^2 + \mathbf{q}_i^3 - \mathbf{q}_i^n + \frac{1}{4}\Delta t[\mathbf{K}(\mathbf{q}_i^n) + \mathbf{K}(\mathbf{q}_i^1)] + \frac{1}{2}\Delta t\mathbf{S}_{fi}^{n+1} \tag{6.7}$$

In each step, the Runge-Kutta coefficient is defined the same as (5.60) while the friction terms are computed implicitly using the method described in section 4.4.3. We will now prove that the IMEX Runge-Kutta scheme can relax the velocity to the correct equilibrium between friction term and bed slope term in a single time step when the friction is very large. Firstly, step 3 and 4 are reformulated into the following form:

Step 3:

$$\mathbf{q}_i^3 = \mathbf{q}_i^n - \frac{1}{4}\Delta t[\mathbf{K}(\mathbf{q}_i^n) + \mathbf{K}(\mathbf{q}_i^1)] - \frac{1}{2}\Delta t\mathbf{S}_{fi}^2 + \frac{1}{4}\Delta t[\mathbf{K}(\mathbf{q}_i^n) + \mathbf{K}(\mathbf{q}_i^1)] + \frac{1}{2}\Delta t\mathbf{S}_{fi}^3 \tag{6.8}$$

Step 4:

$$\mathbf{q}_i^{n+1} = \mathbf{q}_i^n + \frac{1}{4}\Delta t[\mathbf{K}(\mathbf{q}_i^n) + \mathbf{K}(\mathbf{q}_i^1)] + \frac{1}{2}\Delta t\mathbf{S}_{fi}^3 + \frac{1}{4}\Delta t[\mathbf{K}(\mathbf{q}_i^n) + \mathbf{K}(\mathbf{q}_i^1)] + \frac{1}{2}\Delta t\mathbf{S}_{fi}^{n+1} \tag{6.9}$$

If we take $\Delta t \to \infty$, we can obtain from (6.5) that

$$\frac{1}{4}[\mathbf{K}(\mathbf{q}_i^n) + \mathbf{K}(\mathbf{q}_i^1)] + \frac{1}{2}\mathbf{S}_{fi}^2 = \mathbf{0} \tag{6.10}$$

Substituting (6.10) into (6.8) and take $\Delta t \to \infty$, we can obtain that

$$\frac{1}{4}[\mathbf{K}(\mathbf{q}_i^n) + \mathbf{K}(\mathbf{q}_i^1)] + \frac{1}{2}\mathbf{S}_{fi}^3 = \mathbf{0} \tag{6.11}$$

Again, substituting (6.11) into (6.9) and take $\Delta t \to \infty$, we can obtain that

$$\frac{1}{4}[\mathbf{K}(\mathbf{q}_i^n) + \mathbf{K}(\mathbf{q}_i^1)] + \frac{1}{2}\mathbf{S}_{fi}^{n+1} = \mathbf{0} \tag{6.12}$$

If the convective flux terms in (6.12) are neglected, we can find the following balance between bed slope terms and friction terms

$$\frac{1}{2}(\mathbf{S}_{bi}^n + \mathbf{S}_{bi}^1) = \mathbf{S}_{fi}^{n+1} \tag{6.13}$$

Then the velocity can be recovered as

$$\mathbf{u}_i^{n+1} = \sqrt{\frac{\|\frac{1}{2}(\mathbf{S}_{bi}^n + \mathbf{S}_{bi}^1)\|}{C_f}} \cdot \frac{\frac{1}{2}(\mathbf{S}_{bi}^n + \mathbf{S}_{bi}^1)}{\|\frac{1}{2}(\mathbf{S}_{bi}^n + \mathbf{S}_{bi}^1)\|} \tag{6.14}$$

which is effectively the velocity at the equilibrium. Therefore, the IMEX Runge-Kutta scheme can relax the velocity to the correct equilibrium between friction and bed slope in a single time step when the friction is dominantly large. Because of this property, the new IMEX Runge-Kutta scheme is accurate and stable when dealing with non-linear friction terms.

6.4 Software design

6.4.1 *Programming language*

As one of the most popular programming language in software engineering, C++ provides the highest efficiency and enormous versatility to programmers. It is a pre-compiled language so that it avoids the overhead of runtime interpreting needed by script languages such as Java and Python. It supports multiple paradigms such as imperative, object-oriented and generic programming features, which makes designing flexible code structure possible. Moreover, it can be easily combined with NVIDIA CUDA. Therefore, C++ has been chosen to write the main structure of the code. For the components doing computations on GPUs, we use CUDA like we did for the SPH-SWEs model.

6.4.2 *Architecture of the code*

In order to achieve flexibility, we adopt the following strategy: Firstly, we abstract a common pipeline from the computing processes. Secondly, we standardise the interfaces between the steps of the pipeline. Thirdly, we program each step as a component of the code. This strategy brings in some important benefits. First of all, because different computing processes share a common structure, new functionality can be coded easily by combining the components (steps of the pipeline). These components can be easily replaced by others because the interface remains the same, so that existing code can be updated straightforwardly. Then the question is how to design such a pipeline and the interfaces between its steps.

We design the pipeline and interfaces between its steps by analysing the procedure of solving partial differential equations (PDEs) because the proposed multi-hazard modelling system essentially models the physical processes by solving PDEs. Considering the following conservation laws

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{U}}{\partial x} = 0 \tag{6.15}$$

A semi-discrete explicit scheme for solving the above equation can be given as

$$\mathbf{U}^{n+1} = \mathbf{U}^{n} - \Delta t \frac{\partial \mathbf{U}}{\partial x} \tag{6.16}$$

where n and $n + 1$ denotes time step. The basic quantity involved in (6.16) is a field, i.e. a variable $\mathbf{U}$, on which the differential operator $\frac{\partial}{\partial x}$ is applied. The resulting gradient $\frac{\partial \mathbf{U}}{\partial x}$ is also a field, which is then multiplied by a scalar $-\Delta t$ and added to $\mathbf{U}^n$ to obtain a new field $\mathbf{U}^{n+1}$. This process involves two basic type of elements: field like $\mathbf{U}$ and the functions operating on them like $\frac{\partial}{\partial x}$ and $-$ which are termed as operators. Solving more complex PDEs like the SWEs may need a more complicated process but the ingredients are essentially the same. If we define these operators as basic components of the code and take fields as input and output, we can design the pipelines and interfaces very naturally. Sequential arrangement of operators become the pipeline and fields such as $\mathbf{U}$ become the interfaces between different components.

Based on this strategy, we designed the architecture of the code as is shown in Figure 6.1. It contains three major parts: fields, operators and solvers. Fields and operators correspond to the concepts in the procedure of solving PDEs that is aforementioned. Fields are implemented as classes while operators are functions. Solver is the entry of the code, i.e. the main function. It defines the fields and calls the operator functions for solving certain PDEs.

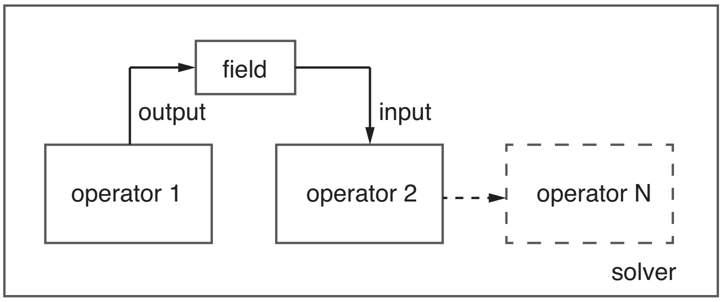


Figure 6.1: Architecture of the code.

6.4.3 *Implementation of the key components*

In this section, more details of implementing the three major parts, i.e. fields, operators and solvers are presented. The details presented in this section also serve as a guideline for understanding the source code and undertaking further development.

1. Field

Rather than a continuous function of space and time, a field in the code is a finite set of values defined on a mesh. Figure 6.2 shows that the data structure of a field contains four layers: basic quantities, array, mesh and mapped field. Each of them is coded as a group of classes.

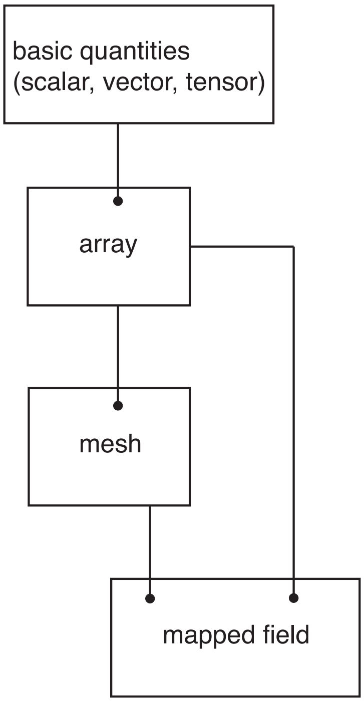


Figure 6.2: Data structure of field.

Basic quantities are the classes *Scalar*, *Vector* and *Tensor*. They are the implementations of the mathematical concepts bearing the same names. They contain 1,2 and 4 double precision float numbers as data members respectively and mathematical operations such as +,–,*,/, dot product, cross product and norm as member functions.

Array is a container that assembles these quantities. *Array* stores data on both host and device memory and manages the synchronisation between them so that a single class can be used for both host and device functions. Data is stored sequentially so that it is very efficient for developing grid based hydrodynamic solvers which accesses the data also sequentially. Similarly, we implemented another class *2DArray* to store basic quantities as a matrix.

The class *Mesh* contains the geometric and topological information of all the computational grids stored in units of *Arrays* or *2DArrays*. Two types of mesh are implemented in this software. The first one is a simplified implementation of half-edge mesh data structure to represent general unstructured grids, but in many applications the Cartesian grids are usually more convenient. Though the half-edge mesh data structure contains all the information Cartesian grids needed, it also contains some redundant information that can be removed. Therefore, another class *CartesianMesh* is implemented specifically for Cartesian grids through further simplifying the half-edge mesh data structure to avoid overhead. In the half-edge data structure, each of the cell is assigned an index. Each edge shared by two cells are divided into two half-edges. All the half-edges are assigned a global index. A local index starting from 1 marking its sequence in the cell is also assigned to each half-edge. The global indices of the half-edges in each cell are stored in a *2dArray* called *CellHalfedges*. Some half-edges lie outside the boundary, each of these half-edges is assigned another global index. Another *2dArray* called *CellNeighbours* is storing the neighbouring cells and boundaries. For the 'ith' edge of an arbitrary cell N, a pair of integers (k,M) is used to identify its adjacent cell. There are two possibilities here:

1) if the half-edge is adjacent to another cell, then M is used to store the index of the neighbour cell, k is used to store the local index of the half-edge of the neighbouring cell that is adjacent to cell 'N'.

2) if the half-edge lies on the boundary, k is set to 0 to denote that it is on the boundary because the local index of a half-edge of a cell is always bigger than 0 and M is used to store the global index of the boundary.

This pair of integers is stored as a structure in ith row of Nth column in *CellNeighbours*. An example of the half-edge data structure is illustrated in Figure 6.3. In this example, cell 0 has a neighbour denoted by (3,1), which means that it is adjacent to the 3rd half-edge of cell 1. Two edges of the cell 0 are at the boundary, which are denoted by (0,0) and (0,1). The centroid position and area of each cell are stored in Arrays following the sequence of cell indices. The outward normal direction and length of the half-edges are also stored in Arrays following the sequence of global indices of half-edges. For Cartesian grids, the index of the half-edge is arranged anticlockwise starting from the south edge. So that the normal directions of Cartesian cell edges always points to one of the four directions (south, east, north and west), the normal directions of cell edges is known once we know the half-edge index. Therefore the normal directions of cell edges is no longer stored in *CartesianMesh* to save some memory.

The class *MappedField* stores a pointer to *Mesh* or *CartesianMesh* so that operator function can query data in the mesh through this class. An array called *data* is included in *MappedField* to store the field values corresponds to the cells within the domain. These values are sequenced by the cell indices. *MappedField* also contains an array *BoundaryValues* that stores the field values at all the ghost cells at the boundary.

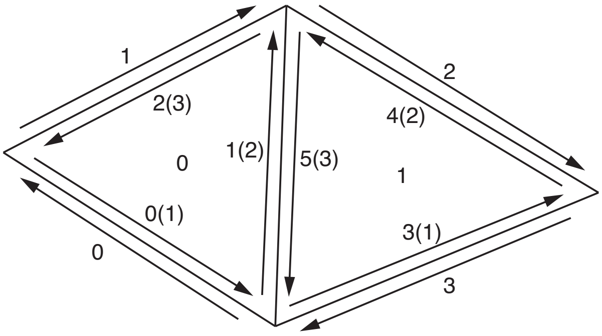


Figure 6.3: An illustrative example of the half-edge data structure.

2. Operators

The operators are functions that can be divided into two categories: algebraic operators and differential operators. Algebraic operators are those do not require information from neighbouring cells and its calculation follows a certain routine. Essentially, algebraic operations on fields are merely repetition of algebraic operation on its data members which are always compositions of the following two basic form of functions: $y = f(a)$ and $y = f(a,b)$. For example. $a^2 + b$ can be accomplished by compositing $f(a) = a^2$ and $f(a,b) = a + b$. Therefore, a way of implementing the algebraic operator functions is implementing two template functions which has either one or two fields as input arguments. One of the template parameter is a function type so that we only have to define $f(a)$ or $f(a,b)$ as an anonymous function (lambda in C++11) or a function pointer as the template parameter to decide which algebraic operation to do. Any algebraic operations can be achieved by compositing these two template functions. Our implementation adds great flexibility to the software because new algebraic operators can now be implemented easily.

Differential operators are those requiring information from neighbouring cells and boundaries. Generally, they loop over the neighbours to enquiry the data needed and then make some calculations. The computation process of differential operators are usually much more complex so that defining a specific routine for them is not feasible. Therefore each differential operator must be implemented separately.

Because the software uses GPUs to accelerate the computation, both algebraic and differential operators contain CUDA kernels as sub-functions to do the computations on GPUs. These kernels subtract raw pointers from the *field* class as arguments to access memory on GPUs.

3. Numerical solvers

Numerical solvers are implemented by combining *MappedFields* and operator functions. Variables in PDEs, for example water depth h in the SWEs, are defined as *MappedFields* in the beginning to be used by the operator functions. Then several operator functions are called following specific sequences to solve the PDEs.

6.5 Results and validation

In this section, two widely used benchmark tests are simulated to validate the proposed model. The results are compared with experimental results or field measurements as well as results by solving the traditional SWEs. The scheme for solving the SWEs is obtained by simply setting $1/\phi^2 = 1$ and ignoring the centrifugal force term. The difference between the numerical aspects of the SWEs model and the non-hydrostatic shallow flow model is therefore minimal, so that we can constrain the comparison solely to the difference between the governing equations.

6.5.1 *Dam-break wave over a triangular sill*

This test case is recommended by the EU IMPACT project to validate shallow flow models. The experimental set-up is shown in Figure 6.4. The water depth history was recorded with three gauges during the experiment. Details of this experiment can be found in Soares-Frazão [2007]. The Manning coefficient is set to 0.011 m/s$^{1/3}$, and a spatial resolution of 0.025 m was used for the simulations.

We compared the simulated water depth history at three gauges with the measured data at Figure 6.5. Because the dam-break waves are being reflected repeatedly between the triangular sill and the walls, multiple peaks appear in the water depth history plot. The SWEs model and the new non-hydrostatic shallow flow model give very similar results, and both of them agree in general with the experiment. The new non-hydrostatic shallow flow model however predicts the arriving time of the first peak in gauge 2 more accurately which may be due to the effect of non-hydrostatic pressure or curvature. This test case suggests that the new non-hydrostatic shallow flow model is as good as the SWEs model for simulating complex dam-break waves over irregular topography.

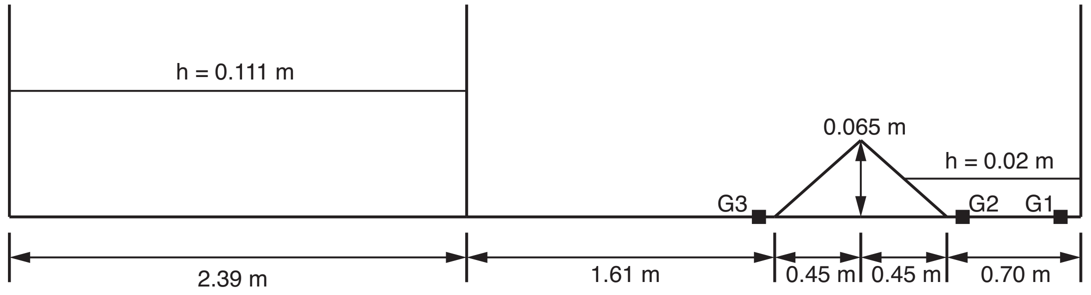


Figure 6.4: Initial configuration of experiment of dam-break wave over a triangular bottom sill.

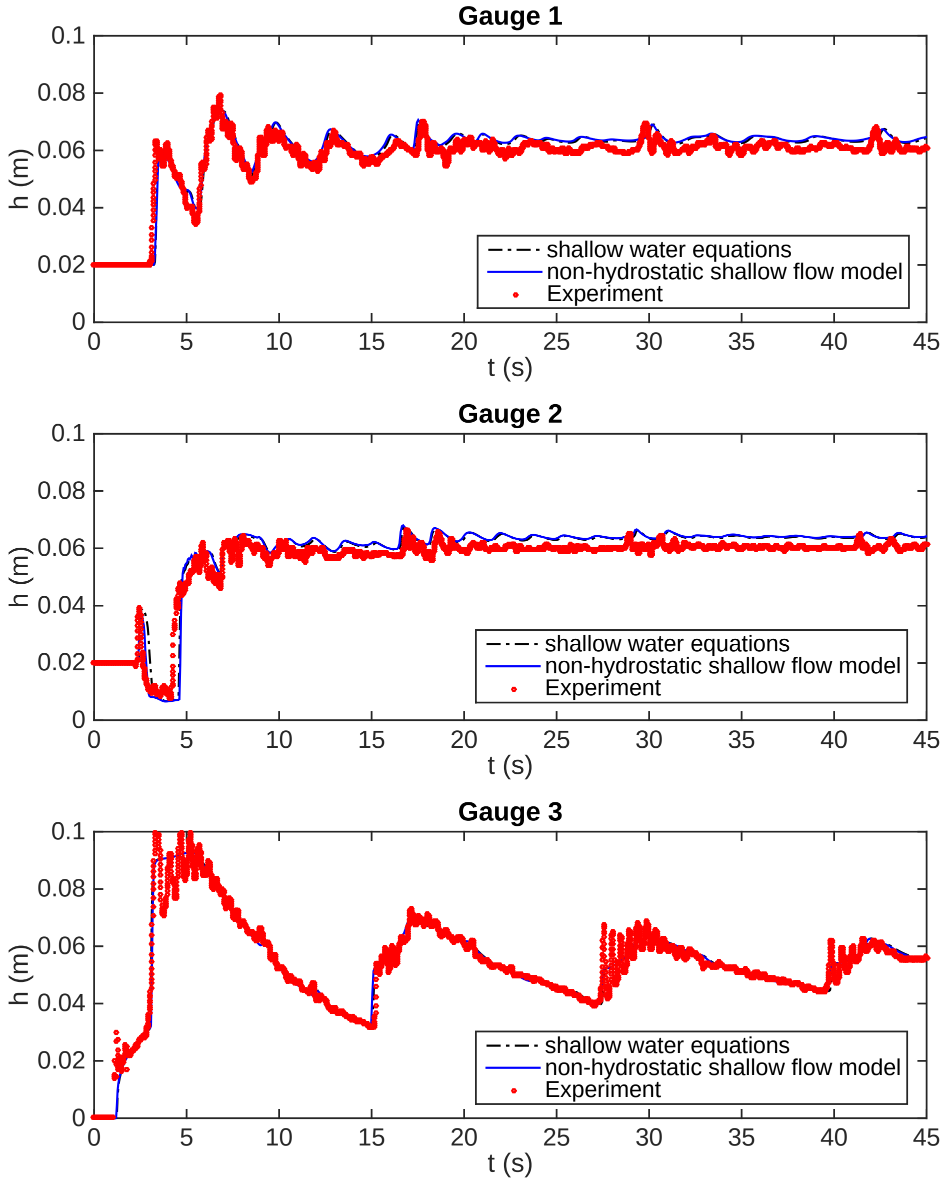


Figure 6.5: Dam-break wave over a triangular sill: comparing simulated water depth and experimental data at three gauges.

6.5.2 *Malpasset dam-break flood*

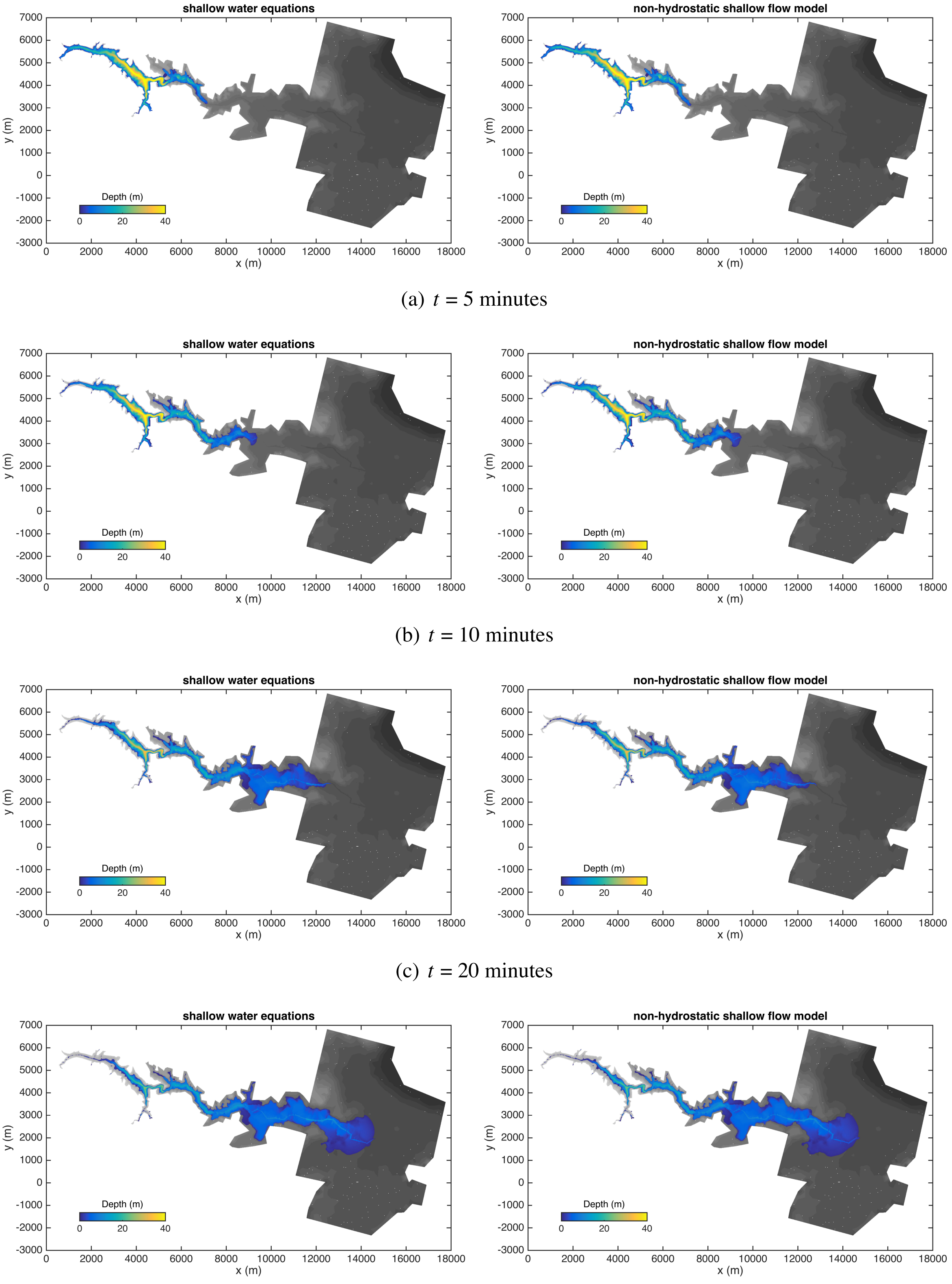
Herein, the Malpasset dam-break flood is simulated to investigate the new non-hydrostatic shallow flow model's performance on more realistic topography. The snapshots of flood extent are compared in Figure 6.6. The results produced by the new non-hydrostatic shallow flow model agree closely with the results by the SWEs model. The only noticeable difference is that the flood front moves slightly faster (about 100 m ahead) in the simulation by the new non-hydrostatic shallow flow model, probably due to the curvature effects. In terms of the maximum water level at the police survey points, we compared the predicted values with the measured data at Figure 6.7. The non-hydrostatic shallow flow model gives slightly better results, which is reflected in the RMSEs shown in Table 6.1. RMSE of the results of the non-hydrostatic shallow flow model is about 12% lower than that of the SWEs model. This test case confirms the current non-hydrostatic shallow flow model's capability for simulating flood waves on realistic topography. On a Tesla M2075 GPU, the SWEs model took 426 s to finish the whole simulations while the non-hydrostatic shallow flow model took 947 s, which is 2.2 times slower apparently due to the additional efforts to calculate non-hydrostatic pressure and centrifugal forces. The performance figures are comparable to another GPU-accelerated second-order Gudunov-type FV code mentioned in chapter 3 (Amouzgar et al. [2014]), which takes 330 s to finish the whole simulation.

Table 6.1: Malpasset dam-break flood: Comparison of RMSEs between the SWEs model and the non-hydrostatic shallow flow model.

shallow water equations	non-hydrostatic shallow flow model
3.65 m	3.23 m

6.6 Conclusions

In this chapter, the non-hydrostatic depth-averaged equations for flow-like landslides is combined with Manning's friction law to simulate flood waves. In order to ensure the accuracy and stability of the model, an Implicit-Explicit Runge-Kutta scheme is devised to discretise the friction term. The resulting model is applied to simulate two dam-break flood test cases and compared with a SWEs model with similar numerical method. Numerical results show that the current non-hydrostatic shallow flow model can give results as good as or slightly better than the SWEs model's results, which confirms that the non-hydrostatic depth-averaged equations derived in chapter 5 can be used to simulate water flow thus it can serve as a unified



(a) t = 5 minutes

(b) t = 10 minutes

(c) t = 20 minutes

(d) t = 30 minutes

Figure 6.6: Malpasset dam-break flood: inundation maps predicted by different models at different times.

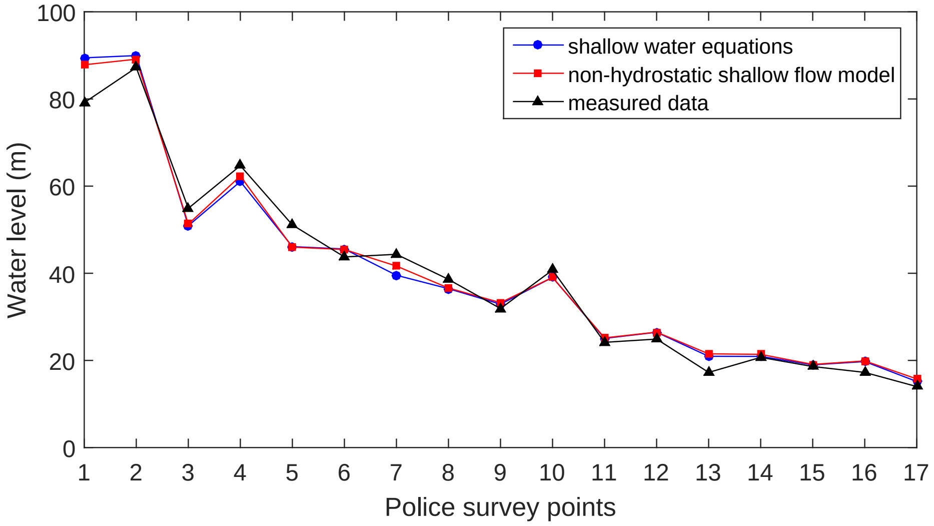


Figure 6.7: Malpasset dam-break flood: comparing simulated water level with police survey data.

mathematical framework for both water flows and mass flows.

In addition, a flexible and efficient software design is presented. The code is written in C++ and CUDA to benefit from the versatility provided by C++ and run on GPUs. The architecture of the code is designed by abstracting a common pipeline and the associated interfaces between the steps from the procedure of solving PDEs. The pipeline comprises several operator functions while fields of the PDEs becomes the interface between the operator functions. The operator functions can be combined to solve new PDEs and replaced without affecting other parts of the code, so that this software framework is flexible enough to implement new functionality or update existing code. Because the code can run on GPUs, it is highly efficient for large-scale simulations. Numerical test shows that the efficiency of the SWEs solver implemented in the current framework is comparable with another CUDA accelerated FV code for the Malpasset dam-break simulation, which suggests that a satisfactory efficiency has been obtained.

The mathematical formulation together with the software design presented in this chapter provide a multi-hazard modelling framework that enables the development towards a catchment scale physically-based multi-hazard model.

Chapter 7. Conclusions and future work

A physically-based catchment scale multi-hazard model based on the solution to the depth-averaged governing equations can be very useful for understanding the underlying physics of the water-related natural hazards, i.e. flash floods, landslides and debris flows, and quantifying their risks. This thesis developed the key simulation technologies that must be used for developing such a model. This chapter summarises these simulation technologies, draws key conclusions of the whole thesis and point out directions for the future development of the proposed multi-hazard model.

7.1 Conclusions

7.1.1 *The choice of numerical method*

FV and SPH represent the two major categories of numerical methods, i.e. Eulerian grid-based method and Lagrangian meshless method. This thesis first compared the FV and SPH numerical methods and assessed their suitability to support the development of the proposed multi-hazards model. The SPH was improved for solving the SWEs through deriving a corrected formulation that balances the bed slope source term and the internal pressure gradient term. Simulation of idealised benchmark tests shows that the corrected SPH formulation is more reliable than the standard formulation for simulating shallow flows over irregular topography. Then the new SPH-SWEs model is accelerated using GPUs and its performance is compared with a second-order Godunov-type FV model in the context of practical flood simulation. The Malpasset dam break and a hypothetic flood event in the Thamesmead floodplain were simulated. Generally, the SPH model may give results similar to those produced by the FV model if simulations are carried out at comparable spatial resolutions. However, the SPH model required much higher computational cost (about 50 times greater) to obtain comparable results. Because improved computational efficiency is crucial for the practical application of the proposed multi-hazards model, it is concluded that the FV model is more suitable for use in developing the proposed multi-hazard model.

7.1.2 *Simulation technologies for overland flows*

Overland flow plays a central role in the formation of and interactions between multiple hazards including flash floods, landslides and debris flows. This thesis presented a model for overland flow simulation, based on the solution to the fully 2D SWEs. The model was implemented with new simulation technologies that are crucial for accurate, stable and efficient simulation of overland flows.

1. A novel surface reconstruction method (SRM) for discretising bed slope source term.

Together with an HLLC Riemann solver, SRM was proposed to compute the bed slope terms correctly in the limit of vanishing water depth, predict correct discharge for flow on slopes even when friction is dominant, maintain stable simulations over discontinuous topographic features, and preserve the lake at rest solution. Several benchmark tests were considered and simulated to demonstrate the much improved numerical accuracy, stability and efficiency achieved by the SRM model, in comparison to other prevailing models implemented with the hydrostatic reconstruction method and augmented Riemann solver.

2. A novel implicit scheme for discretising friction source terms.

The friction terms were computed implicitly using a Newton-Raphson iteration method to effectively handle the strong non-linearity of the friction terms. Particularly, the new scheme can relax the flow velocity to the theoretical equilibrium state obtained by imposing balanced bed slope and friction terms in a single time step. This is an essential feature that a numerical scheme should have to ensure stable simulations with acceptable time steps when simulating overland flows with vanishing water depth and dominating friction effect.

7.1.3 *A high-performance modelling framework for multiple hazards*

The proposed modelling framework comprises the following two essential components:

1. A unified mathematical framework for simulating both water flows and mass flows

Using different coordinate systems makes it difficult to integrate a flow-like landslides model and a water flow model into a single framework. To eliminate this difficulty, we firstly developed a new non-hydrostatic depth-averaged model for flow-like landslides on a global Cartesian coordinate system with the following highlights:

1) The new equations take into account the non-hydrostatic vertical pressure and centrifugal acceleration due to bed curvature. The final governing equations are rotational invariant, hyperbolic and mathematically well-balanced to preserve the lake at rest solution.

2) A second-order Godunov-type finite volume numerical scheme is then implemented to solve the governing equations. To preserve the lake at rest solution and ensure non-negativity of depth for wetting and drying, the hydrostatic reconstruction method is used with necessary modifications made in the context of the new governing equations. A novel discretisation scheme was proposed for the friction terms to properly simulate the static resistance and stop condition. The new model is mathematically and numerically rigorous to support landslide simulations over complex real-world terrain.

Then we developed a new non-hydrostatic shallow flow model for flood simulations by combining the new depth-averaged model with the Manning's friction law. A new IMEX Runge-Kutta scheme was devised to ensure accuracy and stability of the model. Two benchmark tests suggested that the new non-hydrostatic shallow flow model performs as well as a traditional FV shock-capturing SWE model for simulating flood waves. This implies that the new depth-averaged model for flow-like landslides can also be used to simulate water flows, thus providing a unified mathematical framework for developing the proposed multi-hazard model.

2. A flexible and efficient software framework

A flexible and efficient software framework was designed, which is essential for developing the proposed multi-hazard model. The code is written in C++ except the CUDA components that are responsible for the computational intensive tasks. The versatility of C++ makes it possible to develop a flexible code and CUDA enables the use of GPUs to accelerate simulations. Because the proposed multi-hazard modelling system is essentially a set of PDEs solvers, we implemented mathematical units in PDEs, i.e. field and operator as basic components of the code. This brings an advantage in that any PDEs solver can be implemented by combining these basic components and thus the software is flexible in the sense that new functionality can be easily implemented. We compared a second-order Godunov-type FV SWEs solver and a non-hydrostatic shallow flow model implemented in the current software framework with another well tested second-order Godunov-type FV code for solving the SWEs. Numerical tests indicated that their efficiency was comparable, proving that the software has achieved the desired efficiency that can be expected from using GPUs. The software framework will become the infrastructure for the proposed multi-hazard model.

7.2 Future work

7.2.1 *Enable variation of basal friction*

In order to simulate different types of water-related natural hazards in an integrated model, the basal friction terms need to adapt to the change of sediment concentration. A pragmatic way of doing so is to use the quadratic rheology by Julien and Lan [1991] which combines frictional, turbulent and viscous effects in a single formulation. But research efforts are needed in the long term to understand the mechanics of transitions between different hazard types.

7.2.2 *Develop a sediment transport model*

The variation of basal friction depends on the change of sediment concentration so that a sediment transport model needs to be implemented. The sediment transport model needs to consider the sediment concentration change due to erosion, deposition and transport processes, which can be developed by solving a transport equation with some source and sink terms. One of the major challenges is the determination of erosion and deposition rate, which certainly worths more theoretical and experimental investigations though some existing formulae (e.g. Iverson and Ouyang [2015]) can be adopted as the first step.

7.2.3 *Develop an infiltration model*

Pore water pressure and groundwater table change due to infiltration determines the initiation of landslides. To model them, a simple and practical approach is modifying the Green-Ampt infiltration model (Mein and Larson [1973]) although coupling a groundwater model is a more comprehensive alternative.

7.2.4 *Develop a slope failure model*

Based on the computed pore water pressure and groundwater table, the slope stability can be determined by limit equilibrium analysis that is commonly used in geotechnical engineering.

Once the slope is identified as unstable, some certain volume of mass can be added to the hydrodynamic part of the model and the bed profile can be modified accordingly.

7.2.5 *Develop the multi-hazard modelling system*

This thesis and the aforementioned planned work will provide some components and technologies for developing the proposed multi-hazard modelling system. The remaining task is to synthesise these components and utilise these technologies to forge the catchment-scale physically-based multi-hazard modelling system that we desire.

Bibliography

Alcrudo, F. and Benkhaldoun, F. (2001). Exact solutions to the Riemann problem of the shallow water equations with a bottom step. *Computers & Fluids*, 30(6):643–671.

Amouzgar, R., Liang, Q., and Smith, L. (2014). A GPU-accelerated shallow flow model for tsunami simulations. *Proceedings of the Institution of Civil Engineers - Engineering and Computational Mechanics*, 167(3):117–125.

Ata, R. and Soulaïmani, A. (2005). A stabilized SPH method for inviscid shallow water flows. *International Journal for Numerical methods in Fluids*, 47:139–159.

Audusse, E., Bouchut, F., Bristeau, M.-O., Klein, R., and Perthame, B. (2004). A Fast and Stable Well-Balanced Scheme with Hydrostatic Reconstruction for Shallow Water Flows. *SIAM Journal on Scientific Computing*, 25(6):2050–2065.

Audusse, E. and Bristeau, M.-O. (2005). A well-balanced positivity preserving "second-order" scheme for shallow water flows on unstructured meshes. *Journal of Computational Physics*, 206(1):311–333.

Bates, P. D. and De Roo, A. P. J. (2000). A simple raster-based model for flood inundation simulation. *Journal of Hydrology*, 236(1-2):54–77.

Bates, P. D., Horritt, M. S., and Fewtrell, T. J. (2010). A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *Journal of Hydrology*, 387(1-2):33–45.

Benz, W. (1990). Smooth Particle Hydrodynamics: A Review. In Buchler, J. R., editor, *The Numerical Modelling of Nonlinear Stellar Pulsations: Problems and Prospects*, pages 269–288. Springer Netherlands, Dordrecht.

Berger, M. J., George, D. L., LeVeque, R. J., and Mandli, K. T. (2011). The GeoClaw software for depth-averaged flows with adaptive refinement. *Advances in Water Resources*, 34(9):1195–1206.

Bermúdez, A., Dervieux, A., Desideri, J.-A., and Vázquez, M. (1998). Upwind schemes for the two-dimensional shallow water equations with variable depth using unstructured meshes. *Computer Methods in Applied Mechanics and Engineering*, 155(1):49–72.

Bermudez, A. and Vazquez, M. E. (1994). Upwind methods for hyperbolic conservation laws with source terms. *Computers & Fluids*, 23(8):1049–1071.

Bouchut, F. and de Luna, T. (2010). A Subsonic-Well-Balanced Reconstruction Scheme for Shallow Water Flows. *SIAM Journal on Numerical Analysis*, 48(5):1733–1758.

Bouchut, F. and Westdickenberg, M. (2004). Gravity driven shallow water models for arbitrary topography. *Communications in Mathematical Sciences*, 2(3):359–389.

Bradbrook, K. F., Lane, S. N., Waller, S. G., and Bates, P. D. (2004). Two dimensional diffusion wave modelling of flood inundation using a simplified channel representation. *International Journal of River Basin Management*, 2(3):211–223.

Brodtkorb, A. R., Sætra, M. L., and Altinakar, M. (2012). Efficient shallow water simulations on GPUs: Implementation, visualization, verification, and validation. *Computers and Fluids*, 55:1–12.

Brufau, P., García-Navarro, P., and Vázquez-Cendón, M. E. (2004). Zero mass error using unsteady wetting–drying conditions in shallow flows over dry irregular topography. *International Journal for Numerical Methods in Fluids*, 45(10):1047–1082.

Burguete, J., García-Navarro, P., and Murillo, J. (2008). Friction term discretization and limitation to preserve stability and conservation in the 1D shallow-water model: Application to unsteady irrigation and river flow. *International Journal for Numerical Methods in Fluids*, 58(4):403–425.

Busaman, A., Mekchay, K., and Siripant, S. (2015). Dynamically adaptive tree grid modeling for simulation and visualization of rain-water overland flow. *International Journal for Numerical Methods in Fluids*, 79(11):559–579.

Castro-Orgaz, O., Hutter, K., Giraldez, J. V., and Hager, W. H. (2014). Nonhydrostatic granular flow over 3-D terrain: New Boussinesq-type gravity waves. *Journal of Geophysical Research: Earth Surface*, 120(1):1–28.

Caviedes-Voullième, D., García-Navarro, P., and Murillo, J. (2012). Influence of mesh structure on 2D full shallow water equations and SCS Curve Number simulation of rainfall/runoff events. *Journal of Hydrology*, 448-449:39–59.

Cea, L. and Blade, E. (2015). A simple and efficient unstructured finite volume scheme for solving the shallow water equations in overland flow applications. *Water Resources Research*, 51:5464–5486.

Cea, L., Garrido, M., and Puertas, J. (2010). Experimental validation of two-dimensional depth-averaged models for forecasting rainfall-runoff from precipitation data in urban areas. *Journal of Hydrology*, 382(1-4):88–102.

Cea, L. and Vazquez-Cendon, M. (2010). Unstructured finite volume discretization of two-dimensional depth-averaged shallowwater equations with porosity. *International Journal for Numerical Methods in Fluids*, 63:903–930.

Chen, G.-Q., Levermore, C. D., and Liu, T.-P. (1994). Hyperbolic conservation laws with stiff relaxation terms and entropy. *Communications on Pure and Applied Mathematics*, 47(6):787–830.

Chen, H. X., Zhang, S., Peng, M., and Zhang, L. M. (2016). A physically-based multi-hazard risk assessment platform for regional rainfall-induced slope failures and debris flows. *Engineering Geology*, 203:15–29.

Chow, V. T. (1959). *Open Channel Hydraulics*. McGraw-Hill Book Company.

Costabile, P., Costanzo, C., and Macchione, F. (2012). Comparative analysis of overland flow models using finite volume schemes. *Journal of Hydroinformatics*, 14(1):122.

Costabile, P., Costanzo, C., and MacChione, F. (2013). A storm event watershed model for surface runoff based on 2D fully dynamic wave equations. *Hydrological Processes*, 27(4):554–569.

Coussot, P., Laigle, D., Arattano, M., Deganutti, A. M., and Marchi, L. (1998). Direct determination of rheological characteristics of debris flow. *Journal of Hydraulic Engineering-Asce*, 124(8):865–868.

Dai, F., Lee, C., and Ngai, Y. (2002). Landslide risk assessment and management: an overview. *Engineering Geology*, 64(1):65–87.

De Leffe, M., Le Touzé, D., and Alessandrini, B. (2010). SPH modeling of shallow-water coastal flows. *Journal of Hydraulic Research*, 48(sup1):118–125.

Delannay, R., Valance, A., Mangeney, A., Roche, O., and Richard, P. (2017). Granular and particle-laden flows: from laboratory experiments to field observations. *Journal of Physics D: Applied Physics*, 50(5):053001.

Delestre, O., Cordier, S., Darboux, F., and James, F. (2012). A limitation of the hydrostatic reconstruction technique for Shallow Water equations. *Comptes Rendus Mathematique*, 350(13-14):677–681.

Deltares Open Software (2016). Delft-3D (https://oss.deltares.nl/web/delft3d).

Denlinger, R. P. and Iverson, R. M. (2001). Flow of variably fluidized granular masses across three-dimensional terrain: 2. Numerical predictions and experimental tests. *Journal of Geophysical Research*, 106(B1):553–566.

Denlinger, R. P. and Iverson, R. M. (2004). Granular avalanches across irregular three-dimensional terrain: 1. Theory and computation. *Journal of Geophysical Research*, 109(F1):F01014.

Di Giammarco, P., Todini, E., and Lamberti, P. (1996). A conservative finite elements approach to overland flow: the control volume finite element formulation. *Journal of Hydrology*, 175(1-4):267–291.

Domínguez, J. M., Crespo, A. J. C., and Gómez-Gesteira, M. (2013a). Optimization strategies for CPU and GPU implementations of a smoothed particle hydrodynamics method. *Computer Physics Communications*, 184(3):617–627.

Domínguez, J. M., Crespo, A. J. C., Valdez-Balderas, D., Rogers, B. D., and Gómez-Gesteira, M. (2013b). New multi-GPU implementation for smoothed particle hydrodynamics on heterogeneous clusters. *Computer Physics Communications*, 184(8):1848–1860.

Duran, A., Liang, Q., and Marche, F. (2013). On the well-balanced numerical discretization of shallow water equations on unstructured meshes. *Journal of Computational Physics*, 235:565–586.

Einfeldt, B., Munz, C. D., Roe, P. L., and Sj??green, B. (1991). On Godunov-type methods near low densities. *Journal of Computational Physics*, 92(2):273–295.

Erduran, K. S., Kutija, V., and Hewett, C. J. M. (2002). Performance of finite volume solutions to the shallow water equations with shock-capturing schemes. *International Journal for Numerical Methods in Fluids*, 40(10):1237–1273.

Federico, F. and Cesali, C. (2015). An energy-based approach to predict debris flow mobility and analyze empirical relationships. *Canadian Geotechnical Journal*, 52:2113–2133.

Fiedler, F. R. and Ramirez, J. a. (2000). A numerical method for simulating discontinuous shallow flow over an infiltrating surface. *International Journal for Numerical Methods in Fluids*, 32(2):219–240.

FLO-2D Software INC (2016). FLO-2D (http://stage.flo-2d.com/).

Fraccarollo, L., Capart, H., and Zech, Y. (2003). A Godunov method for the computation of erosional shallow water transients. *International Journal for Numerical Methods in Fluids*, 41(9):951–976.

Fraccarollo, L. and Toro, E. F. (1995). Experimental and numerical assessment of the shallow water model for two-dimensional dam-break type problems. *Journal of Hydraulic Research*, 33(6):843–864.

GDR Midi (2004). On dense granular flows. *The European physical journal. E, Soft matter*, 14(4):341–365.

George, D. L. (2008). Augmented Riemann solvers for the shallow water equations over variable topography with steady states and inundation. *Journal of Computational Physics*, 227(6):3089–3113.

George, D. L. and Iverson, R. M. (2014). A depth-averaged debris-flow model that includes the effects of evolving dilatancy. II. Numerical predictions and experimental tests. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 470(20130820).

Gingold, R. A. and Monaghan, J. J. (1977). Smoothed particle hydrodynamics-theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society*, 181:375–389.

Godunov, S. (1959). A Finite Difference Method for the Computation of Discontinuous Solutions of the Equations of Fluid Dynamics. *Math Sbornik*, 47:271–306.

Govindaraju, R. (1988). On the diffusion wave model for overland flow: 1. Solution for steep slopes. *Water Resources Research*, 24(5):734–744.

Gray, J. M. N. T. and Edwards, A. N. (2014). A depth-averaged μ(I)-rheology for shallow granular free-surface flows. *Journal of Fluid Mechanics*, 755:503–534.

Gray, J. M. N. T., Tai, Y.-C., and Noelle, S. (2003). Shock waves, dead zones and particle-free regions in rapid granular free-surface flows. *Journal of Fluid Mechanics*, 491:161–181.

Gray, J. M. N. T., Wieland, M., and Hutter, K. (1999). Gravity-driven free surface flow of granular avalanches over complex basal topography. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 455(November):1841–1874.

Greenberg, J. and Leroux, A. (1996a). A Well-Balanced Scheme for the Numerical Processing of Source Terms in Hyperbolic Equations. *SIAM Journal on Numerical Analysis*, 33(1):1–16.

Greenberg, J. M. and Leroux, A. Y. (1996b). A Well-Balanced Scheme for the Numerical Processing of Source Terms in Hyperbolic Equations.

Harten, A., Lax, P., and Leer, B. (1983). On Upstream Differencing and Godunov-Type Schemes for Hyperbolic Conservation Laws. *SIAM Review*, 25(1):35–61.

Henderson, F. M. and Wooding, R. A. (1964). Overland flow and groundwater flow from a steady rainfall of finite duration. *Journal of Geophysical Research*, 69(8):1531–1540.

Hérault, A., Bilotta, G., and Dalrymple, R. A. (2010). SPH on GPU with CUDA. *Journal of Hydraulic Research*, 48(sup1):74–79.

Hergarten, S. and Robl, J. (2015). Modelling rapid mass movements using the shallow water equations in Cartesian coordinates. *Natural Hazards and Earth System Science*, 15(3):671–685.

Hou, J., Liang, Q., Simons, F., and Hinkelmann, R. (2013a). A 2D well-balanced shallow flow model for unstructured grids with novel slope source term treatment. *Advances in Water Resources*, 52:107–131.

Hou, J., Liang, Q., Simons, F., and Hinkelmann, R. (2013b). A stable 2D unstructured shallow flow model for simulations of wetting and drying over rough terrains. *Computers & Fluids*, 82:132–147.

Hou, J., Simons, F., Liang, Q., and Hinkelmann, R. (2014). An improved hydrostatic reconstruction method for shallow water model. *Journal of Hydraulic Research*, 52(3):432–439.

Huang, Y., Zhang, W., Xu, Q., Xie, P., and Hao, L. (2012). Run-out analysis of flow-like landslides triggered by the Ms 8.0 2008 Wenchuan earthquake using smoothed particle hydrodynamics. *Landslides*, 9(2):275–283.

Hungr, O. (1995). A model for the runout analysis of rapid flow slides, debris flows, and avalanches. *Canadian Geotechnical Journal*, 32:610–623.

Hunter, N. M., Horritt, M. S., Bates, P. D., Wilson, M. D., and Werner, M. G. F. (2005). An adaptive time step solution for raster-based storage cell modelling of floodplain inundation. *Advances in Water Resources*, 28(9):975–991.

Hutchinson, J. (1986). A sliding-consolidation model for flow slides. *Canadian Geotechnical Journal*, 23:115–126.

IPCC (2014). Climate Change 2014: Impacts, Vulnerability and Adaptation. Technical report.

Iverson, R. M. (1997). The physics of debris flows. *Reviews of Geophysics*, 35(3):245–296.

Iverson, R. M. (2014). Debris flows: Behaviour and hazard assessment. *Geology Today*, 30(1):15–20.

Iverson, R. M. and Denlinger, R. P. (2001). Flow of variably fluidized granular masses across three-dimensional terrain: 1. Columb mixture theory. *Journal of Geophysical Research*, 106(B1):537–552.

Iverson, R. M. and George, D. L. (2014). A depth-averaged debris-flow model that includes the effects of evolving dilatancy . I . Physical basis. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 470(2170).

Iverson, R. M. and Ouyang, C. (2015). Entrainment of bed material by Earth-surface mass flows: Review and reformulation of depth-integrated theory. *Reviews of Geophysics*, 53(1).

Jakob, M., Hungr, O., and Jakob, D. (2005). *Debris-flow hazards and related phenomena*. Springer-Verlag Berlin Heidelberg.

JBA Consulting (2016). JFlow (http://www.jflow.co.uk/).

Jin, S. (2012). Asymptotic preserving (ap) schemes for multiscale kinetic and hyperbolic equations: a review. *Rivista di Matematica della Università di Parma*, 3:177–216.

Jop, P., Forterre, Y., and Pouliquen, O. (2006). A constitutive law for dense granular flows. *Nature*, 441(June):727–730.

Juez, C., Murillo, J., and García-Navarro, P. (2013). 2D simulation of granular flow over irregular steep slopes using global and local coordinates. *Journal of Computational Physics*, 255:166–204.

Julien, P. and Lan, Y. (1991). Rheology of Hyperconcentrations. *Journal of Hydraulic Engineering*, 117(3):346–353.

Kappes, M. S., Keiler, M., von Elverfeldt, K., and Glade, T. (2012). Challenges of analyzing multi-hazard risk: A review. *Natural Hazards*, 64(2):1925–1958.

Kawahara, M. and Umetsu, T. (1986). Finite element method for moving boundary problems in river flow. *International Journal for Numerical Methods in Fluids*, 6(6):365–386.

Kelfoun, K. (2011). Suitability of simple rheological laws for the numerical simulation of dense pyroclastic flows and long-runout volcanic avalanches. *Journal of Geophysical Research: Solid Earth*, 116(8):1–14.

Kelfoun, K. and Druitt, T. H. (2005). Numerical modeling of the emplacement of Socompa rock avalanche, Chile. *Journal of Geophysical Research: Solid Earth*, 110(12):1–13.

Kuo, C. Y., Tai, Y. C., Bouchut, F., Mangeney, a., Pelanti, M., Chen, R. F., and Chang, K. J. (2009). Simulation of Tsaoling landslide, Taiwan, based on Saint Venant equations over general topography. *Engineering Geology*, 104(3-4):181–189.

Lagrée, P., Staron, L., and Popinet, S. (2011). The granular column collapse as a continuum : validity of a two-dimensional Navier-Stokes model with a μ(I)-rheology. *Journal of Fluid Mechanics*, 686:378–408.

Laigle, D. and Coussot, P. (1997). Numerical Modelling of Mudflows. *Journal of Hydraulic Engineering*, 123(July):617–623.

Lee, H. and Han, S. (2010). Solving the Shallow Water equations using 2D SPH particles for interactive applications. *The Visual Computer*, 26(6-8):865–872.

LeFloch, P. G. and Thanh, M. D. (2007). The Riemann problem for the shallow water equations with discontinuous topography. *Communications in Mathematical Sciences*, 5(4):865–885.

LeFloch, P. G. and Thanh, M. D. (2011). A Godunov-type method for the shallow water equations with discontinuous topography in the resonant regime. *Journal of Computational Physics*, 230(20):7631–7660.

Legros, F. (2002). The mobility of long-runout landslides. *Engineering Geology*, 63(3-4):301–331.

LeVeque, R. J. (1997). Wave Propagation Algorithms for Multidimensional Hyperbolic Systems. *Journal of Computational Physics*, 131(2):327–353.

LeVeque, R. J. (1998). Balancing Source Terms and Flux Gradients in High-Resolution Godunov Methods: The Quasi-Steady Wave-Propagation Algorithm. *Journal of Computational Physics*, 146(1):346–365.

Liang, D., Falconer, R. a., and Lin, B. (2006). Improved numerical modelling of estuarine flows. *Proceedings of the ICE - Maritime Engineering*, 159(March):25–35.

Liang, D., Lin, B., and Falconer, R. A. (2007). Simulation of rapidly varying flow using an efficient TVD–MacCormack scheme. *International Journal for Numerical Methods in Fluids*, 53(5):811–826.

Liang, D., Özgen, I., Hinkelmann, R., Xiao, Y., and Chen, J. M. (2015a). Shallow water simulation of overland flows in idealised catchments. *Environmental Earth Sciences*, 74(11):7307–7318.

Liang, Q. (2010). Flood simulation using a well-balanced shallow flow model. *Journal of Hydraulic Engineering*, 136:669–675.

Liang, Q. and Borthwick, A. G. L. (2009). Adaptive quadtree simulation of shallow flows with wet–dry fronts over complex topography. *Computers & Fluids*, 38(2):221–234.

Liang, Q., Hou, J., and Amouzgar, R. (2015b). Simulation of Tsunami Propagation Using Adaptive Cartesian Grids. *Coastal Engineering Journal*, 57(04):1550016.

Liang, Q. and Marche, F. (2009). Numerical resolution of well-balanced shallow water equations with complex source terms. *Advances in Water Resources*, 32(6):873–884.

Libersky, L. D., Petschek, A. G., Carney, T. C., Hipp, J. R., and Allahdadi, F. A. (1993). High Strain Lagrangian Hydrodynamics. *Journal of Computational Physics*, 109(1):67–75.

Lighthill, M. and Whitham, G. (1955). On Kinematic Waves . I . Flood Movement in Long Rivers. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 229(1178):281–316.

Liu, G. R. and Liu, M. B. (2003). *Smoothed particle hydrodynamics: a meshfree particle method*. World Scientific Publishing Company.

Liu, Q. Q., Chen, L., Li, J. C., and Singh, V. P. (2004). Two-dimensional kinematic wave model of overland-flow. *Journal of Hydrology*, 291(1-2):28–41.

Liu, T.-P. (1987). Hyperbolic conservation laws with relaxation. *Communications in Mathematical Physics*, 108(1):153–175.

Lucas, A., Mangeney, A., and Ampuero, J. P. (2014). Frictional velocity-weakening in landslides on Earth and on other planetary bodies. *Nature communications*, 5:3417.

Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal*, 82:1013–1024.

Major, J. J. and Iverson, R. M. (1999). Debris-flow deposition: Effects of pore-fluid pressure and friction concentrated at flow margins. *Bulletin of the Geological Society of America*, 111(10):1424–1434.

Mangeney, A., Bouchut, F., Thomas, N., Vilotte, J. P., and Bristeau, M. O. (2007). Numerical modeling of self-channeling granular flows and of their levee-channel deposits. *Journal of Geophysical Research: Earth Surface*, 112(F02017):1–21.

Mangeney-Castelnau, A., Bouchut, F., Vilotte, J. P., Lajeunesse, E., Aubertin, A., and Pirulli, M. (2005). On the use of Saint Venant equations to simulate the spreading of a granular mass. *Journal of Geophysical Research B: Solid Earth*, 110(B09103):1–17.

Marche, F., Bonneton, P., Fabrie, P., and Seguin, N. (2007). Evaluation of well-balanced bore-capturing schemes for 2D wetting and drying processes. *International Journal for Numerical Methods in Fluids*, 53(5):867–894.

Mcdougall, S. and Hungr, O. (2004). A model for the analysis of rapid landslide motion across three-dimensional terrain. *Canadian Geotechnical Journal*, 41(6):1084–1097.

Medina, V., Hürlimann, M., and Bateman, A. (2008). Application of FLATModel, a 2D finite volume code, to debris flows in the northeastern part of the Iberian Peninsula. *Landslides*, 5(1):127–142.

Mein, R. G. and Larson, C. L. (1973). Modeling infiltration during a steady rain. *Water Resources Research*, 9(2):384–394.

Monaghan, J. J. (1994). Simulating free surface flows with SPH. *Journal of Computational Physics*, 110(2):399–406.

Monaghan, J. J. (1997). SPH and Riemann Solvers. *Journal of Computational Physics*, 136(2):298–307.

Monaghan, J. J. and Kocharyan, A. (1995). SPH simulation of multi-phase flow. *Computer Physics Communications*, 87(1):225–235.

Morales De Luna, T., Castro Díaz, M. J., and Parés, C. (2013). Reliability of first order numerical schemes for solving shallow water system over abrupt topography. *Applied Mathematics and Computation*, 219(17):9012–9032.

Morris, E. M. and Woolhiser, D. A. (1980). Unsteady one-dimensional flow over a plane: Partial equilibrium and recession hydrographs. *Water Resources Research*, 16(2):355–360.

Murillo, J. and García-Navarro, P. (2010). Weak solutions for partial differential equations with source terms: Application to the shallow water equations. *Journal of Computational Physics*, 229(11):4327–4368.

Murillo, J., Garcia-Navarro, P., Burguete, J., and Brufau, P. (2007). The influence of source terms on stability, accuracy and conservation in two-dimensional shallowflow simulation using triangular finite volumes. *International Journal for Numerical Methods in Fluids*, 54:543–590.

Noelle, S., Pankratz, N., Puppo, G., and Natvig, J. R. (2006). Well-balanced finite volume schemes of arbitrary order of accuracy for shallow water flows. *Journal of Computational Physics*, 213(2):474–499.

NVIDIA (2015). CUDA C Programming Guide (https://docs.nvidia.com/cuda/cuda-c-programming-guide/).

Ouyang, C., He, S., and Tang, C. (2014). Numerical analysis of dynamics of debris flow over erodible beds in Wenchuan earthquake-induced area. *Engineering Geology*.

Pareschi, L. and Russo, G. (2005). Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. *Journal of Scientific Computing*, 25(1):129–155.

Pastor, M., Blanc, T., Haddad, B., Drempetic, V., Morles, M. S., Dutto, P., Stickle, M. M., Mira, P., and Merodo, J. a. F. (2015). Depth Averaged Models for Fast Landslide Propagation: Mathematical, Rheological and Numerical Aspects. *Archives of Computational Methods in Engineering*, 22(1):67–104.

Pastor, M., Blanc, T., Haddad, B., Petrone, S., Sanchez Morles, M., Drempetic, V., Issler, D., Crosta, G. B., Cascini, L., Sorbino, G., and Cuomo, S. (2014). Application of a SPH depth-integrated model to landslide run-out analysis. *Landslides*, 11(5):793–812.

Pastor, M., Haddad, B., Sorbino, G., Cuomo, S., and Drempetic, V. (2009). A depth-integrated, coupled SPH model for flow-like landslides and related phenomena. *International Journal for Numerical and Analytical Methods in Geomechanics*, 33(2):143–172.

Pitman, E. B. and Le, L. (2005). A two-fluid model for avalanche and debris flows. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 363(1832):1573–1601.

Ponce, V. M., Simons, D. B., and Li, R.-M. (1978). Applicability of kinematic and diffusion models. *Journal of the Hydraulics Division*, 104(3):353–360.

Pouliquen, O. and Forterre, Y. (2002). Friction law for dense granular flows: application to the motion of a mass down a rough inclined plane. *Journal of Fluid Mechanics*, 453:133–151.

Pudasaini, S. P. and Hutter, K. (2003). Rapid shear flows of dry granular masses down curved and twisted channels. *Journal of Fluid Mechanics*, 495:193–208.

Pudasaini, S. P., Hutter, K., Hsiau, S.-S., Tai, S.-C., Wang, Y., and Katzenbach, R. (2007). Rapid flow of dry granular materials down inclined chutes impinging on rigid walls. *Physics of Fluids*, 19(5):053302.

Pudasaini, S. P., Wang, Y., and Hutter, K. (2005). Rapid motions of free-surface avalanches down curved and twisted channels and their numerical simulation. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 363(1832):1551–1571.

Quinlan, N. J., Basa, M., and Lastiwka, M. (2006). Truncation error in mesh-free particle methods. *International Journal for Numerical Methods in Engineering*, 66(13):2064–2085.

Rodriguez-Paz, M. and Bonet, J. (2005). A corrected smooth particle hydrodynamics formulation of the shallow-water equations. *Computers & Structures*, 83(17):1396–1410.

Roe, P. (1981). Approximate Riemann Solvers , Parameter and Difference Schemes. *Journal of Computational Physics*, 372:357–372.

Rogers, B. D., Borthwick, A. G. L., and Taylor, P. H. (2003). Mathematical balancing of flux gradient and source terms prior to using Roe's approximate Riemann solver. *Journal of Computational Physics*, 192(2):422–451.

Rousseau, M., Cerdan, O., Delestre, O., and Dupros, F. (2015). Overland flow modelling with the Shallow Water Equation using a well balanced numerical scheme: Better predictions or just more complexity? *Journal of Hydrologic Engineering*, 20(10):04015012.

Sanders, B. F., Schubert, J. E., and Gallegos, H. A. (2008). Integral formulation of shallow-water equations with anisotropic porosity for urban flood modeling. *Journal of Hydrology*, 362(1-2):19–38.

Sanders, J. and Kandrot, E. (2010). *CUDA by Example: An Introduction to General-Purpose GPU Programming*. NVIDIA.

Sassa, K., Nagai, O., Solidum, R., Yamazaki, Y., and Ohta, H. (2010). An integrated model simulating the initiation and motion of earthquake and rain induced rapid landslides and its application to the 2006 Leyte landslide. *Landslides*, 7:219–236.

Savage, S. B. and Hutter, K. (1989). The motion of a finite mass of granular material down a rough incline. *Journal of Fluid Mechanics*, 199:177–215.

Sene, K. (2013). *Flash floods: forecasting and warning*. Springer.

Silbert, L. E., Landry, J. W., and Grest, G. S. (2003). Granular flow down a rough inclined plane: Transition between thin and thick piles. *Physics of Fluids*, 15(1):1–10.

Simons, F., Busse, T., Hou, J., Özgen, I., and Hinkelmann, R. (2014). A model for overland flow and associated processes within the Hydroinformatics Modelling System. *Journal of Hydroinformatics*, 16:375–391.

Singh, J., Altinakar, M. S., and Ding, Y. (2015). Numerical Modeling of Rainfall-Generated Overland Flow Using Nonlinear Shallow-Water Equations. *Journal of Hydrologic Engineering*, 20(8):04014089.

Singh, V. P. (2002). Is hydrology kinematic? *Hydrological Processes*, 16(3):667–716.

Smith, A. A. (1980). A generalized approach to kinematic flood routing. *Journal of Hydrology*, 45(1):71–89.

Smith, L. and Liang, Q. (2013). Towards a generalised GPU/CPU shallow-flow modelling tool. *Computers & Fluids*, 88:17–20.

Soares-Frazão, S. (2007). Experiments of dam-break wave over a triangular bottom sill. *Journal of Hydraulic Research*, 45(sup1):19–26.

Solenthaler, B., Bucher, P., Chentanez, N., Müller, M., and Gross, M. (2011). SPH Based Shallow Water Simulation. In *Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS (2011)*.

Song, L., Zhou, J., Guo, J., Zou, Q., and Liu, Y. (2011a). A robust well-balanced finite volume model for shallow water flows with wetting and drying over irregular terrain. *Advances in Water Resources*, 34(7):915–932.

Song, L., Zhou, J., Li, Q., Yang, X., and Zhang, Y. (2011b). An unstructured finite volumemodel for dam-break floods with wet/dry fronts over complex topography. *International Journal for Numerical Methods in Fluids*, 67:960–980.

Springel, V. (2001). GADGET: a code for collisionless and gasdynamical cosmological simulations. *New Astronomy*, 6:79–117.

Staron, L., Lagrée, P., and Popinet, S. (2012). The granular silo as a continuum plastic flow : The hour-glass vs the clepsydra. *Physics of Fluids*, 24:103301.

Stoker, J. J. (1958). Long Waves in Shallow Water. In *Water Waves*, pages 289–449. John Wiley & Sons, Inc.

Takahashi, T. (2014). *Debris Flow: Mechanics, Prediction and Countermeasures*. CRC Press, Taylor & Francis, 2nd edition.

Telemac Software (2016). Telemac 2D (http://www.opentelemac.org/index.php/modules-list/17-telemac-2d-presentation).

Testa, G., Zuccalà, D., Alcrudo, F., Mulet, J., and Soares-Frazão, S. (2007). Flash flood flow experiment in a simplified urban district. *Journal of Hydraulic Research*, 45(sup1):37–44.

Teyssier, R. (2014). Grid-Based Hydrodynamics in Astrophysical Fluid Flows. *Annual Review of Astronomy and Astrophysics*, 53:325–364.

Toro, E. F. (2001). *Shock-capturing methods for free-surface shallow flow*. John Wiley & Sons, Ltd.

Toro, E. F., Spruce, M., and Speares, W. (1994). Restoration of the contact surface in the HLL-Riemann solver. *Shock Waves*, 4(1):25–34.

Utili, S., Zhao, T., and Houlsby, G. T. (2015). 3D DEM investigation of granular column collapse: Evaluation of debris motion and its destructive power. *Engineering Geology*, 186:3–16.

Vacondio, R., Rogers, B. D., and Stansby, P. K. (2012a). Smoothed Particle Hydrodynamics: Approximate zero-consistent 2-D boundary conditions and still shallow-water tests. *International Journal for Numerical Methods in Fluids*, 69:226–253.

Vacondio, R., Rogers, B. D., Stansby, P. K., and Mignosa, P. (2012b). Accurate particle splitting for smoothed particle hydrodynamics in shallow water with shock capturing. *International Journal for Numerical Methods in Fluids*, 69(8):1377–1410.

Vacondio, R., Rogers, B. D., Stansby, P. K., and Mignosa, P. (2013). A correction for balancing discontinuous bed slopes in two-dimensional smoothed particle hydrodynamics shallow water modeling. *International Journal for Numerical Methods in Fluids*, 71(7):850–872.

van Leer, B. (1979). Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J. Comput. Phys.*, 32(1):101–136.

Van Wyk De Vries, B., Self, S., Francis, P. W., and Keszthelyi, L. (2001). A gravitational spreading origin for the Socompa debris avalanche. *Journal of Volcanology and Geothermal Research*, 105(3):225–247.

Varnes, D. (1978). Slope movement types and processes. In: Eckel EB (ed) Landslides and engineering practice, special report 28. Technical report, National Academy of Sciences, Washington, DC.

Vieira, J. H. D. (1983). Conditions governing the use of approximations for the Saint-Vénant equations for shallow surface water flow. *Journal of Hydrology*, 60(1-4):43–58.

Voellmy (1955). Über die Zerstörungskraft von Lawinen. *Schweiz Bauztg*, 73:212–285.

Wang, F. and Sassa, K. (2010). Landslide simulation by a geotechnical model combined with a model for apparent friction change. *Physics and Chemistry of the Earth*, 35(3-5):149–161.

Wang, Y., Liang, Q., Kesserwani, G., and Hall, J. W. (2011). A 2D shallow flow model for practical dam-break simulations. *Journal of Hydraulic Research*, 49(3):307–316.

Wang, Z. and Shen, H. T. (1999). Lagrangian simulation of one-dimensional dam-break flow. *Journal of Hydraulic Engineering*, 125(11):1217–1220.

Woolhiser, D. A. and Liggett, J. A. (1967). Unsteady, one-dimensional flow over a plane—The rising hydrograph. *Water Resources Research*, 3(3):753–771.

Xia, X., Huang, R., Liang, Q., and Yu, B. (2015). A New Physically-Based Simulation Framework for Modelling Flow-Like Landslides. *International Journal of Geohazards and Environment*, 1(3):94–100.

Xia, X. and Liang, Q. (2016). A GPU-accelerated smoothed particle hydrodynamics (SPH) model for the shallow water equations. *Environmental Modelling and Software*, 75:28–43.

Yano, K. and Daido, A. (1965). Fundamental study on mud-flow. *Bulletin of the Disaster Prevention Research Institute*, 14(2):69–83.

Yu, C. and Duan, J. (2014). Two-Dimensional Hydrodynamic Model for Surface-Flow Routing. 140(9):1–13.

Zhai, J., Yuan, L., Liu, W., and Zhang, X. (2015). Solving the Savage–Hutter equations for granular avalanche flows with a second-order Godunov type method on GPU. *International Journal for Numerical Methods in Fluids*, 77:381–399.

Zhou, G. G. D. and Ng, C. W. W. (2010). Numerical investigation of reverse segregation in debris flows by DEM. *Granular Matter*, 12(5):507–516.

Zhou, J. G., Causon, D. M., Mingham, C. G., and Ingram, D. M. (2001). The surface gradient method for the treatment of source terms in the shallow-water equations. 168:1–25.

Zienkiewicz, O. C., Chan, a. H. C., Pastor, M., Paul, D. K., and Shiomi, T. (1990). Static and Dynamic Behaviour of Soils: A Rational Approach to Quantitative Solutions. I. Fully Saturated Problems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 429(1877):285–309.