University of Southampton

Faculty of Engineering and Physical Science
Department of Civil, Maritime & Environmental Engineering

High-performance numerical models for large scale flood inundation problems

by
Alireza Shamkhalchian

Thesis for the degree of **Doctor of Philosophy**

January 2021

University of Southampton

Abstract

FACULTY OF ENGINEERING AND PHYSICAL SCIENCES

Department of Civil, Maritime & Environmental Engineering

Thesis for the degree of Doctor of Philosophy

HIGH-PERFORMANCE NUMERICAL MODELS FOR LARGE SCALE FLOOD INUNDATION PROBLEMS

by

Alireza Shamkhalchian

The simulation of large-scale flood inundation problems requires numerical models that are both accurate and have high computational performance. Over the past decades giant achievements have been attained in computational hydraulics and a number of accurate models have emerged. However, the accurate simulations of these models require high-resolution meshes, resulting in high computational cost. Thus, the models providing a rational trade-off between accuracy and time cost are highly demanded. The lack of such tools substantially limits our ability to understand and predict floods that extend over large areas. In this research a new model is presented, which falls into this line. The model solves the two dimensional shallow water equations (SWE) by a Godunov type finite volume (FV) method that makes use of two nested meshes. Runtime computations are performed at a coarse computational mesh (thus increasing computational performance), while a fine mesh is used to incorporate finely resolved information into the solution at pre-processing level (thus substantially reducing the loss of accuracy that would otherwise result). The proposed model presents new upscaling methods that are separately derived for each of the terms in the SWE based on the integration of the governing equations over subdomains defined by the coarse resolution grid cells, such as friction and bed slope source terms. This research also sheds light onto the limitations of some of the widely used assumptions in nested meshes models, such as constant friction slope for upscaling the friction effects. The accuracy and performance of the model are tested through six artificial and a real-world test problems, namely one-dimensional steady flow over a hump, dam-break problem, dam-break wave propagating over three humps, open channel flow around a bend, 1D flow over an irregular bed, compound channel, and the River Tiber. These test cases cover a wide range of hydraulic issues, such as steady and unsteady flows, hydraulic jump, 1D and 2D flood propagation, (mild, large or no) variation of topography and roughness coefficient across the domain of study, propagation of waves with continuous (rarefaction) and discontinuous (shock) water surface. It is shown that that *i*) in general the proposed model provides more accurate results than traditional

FV models currently available and *ii*) for the same accuracy and at low resolution, the proposed methods substantially increase computational performance. These attributions make the proposed model suitable for large-scale real-life problems and those artificial cases with significant changes in topography's features and roughness properties.

Contents

List of Tables

List of Figures

Research Thesis: Declaration of Authorship

Print name: ALIREZA SHAMKHALCHIAN

Title of thesis: High-performance numerical models for large scale flood inundation problems

I declare that this thesis and the work presented in it are my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;
4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Some parts of this work have previously submitted for a number of modules in the University of Southampton as a part of my iPhD course, namely, iPhD project (FEEG6015), 9 month report, and Transfer report.

Signature: Alireza Shamkhalchian Date:January 6, 2021

Acknowledgements

First, I would like to express my sincere gratitude to my supervisor, Dr. Gustavo de Almeida. I greatly appreciate his time, sharp perspective and valuable comments. He patiently encouraged and supported me. This study was funded by the UK Engineering and Physical Sciences Research Council (EPSRC) through the Centre for Doctoral Training in Sustainable Infrastructure Systems (CDT-SIS), grant EP/L01582X/1 and Jacobs Engineering Group. The work made extensive use of the University of Southampton super-computing resources Iridis 4 and Iridis 5. I deeply acknowledge the Flood Modeller developer team of Jacobs Engineering Group—especially Dr Jon Wicks and Dr Konrad Adams—for their constructive comments on how to improve the model with a focus on real-world applications. I would also like to thank Dr Mario Morales-Hernández for sharing data on the flood event of the River Tiber, which formed the basis of a real world test case in this thesis. I would especially like to thank my wife Mahboobeh, whose help, devotion, support, encouragement, kindness and patience are unforgettable. The help and support of my parents are undeniable. I deeply appreciate them.

My highest appreciation belongs to God, who is the source of all kindness, help, support and patience.

Definitions and Abbreviations

Notation

a_f	frontal width of obstructions projected to flow path $\left(\mathrm{m}^{-1}\right)$
A	cross-section area $\left(\mathrm{m}^2\right)$
$b(x,y)$	bed function (m)
$\boldsymbol{B}$	channel width (m)
c	Chézy's coefficient $\left(\mathrm{m}^{\frac{1}{2}}\ \mathrm{s}^{-1}\right)$
$\boldsymbol{C}_D^b$	sub-grid scale flow obstacles' drag coefficient (-)
$\boldsymbol{C}_D^f$	bed friction coefficient (-)
$\boldsymbol{C}_D^0$	dimensionless obstruction drag coefficient (-)
CFL	Courant–Friedrichs–Lewy condition (-)
e_x, e_y	x and y components of unit vector normal to cell's boundary (-)
$\boldsymbol{e}$	unit vector normal to cell's boundary (-,-)
$\|E\|$	absolute errors (m)
$\boldsymbol{E}$	flux tensor $\left(\mathrm{m}^2\ \mathrm{s}^{-1}, \mathrm{m}^3\ \mathrm{s}^{-2}, \mathrm{m}^3\ \mathrm{s}^{-2}\ ;\ \mathrm{m}^2\ \mathrm{s}^{-1}, \mathrm{m}^3\ \mathrm{s}^{-2}, \mathrm{m}^3\ \mathrm{s}^{-2}\right)$
$\boldsymbol{E}_*$	flux vector in Eq. 2.5 $\left(\mathrm{m}^2\ \mathrm{s}^{-1}, \mathrm{m}^3\ \mathrm{s}^{-2}, \mathrm{m}^3\ \mathrm{s}^{-2}\right)$
E_G	mass exchange term between the mobile and stagnant region defined by Guinot (2012) $\left(\mathrm{m\ s}^{-1}\right)$
E_e	estimated fractional error (%)
E_p	error percentage (%)
f	Darcy–Weisbach friction factor (-)
Fr	Froude number (-)
F_s	factor of safety (-)
$\boldsymbol{F}, \boldsymbol{G}$	flux vectors in x and y direction $\left(\mathrm{m}^2\ \mathrm{s}^{-1}, \mathrm{m}^3\ \mathrm{s}^{-2}, \mathrm{m}^3\ \mathrm{s}^{-2}\right)$
$\boldsymbol{F}_m, \boldsymbol{G}_m$	mobile zone flux vectors in x and y direction defined by Guinot (2012) $\left(\mathrm{m}^2\ \mathrm{s}^{-1}, \mathrm{m}^3\ \mathrm{s}^{-2}, \mathrm{m}^3\ \mathrm{s}^{-2}\right)$

g	gravity acceleration $\left(\mathrm{m\ s^{-2}}\right)$
$\boldsymbol{g}$	three dimensional vector of gravitational acceleration $\left(\mathrm{m\ s^{-2}, m\ s^{-2}, m\ s^{-2}}\right)$
h	water depth (m)
h_m	mobile water depth defined by Guinot (2012) (m)
h_s	stagnant water depth defined by Guinot (2012) (m)
$\boldsymbol{H}$	matrix used by Sanders et al. (2008) to transform bed slope source term to a boundary integral $\left(\mathrm{s^{-1}, m\ s^{-2}, m\ s^{-2}\ ; s^{-1}, m\ s^{-2}, m\ s^{-2}}\right)$
$i_p(x, y)$	binary density function (-)
I_1	hydrostatic pressure term $\left(\mathrm{m^3}\right)$
I_2	side reaction term $\left(\mathrm{m^2}\right)$
$\boldsymbol{J}, \boldsymbol{K}$	number of columns and rows of the small cells in a large cell, respectively (-)
l	length (m)
L_d	length of the study domain (m)
n	time level (-)
n_M	Manning coefficient $\left(\mathrm{s\ m^{-\frac{1}{3}}}\right)$
N	number of computational cells (-)
N_w	number of submerged small cells in a large cell (-)
N_Γ	number of submerged small cells adjacent to the cell boundary (-)
p	power describing the relationship between computational resolution and time cost (-)
$\boldsymbol{P}$	pressure $\left(\mathrm{N\ m^{-2}}\right)$
q_x, q_y	x and y components of unit width discharge $\left(\mathrm{m^2\ s^{-1}}\right)$
q_{x_m}, q_{y_m}	x and y components of mobile zone unit width discharge defined by Guinot (2012) $\left(\mathrm{m^2\ s^{-1}}\right)$
$\boldsymbol{q}$	vector of unit width discharge $\left(\mathrm{m^2\ s^{-1}, m^2\ s^{-1}}\right)$
$\|\mathbf{q}\|$	magnitude of flow $\left(\mathrm{m\ s^{-1}}\right)$
$\boldsymbol{Q}$	discharge $\left(\mathrm{m^3\ s^{-1}}\right)$
$\boldsymbol{Q}_0$	inflow/outflow discharge per unit area $\left(\mathrm{m\ s^{-1}}\right)$
$\boldsymbol{Q}_{0_m}$	discharge source term of mobile zone defined by Guinot (2012) $\left(\mathrm{m\ s^{-1}, m^2\ s^{-2}, m^2\ s^{-2}}\right)$
$\boldsymbol{Q}_{0_s}$	discharge source term of stagnant zone defined by Guinot (2012) $(\mathrm{m\ s^{-1}})$
$\boldsymbol{R}$	hydraulic radius (m)
s_0	bed slope (-)
s_{0_x}, s_{0_y}	x and y components of bed slope (-)
s_f	magnitude of the friction gradient (-)

s_{f_x}, s_{f_y}	x and y components of frictional slope (-)
s_l	friction and eddy loss source term in the model of Soares-Frazão et al. (2008) $\left(\mathrm{m}^2\ \mathrm{s}^{-2}\right)$
s_p	bed slope and porosity variation source term in the model of Soares-Frazão et al. (2008) $\left(\mathrm{m}^2\ \mathrm{s}^{-2}\right)$
$\boldsymbol{s}_f$	friction slope (-)
S^L	left wave speed in HLLC scheme $\left(\mathrm{m\ s}^{-1}\right)$
S^M	middle wave speed in HLLC scheme $\left(\mathrm{m\ s}^{-1}\right)$
S^R	right wave speed in HLLC scheme $\left(\mathrm{m\ s}^{-1}\right)$
$\boldsymbol{S}$	vector of source terms $\left(\mathrm{m\ s}^{-1}, \mathrm{m}^2\ \mathrm{s}^{-2}, \mathrm{m}^2\ \mathrm{s}^{-2}\right)$
$\boldsymbol{S}_b$	vector of bed slope source term $\left(\mathrm{m\ s}^{-1}, \mathrm{m}^2\ \mathrm{s}^{-2}, \mathrm{m}^2\ \mathrm{s}^{-2}\right)$
$\boldsymbol{S}_f$	vector of friction slope source term $\left(\mathrm{m\ s}^{-1}, \mathrm{m}^2\ \mathrm{s}^{-2}, \mathrm{m}^2\ \mathrm{s}^{-2}\right)$
$\boldsymbol{S}_{f_m}, \boldsymbol{S}_{g_m}$	source term vectors of bed and friction slope in mobile zone defined by Guinot (2012) $\left(\mathrm{m\ s}^{-1}, \mathrm{m}^2\ \mathrm{s}^{-2}, \mathrm{m}^2\ \mathrm{s}^{-2}\right)$
t	time (s or hr)
t_c	runtime and time cost (s or hr)
$\boldsymbol{T}$	free surface width of a channel (m)
$\boldsymbol{T}_a$	the first term of Eq. 3.17 $\left(\mathrm{m}^{\frac{1}{3}}\ \mathrm{s}^{-1}\right)$
$\boldsymbol{T}_b$	the second term of Eq. 3.17 $\left(\mathrm{m}^{\frac{1}{3}}\ \mathrm{s}^{-1}\right)$
$\boldsymbol{T}_c$	the third term of Eq. 3.17 $\left(\mathrm{m}^{\frac{1}{3}}\ \mathrm{s}^{-1}\right)$
$\boldsymbol{T}_d$	the first term of Eq. 3.37 $\left(\mathrm{m}^{\frac{1}{3}}\ \mathrm{s}^{-1}\right)$
$\boldsymbol{T}_e$	the second term of Eq. 3.37 $\left(\mathrm{m}^{\frac{1}{3}}\ \mathrm{s}^{-1}\right)$
$\boldsymbol{T}_f$	the third term of Eq. 3.37 $\left(\mathrm{m}^{\frac{1}{3}}\ \mathrm{s}^{-1}\right)$
u, v, w	x,y and z velocity components $\left(\mathrm{m\ s}^{-1}\right)$
$u^{\parallel}$	tangential velocity to cells interface $\left(\mathrm{m\ s}^{-1}\right)$
$u^{\perp}$	vertical velocity to cells interface $\left(\mathrm{m\ s}^{-1}\right)$
$\boldsymbol{U}$	vector of conservative variables $\left(\mathrm{m}, \mathrm{m}^2\ \mathrm{s}^{-1}, \mathrm{m}^2\ \mathrm{s}^{-1}\right)$
$\boldsymbol{U}_m$	vector of conservative variables for mobile zone defined by Guinot (2012) $\left(\mathrm{m}, \mathrm{m}^2\ \mathrm{s}^{-1}, \mathrm{m}^2\ \mathrm{s}^{-1}\right)$
$\boldsymbol{U}_s$	vector of conservative variables for stagnant zone defined by Guinot (2012) $\left(\mathrm{m}, \mathrm{m}^2\ \mathrm{s}^{-1}, \mathrm{m}^2\ \mathrm{s}^{-1}\right)$
V_s	shock's speed $\left(\mathrm{m\ s}^{-1}\right)$
$\boldsymbol{V}$	velocity vector $\left(\mathrm{m\ s}^{-1}, \mathrm{m\ s}^{-1}\right)$
$\lVert\boldsymbol{V}\rVert$	magnitude of the velocity $\left(\mathrm{m\ s}^{-1}\right)$

W_d	width of the study domain (m)
x,y	Cartesian coordinates (m)
z_0	channel bed level at $x = 0$ in 1D flows (m)
z	bed level (m)
z'	a binary elevation function giving the minimum of bed and water surface level (m)
$\hat{z}'$	a defined variable in Eq. 3.25 to simplify the equation (m)
z_c	bed level of main channel in a compound channel (m)
z_f	bed level of floodplain in a compound channel (m)
α	BCR (Building coverage ratio) (%)
β	CRF (conveyance reduction factors) (%)
Γ	boundary length (m)
Γ_1 to Γ_4	edges number of a large cell (-)
$\delta x, \delta y$	dimensions of small cells (m)
Δh	difference between a small cell water depth and the average of small cells water depths next to the edge Γ_4 (m)
Δt	time step (s)
$\Delta X, \Delta Y$	dimensions of large cells (m)
Δz	difference between a small cell bed level and the average of small cells bed levels next to the edge Γ_4 (m)
$\Delta z'$	a defined variable in Eq. 3.25 to simplify the equation (m)
ϵ_x, ϵ_y	x and y components of the unit vector parallel to flow (-)
ζ	head loss coefficient accounting for the urban in the model of Soares-Frazão et al. (2008) (-)
η	water surface elevation (m)
η_0	averaged water surface in a cell defined by Sanders et al. (2008) (m)
θ	percentage relative error (%)
$\lambda_{\max}$	maximum wave speed (m/s)
ξ	elevation (m)
Π	viscous stress tensor $\left(\mathrm{N\ m^{-2}, N\ m^{-2}, N\ m^{-2}\ ; N\ m^{-2}, N\ m^{-2}, N\ m^{-2}\ ; N\ m^{-2}, N\ m^{-2}, N\ m^{-2}}\right)$
ρ	mass density $\left(\mathrm{kg\ m^{-3}}\right)$
σ	frictional correction coefficient of intermediate state unit width discharge (-)
σ_z^2	variance of submerged small cells bed level adjacent to the large cell edge $\left(\mathrm{m^2}\right)$
τ	stress $\left(\mathrm{N\ m^{-2}}\right)$

φ	coefficient of unit width discharge distribution (-)
ϕ	volumetric porosity (-)
ϕ_m	porosity of mobile zone defined by Guinot (2012) (-)
ϕ_s	stagnant zone porosity defined by Guinot (2012) (-)
ψ	areal porosity (-)
ψ_a	approximated cross-section's conveyance $\left(\mathrm{m}^2\ \mathrm{s}^{-1}\right)$
$\psi_b(x, y, z, t)$	boundary condition function in derivation of SWE, Appendix A (m)
ψ_e	exact cross-section's conveyance $\left(\mathrm{m}^2\ \mathrm{s}^{-1}\right)$
Ω	cell area $\left(\mathrm{m}^2\right)$
Ω_b	a subdomain of Ω containing the solid obstacle $\left(\mathrm{m}^2\right)$
Ω_s	Small cells area in nested meshes $\left(\mathrm{m}^2\right)$
Ω_w	wet area of a cell $\left(\mathrm{m}^2\right)$

Subscripts and superscript

$\square^B$	boundary cell
$\square^G$	ghost cell
$\square^L$	left
$\square^R$	right
$\overline{\square}$	space-averaged over a large cell
$\square^\wedge$	wetting-drying conditions applied
$\tilde{\square}$	space-averaged over small cells, adjacent to an edge of large cell
$\square^*$	at intermediate state (fractional time step)

Acronyms and abbreviations

1D, 2D, 3D	Number of dimensions used to describe a flow problem
ADE	Alternating Direction Explicit: a finite difference scheme
ADI	Alternating Direction Implicit: a finite difference scheme
AMR	Adaptive Mesh Refinement
ANN	Artificial Neural Network
AOI	Area of Interest
BCR	Building Coverage Ratio
C-property	Conservation Property
CFD	Computational Fluid Dynamics
CPU	Central Processing Unit
CRF	Conveyance Reduction Factors
CUDA	Compute Unified Device Architecture
DCV- FEM	Double Control Volume-Finite Element
DDP	Depth Dependent Porosity
DEM	Digital Elevation Model
DEwI	Diffusive Equations with Inertia
DG	Discontinuous Galerkin
DIP	Dual Integral Porosity
DSM	Digital Surface Model
DTM	Digital Terrain Model
FD	Finite Difference: a numerical method
FE	Finite Elements: a numerical method
FV	Finite Volume: a numerical method
GCI	Grid Convergence Index
GPU	Graphics Processing Unit
GWM	Gravity Wave Model
HLLC	Harten, Lax and van Leer Contact: a Riemann solver
IP	Integral Porosity
KWE	Kinematic Wave Equation
MP	Multi Porosity
MPI	Message Passing Interface
NWM	National Water Model

RMSE	Root Mean Square Error
SG	Sub-grid: the proposed model in this research
SP	Single Porosity
SPH	Smoothed Particle Hydrodynamics
SWE	Shallow Water Equations
T	Traditional model: a particular case of the proposed model in this research making use of a single mesh instead of the nested meshes, similar to conventional finite volume models

Chapter 1

Introduction and literature review

1.1 Introduction

Rivers benefit humans in different ways. They provide drinking and irrigation water, generate energy to sustain a variety of human activities, facilitate traffic of goods and people, enable fishing and also the controlled disposal of waste. Yet, when the capacity of river channels to convey water flows is exceeded, the consequences can be devastating. Flooding currently accounts for a third of the economic loss and 50% or more of fatalities associated with natural disasters globally (Berz, 2000; White, 2000; UNISDR, 2012; Bradford et al., 2012). Examples of devastating flood events are abundant. For instance, in 1998, the Yangtze River basin, China, was hit by a severe flood event from June to August. The event caused over 4100 deaths, destroyed 4.3 million buildings, 20 million homeless people and over 150 billion Yuan economic loss (Wang, 2011). In 2017, certain areas in the world witnessed devastating flooding events. For example, the US experienced a record of $300 billion in disaster losses from hurricanes and flooding together (NOAA, 2018). Also, during the same year, a humanitarian crises emerged in south Asia, where in three countries Nepal, India and Bangladesh, 1000 people lost their lives and the livelihoods of over 41 million were affected by monsoon floods and landslides (Gettleman, 2017; Sanders and Schubert, 2019).

It is expected that in the future the impacts of flooding on communities will increase due to climate change, urbanisation, rising sea levels, population growth, deforestation and rural-to-urban migration, (Kay et al., 2011; McCarthy et al., 2001; Nirupama and Simonovic, 2007; Bradford et al., 2012; Hallegatte et al., 2013; Jongman et al., 2012). As an example, in the UK, the current annual damage to property by flooding is estimated at £1.1bn, but one of the worst case scenarios estimates it could increase up to £27bn by 2080 (Foresight, 2004).

Over recent decades, a large number of studies have been conducted to develop methods and procedures to map the risk of floods in both rural and in urbanised areas. This risk mapping is a central part of decision-making in the field of environmental management (Horritt and Bates, 2001). Three main methods are used to model flood inundation, namely, empirical methods, hydrodynamic models, and simplified conceptual models (Teng et al., 2017) and each of which has advantages and limitations herein discussed in brief.

Empirical methods define relations between two or (most often) many variables, the values of which are typically obtained through fitting observed data from water gauges, field surveys, remote sensing, and statistical models. This approach is relatively quick and can be adopted for the monitoring and assessment of floods. However, this group of methods suffer from serious limitations as follows. These methods typically provide results at low time and spatial resolution, which may not be useful for many engineering applications that require detailed information, although progress in remote sensing has to some extent addressed this issue. In this approach, accurate results rely on acquisition and complex processing of the data, which might be too expensive. The main shortcoming of the approach is perhaps the future is predicted using certain snapshots of past events, while the current and past conditions are not often valid for the coming events.

Hydrodynamic models are capable of capturing the behaviour of flooding in time and space in detail. These models have gained a great popularity in fields such as flood risk assessment, water resources planning, floodplain ecology, and sediment transport. Hydrodynamic models are easily linked to hydrological models through boundary conditions and other source/sink terms (e.g. precipitations and infiltration). Furthermore, by manipulation and perturbation of input data (e.g. initial and boundary conditions, and topographic data), alteration in conditions can be easily transferred to the hydrodynamic models, which pave the way for scenario planning. This approach has certain weaknesses including intensive computations, the propagation of input errors (may reside in uncertainty in data) in time, and the high amount of required data (Teng et al., 2017).

Simplified conceptual models, which are newer compared to the two other approaches, are typically aimed at large scale flood inundation simulations (i.e. flood inundation simulations at scales that are larger than the typical river reach scale, ranging from several kilometers to regional and even global simulations). In these models, simulations can be orders of magnitudes less than hydrodynamic models, which stems from the simplified representation of the physical processes. Studies have shown that simplified conceptual models perform relatively well to the 2D shallow water equations in certain cases, but provide only poor results under certain conditions (Teng et al., 2017).

Because of the clear advantages of hydrodynamic over simplified models, this thesis focuses exclusively on the former. However, other models that fall into the two other types described above are also reviewed in the next sections. The vast majority of flood inundation problems fall into the category of shallow water flows (Wang, 2011). The key feature of a shallow flow is that horizontal length scale is much greater than the vertical length scale, which usually means that vertical flow acceleration can be neglected without substantial loss of accuracy. This assumption results in the so-called hydrostatic approximation, which along with the principles of mass and momentum conservation is used to derive the shallow water equations (SWE). To date, the system of SWE (or simplifications of) is the foundation of the vast majority of commercial and academic codes that are used to model flood inundation problems.

Despite the ability of existing SWE models (in particular those implemented in two horizontal dimensions, or simply 2D) to accurately capture the process of flood inundation, their high computational cost still represents a barrier to a large number of relevant applications. This computational barrier has restricted the application of SWE models to even relatively small domains and short duration flood events (Liang and Smith, 2015), but is even more problematic for large-scale inundation simulations that are needed to inform important decisions. For example, Xia et al. (2019) conducted an investigation to study an extreme flood event caused by the 2015 Storm Desmond in the 2500 km^2 Eden catchment (UK). The study included flood simulations performed through the solution of the 2D SWEs using a shock capturing Godunov type finite volume scheme. This included the validation of the model results (at 5 m resolution, which represents 100 million computational cells) against field data, and the assessment of the sensitivity of the results to mesh resolution and Manning roughness. The study investigated various scenarios (different mesh resolutions and Manning roughness distributions), each of which requiring an extremely large computational time. Another example of computationally demanding flood inundation problems was presented by Echeverribar et al. (2018), who studied a flood event (which occurred in 2015) in a 125 km reach of the River Ebro in Zaragoza (Spain), where the basin's area is 477 km^2. The work was aimed at assessing the use Graphics Processing Units (GPUs, which are discussed in Section 1.2.3 of this thesis) to reduce the simulation time, and also at comparing of the results of the model against field data. The model was based on the solution of the 2D shallow water equations using a Godunov type finite volume scheme over 678,000 cells. These simulations were performed over 21.1 days by an 8-cores Intel® CORE I7-4770 CPU using shared-memory parallelisation. When the same simulations were performed using GPUs, the simulation time reduced to 17.2 hr.

A large number of studies have been conducted aimed at developing robust models that also offer satisfactory computational performance and accuracy. This research can be classified based on the strategies adopted to improve the model performance. For instance, "parallel programming" is the class of studies whereby multiple-core computer architectures are employed to minimise a model's run-time. Another example is the use of simplified models that disregard certain terms in the shallow water equations to reduce the computational costs. Although these types of studies have reported a substantial decrease in computational effort, other recent research (e.g. Liang and Smith, 2015; Cea and Bladé, 2015) showed that these simpler formulations have limitations that may not justify the reported model speedups. To date, none of the approaches (i.e. parallel programming, simplified and others) available provides a perfect model for all flood inundation problems. Low accuracy, insufficient performance for large-scale investigations, or even ill-posedness are some disadvantages that these models suffer from.

In this work, by focusing on the advantages and disadvantages of other approaches, a new model is proposed that falls into the class of "sub-grid" methods. The model is based on the solution of the two-dimensional shallow water equations and makes use of two sets of grids, hereafter referred to as coarse and fine. The equations are solved by a finite volume scheme and the HLLC Riemann solver. The model was developed with the aim of finding a trade-off between speed and accuracy that will enable applications to flood risk assessment at scales that are larger than those that are currently enabled by existing models.

In this chapter, first, the current high-performance models (or the classes of) for flood inundation simulations are reviewed and their strengths and weaknesses are discussed. Then, the main properties of the proposed sub-grid model are overviewed and the aim and objectives of this research are described. A synopsis is also included, giving brief details of the thesis chapters.

1.2 Literature review

Over the last decades a wide range of computational and numerical techniques has emerged that are aimed at reducing computational cost of the flood inundation simulations. In this part of the thesis, some of the most promising techniques are introduced and their main characteristics are discussed. These approaches are namely, simplified models, adaptive grids, parallel computing, porosity models and nested meshes models. In addition, other methods that have recently been employed for fast simulation are also reviewed at the end of this section.

1.2.1 Simplified models

The shallow water equations, which simulate free surface flows, have for a long time been the most popular mathematical approach to simulate 2D flood inundation problems (Wang et al., 2011b; Liang et al., 2008; Mignot et al., 2006). The high computational cost inherently associated with the numerical solution of the full two-dimensional system of equations has, over the last decade, prompted research on the development of potentially more efficient alternatives. One such approach is to use simplified forms of the underlying equations, which may speed up the simulations while representing the main physical characteristics of flow at a plausible level (i.e. lower accuracy, but acceptable relative to 2D SWE). A number of recent research publications falls into this category, such as Bates et al. (2010); Hunter et al. (2008); Lamb et al. (2009); de Almeida et al. (2012); Wang et al. (2011b).

To explain the different simplified forms of the 2D shallow water equations, it is important to review the terms in the 2D SWE (the derivation of which is available in Appendix A) in detail as follows.

$$\frac{\partial}{\partial t}h + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) = 0, \tag{1.1}$$

$$\overbrace{\frac{\partial}{\partial t}(hu)}^{\text{Local acceleration}} + \overbrace{\frac{\partial}{\partial x}\left(hu^2\right) + \frac{\partial}{\partial y}(huv)}^{\text{Convective acceleration}} + \overbrace{\frac{1}{2}\frac{\partial}{\partial x}\left(gh^2\right)}^{\text{Pressure}} - \overbrace{ghs_{0_x}}^{\text{Bed}} + \overbrace{ghs_{f_x}}^{\text{Friction}} = 0, \tag{1.2}$$

$$\frac{\partial}{\partial t}(hv) + \frac{\partial}{\partial y}\left(hv^2\right) + \frac{\partial}{\partial x}(huv) + \frac{1}{2}\frac{\partial}{\partial y}\left(gh^2\right) - ghs_{0_y} + ghs_{f_y} = 0, \tag{1.3}$$

where, h is water depth, t denotes time, (x, y) indicates Cartesian coordinates, u and v are the x and y components of velocity, respectively, g is the gravitational acceleration and s_{f_x},s_{f_y}, s_{0_x} and s_{0_y} are the x and y components of frictional and bed slopes, respectively. Eq. 1.1 is a statement of mass conservation while Eq. 1.2 and Eq. 1.3 express the principle of conservation of linear momentum in the x and y directions, respectively.

In simplified versions of the SWE, the continuity equation (i.e. Eq. 1.1) is maintained unchanged and the simplifications are only applied to the momentum terms. In the derivation of the 2D shallow water equations, three external forces contribute to the change in the fluid's momentum, namely friction between the fluid and the boundaries of the channel, the pressure force between two adjacent bodies of fluid and bed reaction (see the corresponding terms in Eq. 1.2 and Eq. 1.3). The term "local acceleration represents the rate of momentum change at each point of the study domain. This term is used to model the unsteady nature of a flow. The convective acceleration terms model the acceleration due to change in the position of the moving body of fluid along a region of the domain with non-zero velocity gradient.

Primarily meant to reduce the computational cost of solutions, although often also used to improve a model's robustness or to simplify the numerical solution, several models have been developed that neglect one or more terms within the momentum equations. By neglecting the convective acceleration terms in Eq. 1.2 and Eq. 1.3, the so-called "local inertial" or "simplified inertial" or "inertial" formulation is obtained. Ignoring both local and convective acceleration terms simultaneously leads to the Diffusive-Wave Equations, also referred to as zero-inertia equations. These types of equations have been adopted and assessed in a number of previous studies (see e.g. Ponce, 1990; Xia, 1994; Aronica et al., 1998; de Almeida et al., 2012; Martins et al., 2015; Aricò and Nasello, 2018).

The application of a diffusive approach for flood simulations was first proposed by Cunge et al. (1980) and then used in a number of other subsequent works, such as Estrela and Quintas (1994); Bechteler et al. (1994); Yu and Lane (2006). Historically, there has been a tendency to use a diffusive wave model for catchment scale flood inundation simulations, because of the need to reduce the runtime. However, it has later been found that the diffusive wave equations, which neglect inertial terms, have a number of shortcomings, (Bates et al., 2010; Costabile et al., 2012; Cea and Bladé, 2015). For example, the accuracy of solutions of diffusive models deteriorates where spatial gradients of velocity are high or in supercritical flows when Froude number is high and inertial terms become important (see Hunter et al., 2007; Cea and Bladé, 2015). Another issue relates to the time step required in explicit models to maintain numerical stability. The maximum stable time step in a diffusive model decreases with the square of the mesh size (Hunter et al., 2006), which imposes a large computational cost for high resolution meshes compared to the explicit SWE models, in which case the maximum stable time step decreases linearly with mesh size. These shortcomings are particularly relevant for urban flood modelling, since supercritical flows are common in flows over streets and other smooth surfaces and high-resolution simulations are typically required to capture the complexity of the urban topography (Cea and Bladé, 2015). One more limitation to the diffusive model is that the time step stability condition depends on the water surface gradient (Hunter et al., 2006). Therefore, when the solutions tend to horizontal water surface (e.g. solutions near lakes and low land rivers), the CPU time increases dramatically, (Hunter et al., 2006).

In theory, the local inertial model offers a few advantages over the diffusive formulation to describe the dynamics of open channel flows. The local acceleration term takes into account the local rate of momentum change. Then, in terms of the physical representation, the local inertial formulation lies between the diffusion wave approximation and the SWE, (de Almeida and Bates, 2013). Also, the maximum stable time step displays a linear dependency on the size of the computational grid (de Almeida and Bates, 2013). This means that the computational cost of a local inertial model can be significantly lower than the cost of a diffusive model at high-resolution (de Almeida and Bates, 2013). However, the simplification introduced by the local inertial system also degrades the accuracy of simulations. Namely, gradients in the solution are smoothed and the speed of propagation of information is reduced, both effects being more pronounced at high values of the Froude number (de Almeida and Bates, 2013).

Comparing model speedup provided by simplified models is a challenging task for different reasons, such as the variety of numerical schemes, assumptions, accuracy, the diversity of processors running the computational codes, and the properties of test cases. Despite this, in order to provide an overview of the performance of the simplified models, the results of some studies are analysed here. For example, Neal et al. (2012) compared the performance of three models, (1) Bates et al. (2010) model, which is based on local inertial formulation, (2) a model based on the solution of the diffusive equation and (3) a finite volume shallow water equation model adopting the Roe approximate Riemann solver. This study reported that the local inertial approach is up to seven times and 2 orders of magnitude faster than the SWE and diffusive solution, respectively. On the other hand, Cea and Bladé (2015) observed only a marginal speedup when the local inertial approximation was adopted, compared to the full dynamic equations.

The properties of the simplified models can be summarised as,

Positive points:

- The simplified models speed up the simulations in certain flooding problems, such as those involving only gradual changes in time and space (Neal et al., 2012).

Negative points:

- The main limitation of the simplified formulations is lack of generality. Since they neglect part of the physics included in the SWE, strictly speaking they should only be used to simulate problems in which the neglected terms are indeed negligible. This is often difficult to know *a priori*, leading to inaccurate results. Even worse, inaccuracies may be hidden by the modeller's attempt to calibrate the model by adopting unrealistic parameters.
- The speedup achievements of simplified models are not significant in all cases, (Hunter et al., 2008; Néelz and Pender, 2010; Wang et al., 2011b; Liang and Smith, 2015; Cea and Bladé, 2015).
- In simplified models, as some terms of the full dynamic equations are ignored, physical

complexity is reduced. This may result in models that are too sensitive to and dependent on parametrisation, (Costabile et al., 2009, 2012; Fewtrell et al., 2011; Yeh et al., 2011; Liang and Smith, 2015).

- It is recommended that simplified models based on the local inertial and diffusive formulations are only used in subcritical regimes, (de Almeida and Bates, 2013; Cea and Bladé, 2015; Cozzolino et al., 2019). Wet-dry fronts, which are unavoidable in real world flooding simulations, may also be a serious challenge for the local inertia model as the flow fronts is shallow and supercritical. This was addressed and confirmed by Cozzolino et al. (2019).

1.2.2 Adaptive mesh refinement

Local mesh refinement has long been used to provide accurate, high-resolution simulations of flow variables in particular areas of interest (AOI) within a larger computational domain, which is modelled at coarser resolution except at the AOI. The technique delivers detailed results where they are needed, without the additional computational burden that would be required if the whole domain were to be modelled at the same, high resolution. Often the properties of flows vary steeply only at certain regions of the computational domain. In many problems, the AOI is the area with large gradients in the flow properties changing dynamically, as it is often the case during the propagation of floods. This keeps the numerical accuracy at a satisfactory level without incurring excessive increase in computational time (Liang et al., 2015). A well-known structured adaptive refinement grid model was developed initially by Berger and Oliger (1984) and later promoted by others such as Rogers et al. (2001) and Baeza and Mulet (2006).

Different types of adaptive mesh refinement (AMR) strategies have been developed over the last decades, including unstructured mesh, distortion, quadtree refinement, and block structured. Fig. 1.1 illustrates how different resolution levels are defined by each of these mesh refinement techniques. Block-structured and quadtree refinement are two important types of AMR. Details of these methods are described in the following paragraphs.

Block-structured is an appropriate approach for hydrodynamic problems governed by hyperbolic conservation laws, (George, 2011; Ziegler, 2012). The block structured AMR is based on patches (rectangular blocks with multiple levels of refinement, as sketched in Fig. 1.1d) formed around the dynamic features of the solution. These patches are not static but may move during the solution, providing an efficient resolution near steep fronts, (George, 2011). Berger and Oliger (1984) and Berger and Colella (1989) presented two initially successful studies using the block structured

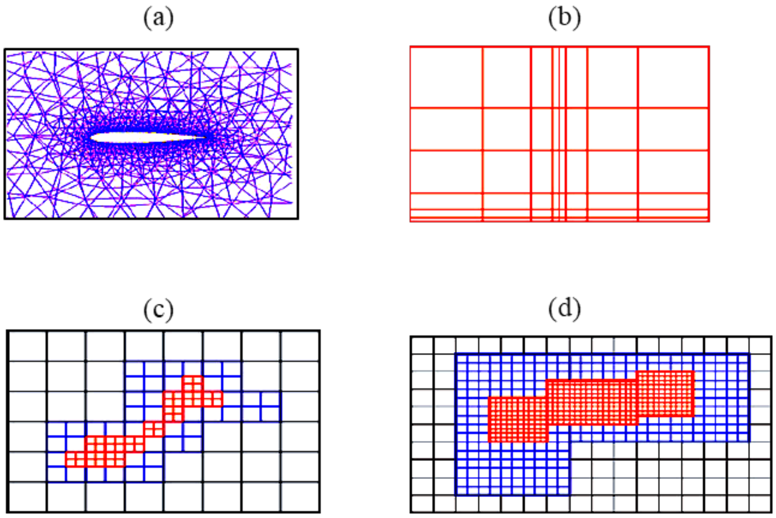


Figure 1.1: Examples of mesh refinement techniques in use to adapt meshes dynamically, (Almgren, 2011). (a) Unstructured, (b) mesh distortion , (c) quadtree refinement, (d) block-structured.

approach. Solutions of the shallow water equations, in particular involving steep fronts such as the dam-break problem, have also been successfully performed using this technique (Berger et al., 2011; George, 2011; Zhang et al., 2015).

Quadtree is a grid adaptation technique in which the unit squares are subdivided into sub-quadrants when interior spatial information (i.e. significant hydraulic properties gradients) confirms a subdivision requirement (Borthwick et al., 2001). Quadtree is a flexible and hieratical data structure, by which a spatial discretisation can be easily refined or coarsened locally with marginal computational overhead (Agbaglah et al., 2011). Adaptive quadtree models have been widely adopted to solve the shallow water equations (see e.g. Rogers et al., 2001; Liang et al., 2004; Liang and Borthwick, 2009; Popinet, 2011; Zhang et al., 2015). Liang (2012) presented a simplified quadtree adaptive grid system. This approach, which is generically aimed at solving partial differential equations, was later used in different numerical schemes to solve the shallow water equations, such as in Wang and Liang (2011), Kesserwani and Liang (2012a), Zhang et al. (2013) and Zhang et al. (2015).

An important issue associated with the solution of the shallow water equations by AMR methods is that the solution may not satisfy both mass conservation and the C-property (i.e. conservation property) concurrently, (Popinet, 2011; Liang and Borthwick, 2009; Liang et al., 2015). Usually, finite volume numerical schemes focus on the homogenous part of the shallow water equations, while the discretisation of the source terms, such as friction, bed slope and inflow/outflow discharges, is performed separately. The agreement between the discretisation of the homogenous part and source terms is critical and a computational model must be able to simulate a steady horizontal water surface (i.e. $\boldsymbol{V} = \mathbf{0}$, $\nabla\eta = 0$, and $\frac{\mathrm{d}\eta}{\mathrm{d}t} = 0$, where $\boldsymbol{V}$ and η are the velocity vector and water surface level, respectively) without producing unphysical waves. Such a model is called well-balanced and satisfies the C-property, (Bermudez and Vázquez-Cendón, 1994; Duran and Marche, 2014). When the C-property is not met, the accuracy and stability of a shallow water model may be affected (Bermudez and Vázquez-Cendón, 1994). On the other hand, the ability of a model to conserve mass is also of utmost importance for accuracy and reliability of a numerical solution. Therefore, the violation of either C-property or mass conservation may result in computational instability or loss of accuracy.

The primary aim of an AMR model is its improved efficiency and accuracy compared with a simple, coarse-grid model. It is therefore important to understand to what extent the computational costs are reduced by the adoption of an AMR strategy. The comparison of the models in terms of efficiency is very difficult, as there is a large range of models available that when applied to different case studies will result in various levels of accuracy. Herein, three studies and their achievements in reducing model run-time are discussed.

Zhang et al. (2014) employed an adaptive grid (quadtree refinement) shallow water model, which utilises an HLLC Riemann solver to estimate the numerical fluxes. The model was designed to simulate large-scale flood inundation problems over irregular beds. The adaptive criteria was defined based on the gradient of free surface elevation. The model was applied to a few cases, one of which (dam break flood wave propagation in Zhanghe River valley, Hubei province, China) is discussed here. In this test, the computational domain covered an area of about 129 km^2. As the domain includes both a narrow river valley and a relatively wide floodplain, high and low-resolution cells were used to discretise the river channel and floodplain, respectively. The maximum and minimum width of cells were 100 m and 25 m, respectively. The research adopted fixed and adaptive meshes, and the AMR approach showed a 60% reduction of the runtime compared to the fixed mesh, while preserving the accuracy.

The second example relates to the work by An et al. (2015), in which the two dimensional shallow water equations are solved by a quadtree AMR. This model, which makes use of a second-order accurate Godunov type finite volume scheme, was employed to study two real flood events in France (the well-known Malpasset dam-break problem) and Korea (levee failure in Backsan). The results of these two simulations proved that the adopted AMR model could save 85%-95% of the computational costs (i.e. speedups of 6.67-20) as opposed to a classical model with the same accuracy.

More recently, Hu et al. (2019) developed a 2D double control-volume finite element (DCV- FEM) flood model based on the shallow water equations. The model makes use of an adaptive unstructured mesh, which is dynamically optimised. The model was adopted to study different scenarios of flooding in an urban area within Greve, Denmark, which is affected by rainfall and sea level variations. The scenarios were defined based on the important recorded flooding events. The results analysis showed that the adaptive model reduced CPU times by 20-88% compared to fixed mesh models.

Herein, the main positive and negative aspects of adaptive mesh models are summarised as

Positive points:

- The high-resolution mesh of AMR in the regions with large gradients of hydraulic properties (velocity and water depth) alterations improves the accuracy.
- Adopting a refined mesh only in the areas of interest and maintaining a low-resolution mesh for the rest of the domain reduces the number of computational cells and consequently computational time, which is desirable in large-scale flood simulations.

Negative points:

- Satisfying mass conservation and C-property concurrently, which is important for both solution accuracy and stability, is a challenge to shallow water adaptive mesh models (Popinet, 2011; Liang and Borthwick, 2009; Liang et al., 2015).
- In AMR models, the time step may be limited as a high level of refinement is applied to a (small) fraction of the domain with complex flow dynamics and high velocity or free-surface gradients, (Liang and Smith, 2015).
- Adaptive mesh models typically adopt the following steps at each time step. (1) According to the prescribed adaptation criteria, a new spatial mesh is generated. (2) The solution from the current mesh is reconstructed and interpolated onto the new mesh. (3) Time integration of the solution is implemented (e.g. using Godunov framework) on the new mesh, (Khakimzyanov et al., 2019). A disadvantage of this methodology is that it makes the solution procedure complex and may cause loss of accuracy by adding governing open more numerical dissipation and dispersion to the solution, (Khakimzyanov et al., 2019).

1.2.3 Parallel computing

Parallel computing solves a single problem using multiple computer resources (core, computers) simultaneously. Large problems are divided into smaller ones, which are then solved concurrently, (Almasi and Gottlieb, 1989; Stout, 2015).

NVIDIA's CUDA, MPI and OpenMP are currently the most widely used parallel programming models. Parallel computing can be broadly divided into two main frameworks, depending on whether different computing cores operate on a single (shared) memory or not. Most desktop computers available today have multiple computer cores and a single memory and therefore can benefit from a shared-memory implementation. However, the structure of most supercomputers consists of a large collection of computers (nodes), each of which has its own memory that is not readily accessible by other machines (distributed memory systems). The Message Passing Interface (MPI) is a standardised and portable message-passing system designed to function on a wide variety of parallel computers, (Message Passing Interface, 2015). Although it is portable to shared-memory environments, the main distinctive feature of MPI is its ability to transfer data across large clusters of computers. For example, in cluster systems a collection of computers work together through the Ethernet, whilst they communicate with each other among the processors using simple commands, similar to read and write. MPI is the most common to have this sort of communication (Stout, 2015). On the other hand, for parallel computers, which are shared-memory systems, there is a standard known as OpenMP, which is a collection of language extensions to Fortran and C/C++, (Stout, 2015).

Another framework for parallel computing that has been attracting recent attention relies on multiple-processor graphics boards capable of performing large volumes of floating number computations. Such technology enables dramatic increases in computing performance by harnessing the power of the GPU (graphics processing unit), (NVIDIA, 2015). CUDA (an acronym for Compute Unified Device Architecture) is the programming language developed by NVIDIA exclusively for this hardware. The CUDA platform, which works in C, C++, and Fortran, provides easier conditions for specialists in parallel programming to utilize GPU resources without need for advanced skills in graphics programming (CUDA, 2015). GPUs with high processing power and massive parallelism, coming from their specific architecture, were initially developed to support computer graphics. However, such architectures have shortly paved the way for further applications including computational tasks. GPUs are now adopted in a variety of fields such as numerical simulation, media processing, medical imaging, eye-tracking, genomics, fluid dynamics, and machine learning (Huisman et al., 2018).

GPU is a type of coprocessor adopted to bear some burden of computationally intensive arithmetic. Although coprocessors have limited functionality compared to a general purpose CPU, they may be optimised for specific tasks, such as floating point arithmetic (Neal et al., 2010). In addition to GPU, there are other examples of coprocessors, such as Clear Speed which is specifically designed for computationally intensive science and engineering purposes (Neal et al., 2010).

In recent years, parallel computing has been applied to the numerical solution of the equations governing open channel flows, especially for large-scale river simulations. For instance, certain two-dimensional hydraulic models have successfully implemented MPI and domain decomposition, such as TRENT (Villanueva and Wright, 2006a,b), CalTWiMS (Pau and Sanders, 2006), TELEMAC-2D (Hervouet, 2000) and RMA (Rao, 2005). In these approaches, the main domain is divided into sub-domains, each of which is run by a core processor (Neal et al., 2010). Models developed by Vacondio et al. (2014); Sætra and Brodtkorb (2010); de la Asunción et al. (2013); Sætra (2014) and Ransom and Younis (2016) provide examples of the application of GPU parallelisation for the solution of the shallow water equations using the finite volume method.

The performance of a parallelised model depends on a number of factors, such as hardware, the method of parallelisation, and even the specific case study of interest. Therefore, it is not easy to compare the models, (Neal et al., 2010). Despite this fact, herein, to provide an overview of the performance of these models, speedups obtained by a few studies are reported. In the case of the programming framework of OpenMP, a flood modelling study, which benefits from a parallel version of the LISFLOOD-FP hydraulic model, was conducted by Neal et al. (2009) achieved a speedup of 5.8 times on 8-core processor relative to a CPU version. Solving a diffusive wave model using 12 cores, Leandro et al. (2014) obtained a speedup of 5.1 times. In the case of GPU, more recently a 2D diffusive model was developed by Park et al. (2019), which was up to 150 times faster than a CPU version. The simulations were performed by a PC with an Intel Core i7-8700 K 4.7-GHz CPU, 16 GB RAM (6 cores) and an NVIDIA GeForce GTX 1060 GPU with 1152 cores.

In the literature, advantages and disadvantages of the parallel programming methods may be summarised as follows:

Positive points:

- Significant run-time reduction rates may be achieved by parallel computing of many flood inundation problems (e.g. Leandro et al., 2014; Park et al., 2019).
- In parallel computation, the speedup is achieved using management and boosting of computer cores. Thus, no further simplifications, assumptions or extra cell coarsening is required for the governing equations (and the solution of) and mesh layout, which translates into speedup without sacrificing accuracy.
- The performance of a model in parallel computing fundamentally depends on the properties

of the hardware utilised. There will definitely be ongoing developments and improvements in technology, influencing the efficiency of these models. Thus, we can expect an outstanding perspective for this approach in solving time consuming flood inundation problems.

Negative points:

- Hardware cost and energy consumption of processors can limit practical usage, (Juez et al., 2016).
- As a model performance depends on parallelisation type, test case, and hardware properties, there may be difficulties in selecting a specific model suitable for a large range of flood inundation problems.
- Parallel processing is not always an appropriate approach to solve time consuming problems. For example,
  - The parallelisation may fail to produce desirable speedups due to “parallel slowdown”. A master core splits up a massive task into cores (workers). The workers accomplish their tasks and return their outputs to the master. Then, there is information exchange between the workers and master cores. In certain cases, a large amount of time is required for communication between the cores degrading the computational model performance. This phenomenon is called “parallel slowdown”, (Maliar, 2015).
  - Scaling is an issue that may arise in some parallel computing simulations, and hardware properties have significant effects on it. There are two types of scaling, strong and weak scaling. In strong scaling, the size of problem is kept unchanged while the number of processors is increased. Weak scaling means that the problem size per processor maintain fixed while further processors are used (Alsabbagh, 2015). In both types, the model efficiency reduces.

1.2.4 Porosity models

Porosity models are a subclass of grid coarsening methods. Large-scale flood propagation simulations with conventional models at fine resolution are too time consuming and this is a serious barrier for a large number of applications. Grid coarsening is an appropriate treatment to reduce the amount of calculations. However, research shows that a simple grid coarsening is not sufficient to represent the effects of the obstacles in the domain of study (e.g. buildings in urban areas) and it accounts for

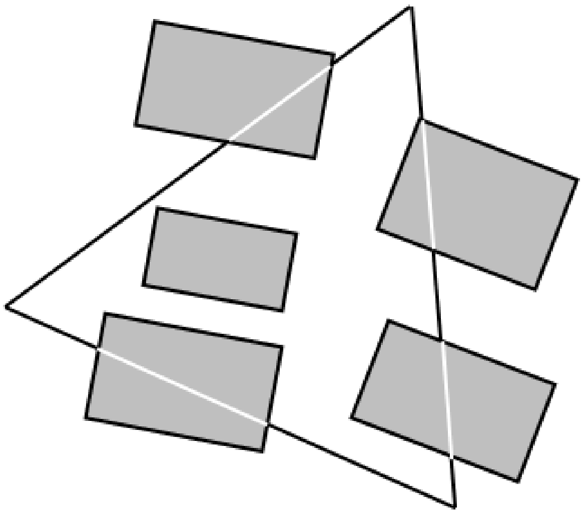


Figure 1.2: Conceptual model of volumetric (ϕ) and areal (ψ) porosity for an arbitrary triangle cell. The grey blocks represent the obstacles (e.g. buildings in an urban area). Volumetric and areal porosity are the ratio of the cell volume, which is represented in white, to the whole cell volume and the black lines (i.e. not occupied by the obstacles) lengths of the cell edges to the whole edge length, respectively (Sanders et al., 2008).

increased numerical error in flood modelling, (Defina, 2000; Bates, 2000; Yu and Lane, 2006, 2011; Brown et al., 2007; Sanders et al., 2008). Part of this loss of accuracy stems from the fact that meshes can only capture the topographic features greater than grid size. Therefore, the idea of using sub-grid data was devised to reflect the effects of important small (relative to mesh size) topographic data in the process of representation (Defina et al., 1994; Bates, 2000; Yu and Lane, 2006, 2011; McMillan and Brasington, 2007; Sanders et al., 2008). Such an idea (incorporation of sub-grid data in the process of simulation) may be implemented by the inclusion of porosity parameters (as explained in the next paragraphs) in the governing equations.

The non-dimensional factor of volumetric porosity is a fraction of a cell area occupied by voids (Bear, 1988; Sanders et al., 2008; Kim et al., 2015) representing the proportion of a cell able to store floodwater. Areal porosity accounts for a percentage of a cell border (edge), that contributes to fluxes and is not blocked by the obstacles (see Fig. 1.2). The areal porosity can be either constant or varies for all edges.

Modelling the sub-grid scale topography information via additional porosity parameters, such as volumetric porosity (ϕ) and areal porosity (ψ), provides improved descriptions of storage and conveyance capacities of the computational cells (coarse cells) during the numerical process (Defina, 2000; Yu and Lane, 2011; McMillan and Brasington, 2007; Sanders et al., 2008; Soares-Frazão et al., 2008; Cea and Vázquez-Cendón, 2010; Chen et al., 2012; Guinot, 2012; Schubert and Sanders, 2012; Kim et al., 2015).

Porosity can be treated as either isotropic or anisotropic depending on whether the orientation of obstacles (and the corresponding effect on the flow) is taken into account. Anisotropy porosity models the effects of obstacles that are unevenly spaced in the cross-flow and along-flow directions, (Kim et al., 2015). In practical simulations, anisotropy in parameters is a part of the simulation nature. However, a numerical model can benefit from isotropy porosity conditions in some cases, such as certain artificial idealised problems. In urban flood inundation studies, the inclusion of porosity parameters distribution in the process of solutions may lead to longer simulations under the same resolution, and so, model simplifications to speed up the simulation are of higher priority. Then, the anisotropy may be neglected.

There are different versions of the porous approach described in the literature. The diversity of porous models stems from different factors, such as the governing equations, the schemes adopted to solve them; and the model's ability to consider isotropy and anisotropy porosities. Herein, some important approaches are reviewed.

In recent years, a number of studies have focused on the use of porosity to model shallow water flows (Sanders et al., 2008; Guinot, 2012; Schubert and Sanders, 2012; Kim et al., 2014, 2015) which report improved accuracy when tested using field- and experimental scale tests. In Sanders et al. (2008), Schubert and Sanders (2012) and Kim et al. (2014, 2015), the porous shallow water formulation is defined as follows,

$$\frac{\partial}{\partial t}\int_{\Omega} i_p \boldsymbol{U} \mathrm{d}\Omega + \oint_{\Gamma} i_p \left(\boldsymbol{F(U)} e_x + \boldsymbol{G(U)} e_y\right) \mathrm{d}\Gamma = \oint_{\Gamma} i_p \boldsymbol{H}\cdot\boldsymbol{e} \mathrm{d}\Gamma + \int_{\Omega} i_p \boldsymbol{S} \mathrm{d}\Omega, \tag{1.4}$$

where,

$$\boldsymbol{U} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \quad \boldsymbol{F(U)} = \begin{pmatrix} q_x \\ \frac{q_x^2}{h} + \frac{g}{2}h^2 \\ \frac{q_x q_y}{h} \end{pmatrix}, \quad \boldsymbol{G(U)} = \begin{pmatrix} q_y \\ \frac{q_x q_y}{h} \\ \frac{q_y^2}{h} + \frac{g}{2}h^2 \end{pmatrix}, \tag{1.5}$$

$$\boldsymbol{S} = \begin{pmatrix} 0 \\ -\left(C_D^f + C_D^b\right) u \parallel \boldsymbol{V} \parallel \\ -\left(C_D^f + C_D^b\right) v \parallel \boldsymbol{V} \parallel \end{pmatrix}, \quad \boldsymbol{H} = \begin{pmatrix} 0 & 0 \\ \frac{1}{2} g h|_{\eta_0}^2 & 0 \\ 0 & \frac{1}{2} g h|_{\eta_0}^2 \end{pmatrix}, \tag{1.6}$$

where, $\parallel \boldsymbol{V} \parallel = \sqrt{u^2 + v^2}$ is the magnitude of the velocity vector, Ω is an arbitrary domain where the equations are solved on (computational cell) and Γ is the domain boundary, e_x and e_y are the x

and y components of the unit vector normal to Γ, respectively. In reality, the water surface elevation (η) varies around the obstacles but Sanders et al. (2008) assumes that η_0 (the averaged water surface within a computational cell) is piecewise constant. Then, water depth is $h|_{\eta_0} = \eta_0 - z$, where z represents ground elevation (assumed linearly varying within a cell by Sanders et al. (2008)). The variable $i_p(x, y)$ is a binary density function providing the two values of zero and one in the different coordinates depending on if there is a solid obstacle at a point ($i_p = 0$) or not ($i_p = 1$):

$$i_p = \begin{cases} 0 & \text{if } (x, y) \in \Omega_b \\ 1 & \text{if } (x, y) \notin \Omega_b \end{cases} \tag{1.7}$$

where Ω_b is a subdomain of Ω containing the solid obstacle. Using the binary function $i_p(x, y)$, the two volumetric and porosity parameters are defined as

$$\phi_i = \frac{1}{\Omega_i} \int_{\Omega_i} i_p \mathrm{d}\Omega, \quad \psi_m = \frac{1}{\Gamma_m} \int_{\Gamma_m} i_p \mathrm{d}\Gamma, \tag{1.8}$$

where i and m are cell and edge number, respectively. In Eq. 1.6, C_D^f and C_D^b denote bed friction and sub-grid scale flow obstacles' drag coefficient, respectively. The former can be estimated by Manning n_M, Chézy c, or Darcy–Weisbach f as follows, (Schubert and Sanders, 2012),

$$C_D^f = \frac{g n_m^2}{h^{\frac{1}{3}}} = \frac{g}{c^2} = \frac{f}{8}, \tag{1.9}$$

and C_D^b is calculated by the following expressions available to model drag (e.g. Nepf, 1999),

$$C_D^b = \frac{1}{2} C_D^0 a_f h \tag{1.10}$$

where, C_D^0 is the dimensionless obstruction drag coefficient (Kim et al., 2014) which is a function of the shape of the flow obstacle and Reynolds number (Sanders et al., 2008; Kim et al., 2015). Values of C_D^0 for 2D flows have been tabulated in various studies (e.g. Munson et al., 2006). The term a_f represents a frontal width of obstructions projected to flow path in a cell with the area of Ω, (Sanders et al., 2008; Kim et al., 2015; Schubert and Sanders, 2012; Nepf, 1999). Further guidance for the estimation of a_f and C_D^0 in practical applications is available in Schubert and Sanders (2012). The model of Sanders et al. (2008) was developed based on an integral form of porous shallow water equations to account for anisotropy. The parameters Ω and ψ are not considered fixed across the domain study and depend on the shape, size and position of an obstacle in the cells.

Guinot (2012) proposed a different approach (differential form) to apply the anisotropy of porosity parameters in porous shallow water equations. This model is based on multiple porosity parameters to represent a macroscopic modelling of floods in urban areas through splitting of the domain into zones with distinct porosities, water depths and velocity fields, such as shown in Fig. 1.3. This research showed that multi porosity models provide a better representation of flood propagation than the single porosity model. Guinot (2012) divides the domain into three parts (see Fig. 1.3), (1) buildings are impermeable to the flow and no water is modelled in this zone, (2) mobile water with the hydraulic characteristics of h_m, q_{xm} and q_{ym} (water depth and unit width discharges in the x and y directions, respectively), and (3) stagnant water with still water depth h_s, and null velocity and discharge. The governing equations are defined as

$$\begin{aligned}&\tfrac{\partial}{\partial t}\boldsymbol{U}_m + \tfrac{\partial}{\partial x}\boldsymbol{F}_m + \tfrac{\partial}{\partial y}\boldsymbol{G}_m = \boldsymbol{S}_m^{(f)} + \boldsymbol{S}_m^{(g)} + \boldsymbol{Q}_m,\\ &\tfrac{\partial}{\partial t}\boldsymbol{U}_s = \boldsymbol{Q}_s,\end{aligned} \tag{1.11}$$

$$\begin{gathered}\boldsymbol{U}_m = \begin{pmatrix}\phi_m h_m\\ \phi_m q_{xm}\\ \phi_m q_{ym}\end{pmatrix},\ \boldsymbol{F}_m = \begin{pmatrix}\phi_m q_{xm}\\ \phi_m \frac{q_{xm}^2}{h_m} + \phi_m \frac{g}{2} h_m^2\\ \phi_m \frac{q_{xm}q_{ym}}{h_m}\end{pmatrix},\ \boldsymbol{G}_m = \begin{pmatrix}\phi_m q_{ym}\\ \phi_m \frac{q_{xm}q_{ym}}{h_m}\\ \phi_m \frac{q_{ym}^2}{h_m} + \phi_m \frac{g}{2} h_m^2\end{pmatrix},\\ \boldsymbol{S}_m^{(f)} = \begin{pmatrix}0\\ -\phi_m g h_m s_{f_x}\\ -\phi_m g h_m s_{f_y}\end{pmatrix},\ \boldsymbol{S}_m^{(f)} = \begin{pmatrix}0\\ \left(\phi_m g h_m s_{0_x} + \frac{h_m^2}{2}\frac{\partial \phi_m}{\partial x}\right) g\\ \left(\phi_m g h_m s_{0_y} + \frac{h_m^2}{2}\frac{\partial \phi_m}{\partial y}\right) g\end{pmatrix},\\ \boldsymbol{Q}_m = \begin{pmatrix}\phi_m E_G\\ \phi_m \frac{E_G - |E_G|}{2} u_m\\ \phi_m \frac{E_G - |E_G|}{2} v_m\end{pmatrix},\ \boldsymbol{U}_s = \left(\phi_s h_s\right),\ \boldsymbol{Q}_s = \left(-\phi_m E_G\right),\end{gathered} \tag{1.12}$$

where the subscripts b, s and m correspond to the building, stagnant and mobile zones, respectively, E_G accounts for the mass exchange between the stagnant and the mobile water, and is positive when mass is transferred from the stagnant to mobile zone. Guinot (2012) has presented several models for estimation of E_G.

Eq. 1.11 and Eq. 1.12 provide a more detailed description of how porosity can influence the physics of open channel flow in urban flood inundation simulations. For example, Fig. 1.4 shows an urban area where the flow generally moves in the x direction. Owing to the alignment of the buildings, water is trapped within the urban blocks. This stagnant water or passive water, which is represented by the hatched areas, has zero or near-zero velocity, whereas the mobile water flow has non-zero velocity in the rest of the domain. This is translated into a multi-modal velocity field (i.e. zero and non-zero velocity fields). The approach by Guinot (2012), which makes use of multi porosity, is able to model the multi velocity fields that appear in the case of Fig. 1.4. In such cases, the multi porosity approach has advantages over the single porosity model. Namely, it improves the accuracy of the solution and also results in more robustness in computing the head losses. Hence, head losses only occur in the mobile regions of the domain Guinot (2012).

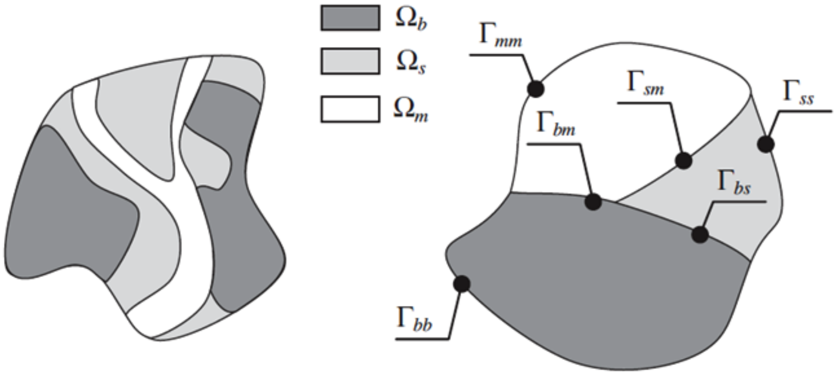


Figure 1.3: Division of the domain into three distinct parts, buildings, stagnant water and mobile water, defined by (Guinot, 2012). Variables Ω and Γ denote the area and boundaries of each zone, respectively. The subscripts b, s and m correspond to the building, stagnant and mobile zones, respectively. The double subscripts of the borders specify a boundary common between two zones. For example, Γ_{mb} is the boundary common between the building and mobile water areas. Each region has distinct porosities, water depths, and velocity field (Guinot, 2012). The left figure shows the mixed zones in a real case and the right figure illustrates how the porosities are estimated for the case in Guinot (2012).

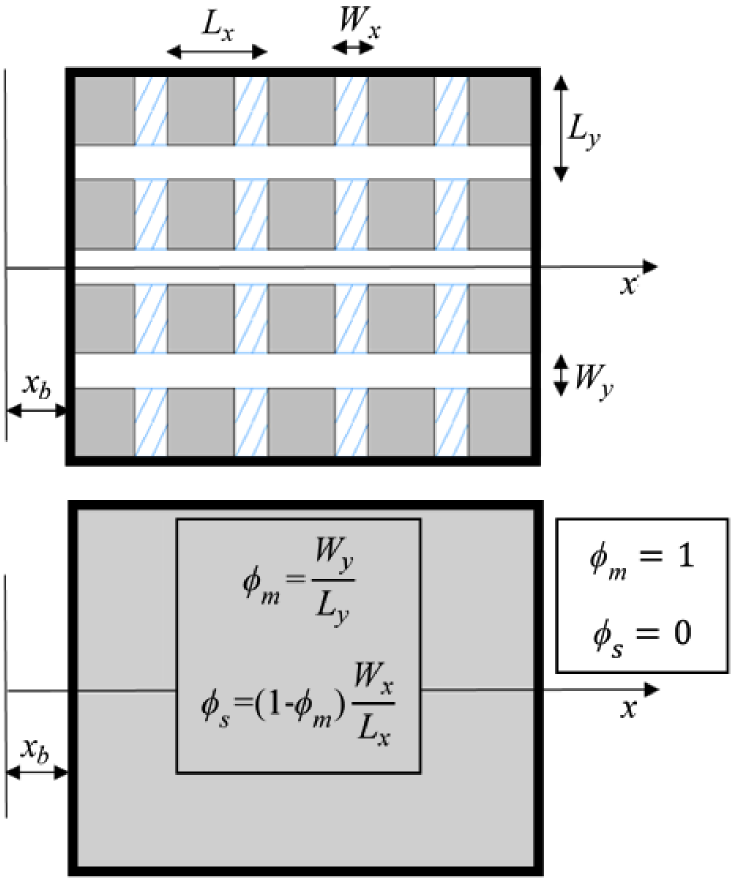

L_x
W_x
L_y
x
W_y
x_b
$\phi_m = \frac{W_y}{L_y}$
$\phi_s = (1-\phi_m)\frac{W_x}{L_x}$
$\phi_m = 1$
$\phi_s = 0$
x
x_b

Figure 1.4: Treatment of an urban area using the multiple porosity model of Guinot (2012). The top figure shows the plan view of an urban area. The urban domain is split into three distinct parts: the building impermeable zone (in grey), the stagnant water zone or passive water trapped between the high elevation ground features (hatched); and the mobile water zone (in white). The bottom figure shows a zoomed-in area in the model of Guinot (2012) that uses ϕ_m and ϕ_s to represent important details, such as blockage and stagnant water in the computational process. This approach allows the model to simulate two very different velocity field regions (the velocity of mobile part and null velocity of immobile parts) leading to higher accuracy. The figure has been adapted from Guinot (2012) for clarity.

Other versions of the porous model are also available in the literature that adopt much simpler governing equations. For example, Soares-Frazão et al. (2008) employed the following 2D porous shallow water equations to simulate urban floods.

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x} + \frac{\partial \boldsymbol{G}}{\partial y} = \boldsymbol{S}, \tag{1.13}$$

$$\begin{aligned} \boldsymbol{U} &= \begin{pmatrix} \phi h \\ \phi hu \\ \phi hv \end{pmatrix}, \quad \boldsymbol{F} = \begin{pmatrix} \phi hu \\ \phi\left(hu^2 + \frac{g}{2}h^2\right) \\ \phi huv \end{pmatrix}, \\ \boldsymbol{G} &= \begin{pmatrix} \phi hv \\ \phi huv \\ \phi\left(hv^2 + \frac{g}{2}h^2\right) \end{pmatrix}, \quad \boldsymbol{S} = \begin{pmatrix} 0 \\ s_{p_x} - s_{l_x} \\ s_{p_y} - s_{l_y} \end{pmatrix} \end{aligned} \tag{1.14}$$

$$s_{p_x} = \phi ghs_{0_x} + \frac{g}{2}h^2\frac{\partial \phi}{\partial x}, \quad s_{p_y} = \phi ghs_{0_x} + \frac{g}{2}h^2\frac{\partial \phi}{\partial y}, \tag{1.15}$$

$$s_{l_x} = \left(s_{f_x} + \frac{1}{L}\zeta\left(u^2+v^2\right)^{\frac{1}{2}}\frac{u}{2g}\right)\phi gh, \quad s_{l_y} = \left(s_{f_y} + \frac{1}{L}\zeta\left(u^2+v^2\right)^{\frac{1}{2}}\frac{v}{2g}\right)\phi gh, \tag{1.16}$$

where, s_{l_x} and s_{l_y} are friction and eddy loss source terms in the x and y directions, respectively and ζ is head loss coefficient. The above equations are very similar to the other aforementioned porous models and the major changes correspond to the source term vector $\boldsymbol{S}$, where s_{p_x} and s_{p_y} comprise the pressure source term stemming from: (1) bed elevation variation expressed by s_{0_x} and s_{0_y} and (2) changes in cross-section width associated with $\frac{\partial \phi}{\partial x}$ and $\frac{\partial \phi}{\partial y}$.

In this model, only one porosity parameter ϕ is included, which represents both volumetric and areal porosity and is both homogeneous (i.e. constant over the whole domain) and isotropic.

Other simplified porosity-based models that are not based on the full version of the SWEs have also been proposed, such as those of Huang et al. (2014) and Yu and Lane (2006), which utilise the zero-inertia formulation. In non-inertia models, flow acceleration is assumed negligible compared to the gravitational and frictional terms. Thus, such models have simpler form regarding the shallow water equations. The porosity form of these models can be written as

$$\begin{aligned} &\tfrac{\partial(\phi h)}{\partial t} + \tfrac{\partial(hu)}{\partial x} + \tfrac{\partial(hv)}{\partial y} = Q, \\ &-\tfrac{\partial \eta}{\partial x} = u\left[\tfrac{n_M^2\sqrt{u^2+v^2}}{h^{\frac{4}{3}}}\right], \\ &-\tfrac{\partial \eta}{\partial y} = v\left[\tfrac{n_M^2\sqrt{u^2+v^2}}{h^{\frac{4}{3}}}\right], \end{aligned} \tag{1.17}$$

where, Q is the discharge source term.

The application of the detailed bathymetric and sub-grid data is not limited to the porosity parameters, such as volumetric and areal porosity. Various studies, including those of Defina (2000); Yu and Lane (2011); Casulli (2009); Casulli and Stelling (2011) and Stelling (2012), use detailed data for friction calculations (typically in a somewhat simplified way). In these studies, sub-grid bed information inside a large cell is adopted to estimate the “effective depth” of a computational cell that is then used in Manning’s equation $s_f = \frac{n_M^2 \sqrt{u^2+v^2}}{h^{\frac{4}{3}}}$ to estimate the friction slope. In large computational cells, the assumption that the flow variables are constant across the cell may be an extremely inaccurate representation of what occurs in the real-world. Thus “effective depth” is an example of an attempt to include such variations of flow properties (in this case, depth) through a relatively simple concept (e.g. see Eq. 1.18, which is described in the following section).

Sub-grid models may be classified into three categories, namely single porosity (SP), integral porosity (IP), and multi porosity (MP). Single porosity approaches include only one value that accounts for the both effective cell volume and conveyance. The model by Soares-Frazão et al. (2008) is a good example of this approach. Two limitations are reported for these models. First, SP models are mostly adopted for the simulation of urban flooding events, where buildings are assumed unsubmerged during a flood event. As a result, except for the early versions (e.g. Defina, 2000), most of the SP models make use of a depth-independent porous coefficient (Guinot et al., 2018). Second, conventional SP models do not include anisotropy effects. However, this shortcoming has more recently been addressed, (see e.g. Viero and Valipour, 2017). The integral porosity approach (e.g. Sanders et al., 2008) utilises both volume (ϕ) and areal (ψ) porosity parameters, and is able to incorporate anisotropy of porous coefficients into the solution. These models, which were originally designed for urban flooding simulations, do not typically use dynamic porosity coefficients (i.e. ϕ and ψ are depth-independent). Nevertheless, some recent studies have been conducted that include depth-dependent porosity via IP (see Özgen et al., 2016a,b). The multi porosity approach (Guinot, 2012) was developed to improve the SP approach by incorporating the anisotropy of porous coefficients into the solutions. Later, Guinot et al. (2017) presented Dual Integral Porosity (DIP) to improve IP. DIP is not a new version of MP but has some of its properties. The idea of DIP stems from this point that as we define storage (within the cell) differently from conveyance (at cell boundaries) porosity, we should use different conservative variables within the cell from boundaries. In other words, the values of conservative variables within a cell are modified for the cell boundaries through a method based on the assumption that obstacles in the cells affect the flow velocity significantly, whereas their effect on free surface elevation is negligible. Guinot et al. (2017) showed that the DIP approach leads to more accurate mass and momentum fluxes compared to IP models. The DIP method was recently upgraded to Depth-Dependent Porosity (DDP), which dynamically estimates the porosity, (Guinot et al., 2018).

Overall, the main advantages and disadvantages of porosity-type sub-grid models can be summarised as follows.

Positive points:

- The porosity coefficients, such as volumetric and areal porosity, in the governing equations provide accurate descriptions of the pure storage and conveyance of a cell that contributes to mass and momentum exchanges and results in a more accurate solution compared to non-porous solutions on low-resolution meshes. Porosity models, which perform simulations at coarse resolutions, reduce runtime.

Negative points:

- In current sub-grid models, the porosity parameters ϕ and ψ are typically water depth-independent, even though obstacles may be submerged during a flood event. Although this issue has recently been addressed (e.g. Özgen et al., 2016a,b; Guinot et al., 2018), further studies are still required to test the performance of such models.
- In the multiple porosity model of Guinot (2012), each cell is divided into three parts, mobile water, stagnant water, and rigid obstacles. The division of the domain into different zones results in higher accuracy. Nevertheless, in many real-world case studies, it might be difficult to distinguish these zones and this might impose a new complexity to the model.
- Kim et al. (2015) showed that a porous model may fail to predict wakes and wave reflections around flow obstructions.
- According to Kim et al. (2015), numerical errors in porous models may be divided into three groups: structural model errors, scale errors, and porosity model errors. Structural errors are related to the governing equations and their limitations, such as assumptions and simplifications to predict the behaviour of flood dynamics. Scale errors correspond to grid size. Porosity model errors stem from the parametrisation of the sub-grid obstacles. Kim et al. (2015) reported that porosity model errors might be higher than scale errors, which highlights the importance of porosity parameter estimations.

1.2.5 Nested meshes, a subclass of sub-grid models

Models within this class typically make use of two meshes. A finely resolved mesh stores detailed data (e.g. topography and roughness), whilst the governing equations are solved at the coarse resolution mesh. Information at fine resolution is incorporated into the numerical solution performed at coarse grid resolution in different ways. Such an approach may also be referred to as sub-grid (instead of nested meshes) modelling. In other words, the nested meshes method is a subclass of the sub-grid approach.

Sub-grid models are aimed at low-resolution computational mesh adaptation to reduce the computational cost. However, to manage the degradation of the accuracy of a solution (because of computational grid coarsening) the effects of finely resolved data are included in the solution process. In the literature, there are a wide range of sub-grid models. The diversity in sub-grid resides in factors such as governing equations, and schemes are adopted to upscale finely resolved data (e.g. using porosity parameters to modify the conveyance and storage capacity of cells, or to provide an accurate estimate of a cell's mean water depth where bed is irregular). Nowadays, sub-grid methods have gained great popularity in various areas of computational hydrodynamics, such as modelling flow in porous media (e.g. Hageman and de Borst, 2020; de Borst, 2017; Mohammadnejad and Khoei, 2013), open channels (e.g. Volp et al., 2013; Sanders and Schubert, 2019), and coastal and ocean environments (e.g. Sehili et al., 2014; Zhang, 2020).

Sub-grid models for the type of flows that concern this thesis –namely free surface flood inundation flows in rivers and floodplains– typically make use of two sets of resolved data in the solution process, namely: topography and roughness. Most of the research efforts to date have been devoted to upscaling sub-grid bathymetric data (the main stream of the sub-grid bathymetry data adaptation, which relates to the porosity approach discussed previously), whereas minimal attention has given to the upscaling of the frictional terms in governing equations. Friction has been represented in different ways within sub-grid models. In many sub-grid models, friction effects are based on the assumption that the friction slope is constant over a computational cell (e.g. Volp et al., 2013; Platzek et al., 2016). This assumption is widely used by models that focus solely on sub-grid bathymetric parametrisation, such as porosity models. Although these models have shown acceptable accuracy in some simulations, the assumption may be problematic in many other cases such as those dealing with local large variation of bed topography and spatially varying shear. In these cases, there is a misrepresentation of the linkage between energy loss, velocity magnitude, and water surface elevation, leading to inaccurate estimation of friction as a result of velocities and fluxes (e.g. Sehili et al., 2014; Volp et al., 2013; Shin, 2016).

The employment of dynamic (time-varying) rather than constant frictional parameters can be a step toward friction computations, particularly for partially wet cells. Such approaches that typically rely on roughness height (vegetation or land cover), mixing layer theory, and the resolved water depth (e.g. Casas et al., 2010) have been based on either a balance between gravitational force, and drag (surface and form) force (e.g. Huthoff et al., 2007), or adaptation of a scaling factor (e.g. Özgen et al., 2015) for incorporation of local friction (Shin, 2016). It should be noted that these approaches only address a part of the problem, which is the variation of frictional parameter in time. However, models that consider both spatial and temporal distribution of frictional parameters, such as the model proposed in this thesis, may substantially improve the solutions' accuracy in certain cases.

In this section, three nested meshes models are reviewed. The first corresponds to Casulli (2009), which has a particular importance within the literature, especially because it is implemented within the popular computational model HEC-RAS (U.S. Army Corps of Engineers, Hydraulic Engineering Centre, Davis, CA) Version 5. The two other approaches refer to works recently published by Hénonin et al. (2015) and Sanders and Schubert (2019), which have close similarities to the model proposed in this thesis. Because of these important similarities, herein, these models are presented and briefly discussed.

Casulli (2009) makes use of a semi implicit non-Godunov finite volume scheme. This model adopts two sets of irregular meshes, fine and coarse. The coarse mesh is the computational mesh on which the governing equations are solved. The fine mesh provides high-resolution topographic data for the model (i.e. sub-grid bathymetric information).The scheme employed by Casulli (2009) discretises the solution for the water surface level, which is assumed constant within each computational cell, and the velocities at the midpoint on cell edges (the edges of considered control volume). In Casulli (2009), as computational cells are allowed to be partially wet [i.e. boundaries of the control volumes (cells) are moving] and also to improve the reconstruction of hydraulic variables at cell edges, the sub-grid topographic data is adopted. Casulli (2009) uses sub-grid information for three main purposes, namely, the reconstruction of water depth on the cell edges ($h = \eta - z$), estimation of the wet cross-section at each edge of a cell, and finally the computation of water volume within each cell.

One of the latest versions of the nested meshes approach was presented by Hénonin et al. (2015). This method benefits from two sets of meshes, coarse and fine. The coarse mesh acts as the computational mesh and the small mesh provides the fine scale topography data (see Fig. 1.5). In a coarse cell, the water level (η) is assumed constant at each time level but the water depth (h) varies as $h = \eta - z$, where z is the finely resolved bed elevation. Hénonin et al. (2015) employed this distribution of water depth inside a coarse cell in two parts. First, the approximation of averaged water depth $\left(\overline{h}\right)$ at coarse cells by

$$\overline{h} = \frac{\sum_{j=1}^{N_w} \max\left[\left(\eta - z_j\right), 0\right] \Omega_{s_j}}{\sum_{j=1}^{N_w} \Omega_{s_j}} = \frac{\sum_{j=1}^{N_w} \max\left[\left(\eta - z_j\right), 0\right]}{N_w}, \tag{1.18}$$

where, Ω_s denotes the area of a small cell, and N_w is the number of submerged small cells within a large cell. This averaged depth $\overline{h}$ is used to compute mass and momentum flux exchanges. Second, at post-processing, the unit width discharges may be downscaled from the coarse to fine mesh through a simple formulation (a combination of linear interpolation of coarse cells fluxes and $h^{\frac{2}{3}}$), which uses the water depths of the small cells. Fig. 1.5 presents an example of Hénonin et al. (2015) nested cells. Fig. 1.5a displays the dual meshes (nested meshes) and bed level distribution of small cells within a coarse cell. The plan view of Fig. 1.5b shows how the model represents a partially wet cell when the water elevation η is at a level that cannot submerge some small cells. The governing equations are solved by the Alternating Direction Implicit (ADI) method, following Abbott (1979).

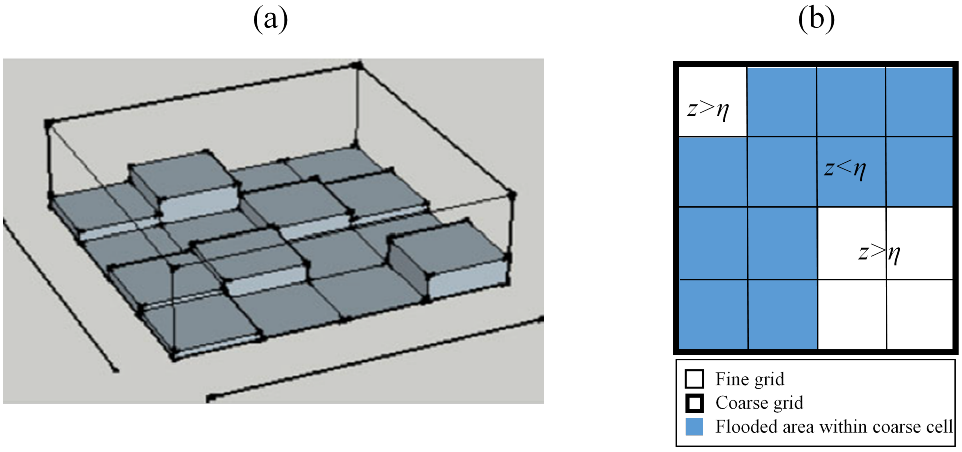


Figure 1.5: The dual meshes of Hénonin et al. (2015). (a) The arrangement of small cells within a large cell. (b) The possibility of partially wet cells appearing, where the water surface η in a large cell cannot submerge the whole coarse cell due to the bed levels of the small cells, (Hénonin et al., 2015).

More recently, Sanders and Schubert (2019) proposed a sub-grid model PRIMo (Parallel Raster Inundation Model). The model makes use of two sets of meshes, U-grid and z-grid. The latter, which is nested in the U-grid, incorporates a high-resolution raster digital model into the solution process. The mesh for the U-grid is far coarser than that of the z-grid, and so the solution is implemented on the U-grid to achieve speedup. The governing equations are the integral form of 2D shallow water equations (SWE) that are solved in the framework of Godunov type finite volume method.

In this model, the aim of fast simulations is achieved as a combined result of parallel computing and the sub-grid approach used in the solution process. The effective cell volume and boundary are automatically incorporated into the solution through comparing water surface elevation and z-grid topography data. Also, to improve accuracy, flux exchanges between the cells are computed at z-grid and then integrated at U-grid resolution.

Generally, the following conclusions can be drawn about nested meshes methods.

Positive points:

- These methods are based on the idea of coarsening the computational grid, which is efficient at reducing the number of cells and CPU time.
- Nested-mesh models require (marginally) extra computational effort compared to simple grid coarsening models for simulations when the same computational resolution is used. However, since the nested meshes provide improved accuracy at coarse resolution, faster simulations may be achieved for a given target accuracy.

Negative points:

- Upscaling of terms within the current generation of sub-grid models is still in its infancy and a rigorous treatment of how to upscale the different terms in the equations is missing.
- Nested meshes models typically make limited use of high-resolution mesh data (mainly the estimation of the average water depth and velocity in a cell) whereas the information can be utilised in all parts of the computational process, including calculations of friction and bed slope effects, and mass and momentum fluxes between the computational cells.

1.2.6 Other approaches

The previous sections have shown the relatively wide range of methods developed over the last decades to improve the computational performance of flood inundation models. However, given the incredibly large number of publications in this area, only the methods that are most popular or those deemed most promising have been described. This section is aimed at providing further, albeit brief, discussion of other techniques that have been proposed for modelling flood inundation. These include, for example, the coupling of 1D-2D models, smoothed particle hydrodynamics (SPH), and machine learning.

a) 1D-2D coupled models

In recent years, a wide range of computational models has been developed that are based on the shallow-water equations (SWE), which are mostly used in studies of free-surface flows, such as rivers, lakes, and estuaries. These models can be written in one-dimensional (1D), two-dimensional (2D), or three-dimensional (3D) forms. Models in two and three dimensions are capable of providing a more detailed description of the flow, at the expense of more calculations and longer run time. Indeed, the selection of the number of dimensions used to model the problem is a trade-off between the level of detail in the results (and often accuracy) and the computational cost, although the availability of data sometimes also restricts the options. One dimensional models are commonly used to estimate the water depth (or alternatively the flow area) and discharge (or velocity) of flow routing along a river, whereas 2-D and 3-D provide the distribution of the conserved variables in two (usually horizontal) and three spatial dimensions with vector quantities (e.g. velocity) represented by two and three cartesian components, respectively. These types of models are mostly applied to compute the hydraulic properties of flow in local river reaches and floodplains, lakes and reservoirs, and estuaries and coastal regions (Verwey, 2001; Twigt et al., 2009; Kuiry et al., 2010; Fernández-Pato and García-Navarro, 2018; Morales-Hernández et al., 2018) even though 2D models are being increasingly adopted to model very large scale problems.

On the one hand, 1D models often provide enough information for a range of practical applications, especially when the main focus is on flow discharges within the capacity of the main channel. However, in general the ability of 1D models to predict accurately the propagation of flood waves over floodplains or urban areas is limited. On the other hand, 2D (or 3D) models are capable of accurate predictions of complex and non-canalised flows, although they may impose a heavy computational cost to most medium- to large-scale real-world flooding studies. The coupling of such models, such as 1D-2D (i.e. 1D for the main channel and 2D for the floodplains) and 1D-3D, has emerged as a natural solution to these difficulties.

Despite the fact that 1D-3D models have become popular in certain practical applications, such as the interaction between rivers and oceans (e.g. Chen et al., 2013), 1D-2D are by far the most popular example of coupled models. For instance, coupled 1D-2D models have been widely employed by many researchers for practical and academic purposes, such as Bladé et al. (1994), Kuiry et al. (2010), Villanueva and Wright (2006b), McMillan and Brasington (2007), Castellarin et al. (2011), Morales-Hernández et al. (2016), Fernández-Pato and García-Navarro (2018), Morales-Hernández et al. (2018), and also been incorporated in certain well-known commercial packages (e.g. MIKE FLOOD, SOBEK, Flood Modeller , TUFLOW, HEC-RAS).

One of the most important differences between existing 1D-2D models is the way mass and momentum are exchanged between the two components. In many 1D-2D models, this connection is formulated simplistically using lateral weir equation (see e.g. Dhondia and Stelling, 2002; Di Baldassarre et al., 2009; Verwey, 2001) whereby the exchange of mass is modelled through level differences, (Lin et al., 2006; Morales-Hernández et al., 2016). Miglio et al. (2005) introduced an iterative method to transform 2D variables into 1D by integration of 2D quantities and applying continuity at the interfaces. Yu and Lane (2007) employed a loosely coupled approach in which 1D model represents the boundary conditions for the 2D model at the interfaces of the floodplain and main channel before the initialization of the 2D model. Finaud-Guyot et al. (2011) presented a 1D-2D model that in addition to mass exchange, includes momentum transfer between the main channel and floodplain. In order to achieve a fully conservative 1D-2D model, Morales-Hernández et al. (2013) conducted a study to incorporate information otherwise omitted from each computational domain into the boundary conditions of the other model. This study focused on a set of academic test cases to check the performance of the model under the different flow regimes at the coupling zones. later, Morales-Hernández et al. (2016) in the light of Morales-Hernández et al. (2013), carried out a research study to pave the way for the simulation of real-word and practical engineering problems.

The weaknesses and strengths of the coupling of 1D-2D models can be explained as follows.

Positive points:

- As 1D-2D models use less computational cells than fully 2D models, their computational costs are substantially lower.

Negative points:

- It has been shown that different 1D-2D models present a large range of predictions of the volume of water exchanged between the main channel and floodplain of rivers in the UK EA benchmark test (Néelz and Pender, 2010).

b) Smoothed particle hydrodynamics

The methods described in the previous sections are all examples of Eulerian methods, whereby the computational domain is discretised using meshes (fixed or adaptive) on which the values of partial derivatives (or integrals) are approximated. Such an approach (Eulerian) has been the basis of the revolution made in Computational Fluid Dynamics (CFD) from the late 1960s and early 70s. On the other hand, Lagrangian methods represent a different class of computational methods, whereby the governing equations are defined and evaluated in a frame of reference that moves with the fluid. A common approach in Lagrangian methods is to use discrete particles, which are carried with the flow. In this case, the values of variables are taken to be either the particle values at the exact particle location or else a weighted average of all particle values in a prescribed neighbourhood. Methods adopting this procedure to obtain values of variables and their derivatives based on a weighted averaging of values obtained from moving neighbour particles are referred to as Smoothed Particle Hydrodynamics (SPH) (Cossins, 2010).

Gingold and Monaghan (1977) and Lucy (1977) proposed the SPH idea independently. This method was first developed in astrophysics, where the fluid density typically varies in time and space many orders of magnitude within an unbounded environment. Later, the SPH approach was applied [firstly by Monaghan (1994)] to the solution of equations governing the motion of free surface fluids.

SPH methods have found applications in different areas such as coastal engineering (Barreiro et al., 2013; Altomare et al., 2017; Crespo et al., 2017), flooding (Vacondio et al., 2013; Prakash et al., 2014), turbulent flows (Violeau and Issa, 2007; Ren et al., 2016), and multiphase flows (Mokos et al., 2017; Gong et al., 2016) [a long list of applications and references is available in Altomare et al. (2018) and Gomez-Gesteira et al. (2012)]. This wide range of applications stems from the significant capacities of SPH methods, such as relative ease of programming even for 2D and 3D simulations (at least for basics); handling of multi-fluid and multi-physics problem; the ability to capture the interaction of fluid and structure, where structure is rigid or elastic; handling of distorted and rapidly moving free surfaces without any limitation in topology (Violeau and Rogers, 2016).

The advantages and disadvantages of smoothed particle hydrodynamic methods are summarised as follows.

Positive points:

- In addition the merits mentioned above for the SPH method, its main advantage may be in application to certain complicated free surface problems, such as wave breaking, and sloshing, where modelling through mesh-based algorithms is difficult or even impossible.

Negative points:

- The main disadvantages of SPH methods can be summarised as follows (Violeau and Rogers, 2016)
    - high computational cost, particularly for 3D simulations,

- difficulties in applying boundary conditions at both close (zero- discharge) and open (inflow/outflow) boundaries,
- Lack of mathematical foundations for defining properties of numerical schemes, such as convergence and stability.
- difficult handling of variable space resolution for (near) incompressible flows.

c) Machine learning

In the past decades, numerous physics-based hydraulic models have been developed, which represent the hydraulic characteristics of flooding flows very accurately. However, when it comes to large scale problems, because of the computational cost, simpler but faster (albeit with much lower accuracy) models are practically preferred. A good example in this case relates to the National Water Model (NWM), a hydrologic and hydraulic model that predicts the weather effects on streams and simulates flood extent and water surface elevations for 2.7 million river reaches in the continental United States (Cohen et al., 2018; Hosseiny et al., 2020). Due to the extremely large number of tasks and computations required, this model (NWM) adopts a very simple hydraulic method, which relies on several simplifying assumptions, including a trapezoidal channel cross-section and quasi-normal flow regime (Maidment et al., 2016; Hosseiny et al., 2020). Such a model unfortunately cannot accurately capture the detailed variation of flow depths and velocities that is needed for many applications. In this type of applications, which a huge amount of tasks are required to accomplish, high-resolution simulations are only achieved by high-performance algorithms. Machine learning offers a potential approach to obtain high-performance algorithms, by improving our understanding from the real-world problems through learning from current data and experiments [e.g. data regression, and classification are two important tools of the machine learning technique (Hosseiny et al., 2020)]. This approach is typically used when (*i*) the current mathematical models are unable to capture physical phenomena, (*ii*) computational costs make it infeasible to adopt the accurate physical (herein hydraulic) models, and finally (*iii*) there is restricted knowledge about the phenomena (Hosseiny et al., 2020). In flood inundation studies, long simulations of certain cases (e.g. large-scale inundation problems) are the main reason motivating researchers to adopt the machine learning technique, as in the literature, there are numerous computational models capable of the accurate representation of flood propagation mechanics.

Several machine learning algorithms have been employed in water resources engineering, such as (see Mosavi et al., 2018) artificial neural networks (ANNs) (Kim et al., 2016), neuro-fuzzy (Mosavi and Edalatifar, 2018; Dineva et al., 2014), support vector machine (Suykens and Vandewalle, 1999), and support vector regression (Gizaw and Gan, 2016; Taherei Ghazvinei et al., 2018). Among all machine learning algorithms, ANNs that mimic biological neural networks are the most widely used for flood predictions (Mosavi et al., 2018). This algorithm, which was first used in the 1990s (Wu and Chau, 2010) provides reliable data-driven tools to construct block-box models for the complex, non-linear relationship between rainfall and flooding (Sulaiman and Wahab, 2018) and also river flow and discharge (Kar et al., 2010). Compared to conventional statistical models, the ANN algorithm delivers more accurate results (Li et al., 2010). Within the literature, there are many successful examples of ANN applications in different fields related to the flood, such as river flow (Shamseldin, 2010; Badrzadeh et al., 2013), rainfall–runoff (Smith and Eli, 1995), precipitation–runoff modeling (Taormina et al., 2012), water quality (Fotovatikhah et al., 2018), evaporation (Faizollahzadeh Ardabili et al., 2018), river stage prediction (Thirumalaiah and Deo, 1998), and river time series (Tsai and Yang, 2012).

Within the literature, the following shortcomings have been reported for the machine learning approach.

Positive points:

- Advantages of the machine learning approach include (Dataflair Team, 2019)
  - trends and patterns are easily identified,
  - the process of learning and improvement is automatically performed and no human intervention is required,
  - as further experiences are obtained by the algorithms, further improvement in accuracy and efficiency is gained,
  - its algorithms are capable of dealing with multi-dimensional and multi-variable data even though in dynamic and uncertain environments,
  - it has gained a wide range of applications in different fields because of its extraordinary performance.

Negative points:

- For the machine learning approach, a number of limitations are mentioned such as (Dataflair Team, 2019)
  - it requires massive sets of high quality and inclusive/unbiased data,
  - it may require much time and computer resources to enable the algorithm to become sufficiently trained that it delivers,

- particular expertise is required in interpreting the results. Also, an algorithm should be selected in accordance with the defined purpose,
- it is autonomous but susceptible to errors. When an algorithm is trained by a limited number of data sets, it may generate biased results.

1.3 Aim and objectives

In this thesis, a new model for large-scale flood inundation is developed, which is aimed at overcoming various weaknesses affecting currently available high performance models.

1.3.1 Research gap

Computational models based on the two-dimensional shallow water equations (SWE) have been employed for many decades to study a variety of free surface flow problems. Examples of such applications include flood inundation, river hydraulics, tidal hydrodynamics, and tsunamis. The SWE is a system of nonlinear hyperbolic equations, the solutions of which may display features such as shocks, rarefactions and transcritical flows (Toro, 2001) that may pose difficulties to simple numerical solution methods. Over the last decades, sophisticated numerical methods have been developed that are capable of capturing these features accurately and robustly. In particular, finite volume methods have gained significant popularity as a robust numerical approach for the solution of the SWE (e.g. Bermudez and Vázquez-Cendón, 1994; Toro, 2001; Brufau et al., 2004; Liang and Borthwick, 2009; Liang and Marche, 2009; Hou et al., 2015; Liang and Smith, 2015; Sanders and Schubert, 2019). Finite Volume methods handle discontinuities explicitly and ensure exact mass and momentum conservation. However, despite their robustness and accuracy, these models face important barriers when it comes to simulating large-scale flood inundation. The main challenge relates to the computational time that is required to solve these problems at grid resolutions that are needed for accurate solutions. This computational limitation also restricts our ability to perform probabilistic risk assessments, which require many scenarios to be modelled. In summary, current models do not have the required efficiency to simulate large-scale floods at levels of detail and accuracy that are relevant to decision makers (Fleischmann et al., 2019). Therefore, this research focuses on developing a high performance model for large-scale flood inundation problems. This model is defined on the basis of 2D shallow water equations solved in the framework of a Godunov type finite volume method. The proposed model is classified in the category of grid coarsening and sub-grid approach.

Substantial research efforts have been devoted to reduce the computational limitations of inundation models through the development of modelling techniques that offer different trade-offs between computational speed and accuracy. In previous sections, the main approaches have been reviewed. One such approach is the use of sub-grid parameterisations of the governing equations to approximate part of the physics operating within a large computational cell, so that the accuracy of simulations performed at a coarse resolution is improved. Because the computational cost t_c of explicit numerical schemes for the solution of the 2D SWE is very sensitive to the grid size (i.e., typically, $t_c \sim \Delta x^{-3}$), grid coarsening can deliver outstanding model speedups. For example, an increase in cell size by a factor of 10 would typically translate into a 1000x model speedup. However, simple grid coarsening also degrades the accuracy of numerical results, as a result of the growth of truncation errors. Of particular importance to the accuracy of such models is the misrepresentation of topographical details that are of primary importance to the propagation of the flood wave (Yu and Lane, 2006; Begnudelli et al., 2008). In the past, grid-coarsening techniques have been developed that incorporate important information at sub-grid scale –in particular fine-scale topographical data– into the model's solution procedure (e.g. McMillan and Brasington, 2007; Sanders et al., 2008) via the concept of porosity (Defina et al., 1994; Defina, 2000) . In particular, volumetric and area porosity parameters have been incorporated into the governing equations to account for the fact that: *i*) parts of computational cells may be blocked and not contribute to storage of mass and momentum, and *ii*) only a fraction of cell edges may contribute to fluxes (Sanders et al., 2008; Kim et al., 2015). Since first proposed by Defina et al. (1994), the use of porosity parameters to represent partially wet cells has received increased attention within the literature (Bates, 2000; Defina, 2000; Yu and Lane, 2011; Guinot and Soares-Frazão, 2006; Soares-Frazão et al., 2008; Sanders et al., 2008; Cea and Vázquez-Cendón, 2010; Guinot, 2012; Schubert and Sanders, 2012; Huang et al., 2014; Kim et al., 2015; Özgen et al., 2016a,b; Guinot, 2017; Bruwier et al., 2017; Guinot et al., 2017). Porosity models have mainly focused on the representation of buildings and other types of flow blockages that are typically found in urban areas (e.g. Soares-Frazão et al., 2008; Guinot, 2012) . More sophisticated versions of this concept have also been developed to include anisotropy properties of the porosity parameters, which stem from the arrangement of obstacles within the computational cells (Sanders et al., 2008; Guinot, 2012).

Despite their main advantages over simple (i.e. non-porosity) models, modelling sub-grid topography through porosity has a number of shortcomings (research gap). First, while modelling obstructed regions such as buildings by constant porosity parameters may realistically capture part of their effect on flood propagation, extending this idea to more complex topography (non-prismatic features) requires the definition of depth-dependent porosity parameters, which has been addressed in more recent studies (Guinot et al., 2018; Özgen et al., 2016a,b). Second, while porosity may correctly model part of the physics (in particular, storage capacity), other factors that are influenced by the irregular topography such as friction are not explicitly represented by porosity. This is because the conserved variables are averaged over the cell to become the arguments of the nonlinear function used to compute friction. Coarse computational cells may also display large variations in roughness, which may not be accurately modelled through simple averaging. A similar issue occurs when fluxes are estimated at the edges of two neighbouring computational cells. When large variations of topography occur at these edges, fluxes estimated by Riemann solvers from averaged quantities may be highly inaccurate. Also, the effects of bed slope computed based on micro-topography and averaged bed level (or the level of the terrain at the centroid of a cell) may differ substantially.

The research conducted in this PhD thesis aims to address the lack of rigorous approaches to upscale the shallow water equations. As it has been identified in Section 3.5, current sub-grid models developed for the solution of the SWE adopt simplifying methods that in general are not strongly rooted on the physics of the problem. Examples of this lack of physical (and/or mathematical) foundation are found in recent approaches proposed by Hénonin et al. (2015) (which adopts average quantities of variables) or the velocity distribution method proposed by Sanders and Schubert (2019) (the formulation of which leads to unphysical velocities tending to infinity as the depth of the flow tends to zero). The approach proposed in this PhD thesis addresses this gap through a formulation that is developed based on the integration of the governing equations over the domain of a coarse computational cell. As part of this approach, simplifying assumptions and mathematical approximations, while limited in number, are explicitly identified where they are adopted. This provides the reference needed for the validity of assumptions for modelling specific conditions to be judged, and to understand the order of the mathematical approximations used.

The proposed model aims to address the limitations of porosity-based approaches to model storage and boundary conveyance of a computational cell that includes non-prismatic sub-grid topographical features. This research also addresses the lack of rigorous methods to upscale friction and mass/momentum fluxes through the inclusion of topographical information within computational cells.

1.3.2 Aim and perspective of the study

The aim of this thesis is to develop a new sub-grid model that addresses the main shortcomings of other existing sub-grid methods (i.e. porous and nested meshes models). The model is based on the integral form of 2D shallow water equations and uses two nested meshes, coarse and fine (see Fig. 1.6). The governing equations are solved on the coarse mesh, leading to significant model speedup. A high-resolution mesh nested within the coarse grid is used to improve the accuracy of the coarse resolution model. This is achieved by defining parameters that represent, either exactly or approximately, the effects of finely resolved sub-grid topography on storage, momentum and friction.

The proposed model represents a significant improvement on current sub-grid models as follows. First, although porosity parameters are not explicitly defined in the model, the formulation resolves porosity effects (e.g. the effect of topography on storage) and anisotropy in a more generalised way that includes the intrinsic dependency on the free surface elevation. That is, the effects of porosity and anisotropy are not constant in time during the propagation of a flood wave, which is conceptually similar to dynamic porosity. Second, the proposed model implements a new technique to upscale the effects of fine-scale distribution of bed elevation and roughness on friction and fluxes across cell interfaces. Finally, the model improves the solution of the governing equations at partially wet cells using sub-grid, fine resolution data. Most of the above improvements require additional computational effort only at pre-processing level.

The model presented herein bears close similarities to other sub-grid models recently proposed in the literature. For example, Volp et al. (2013); Hénonin et al. (2015) and Sanders and Schubert (2019) also used a nested computational mesh, although their proposed methods to upscale each of the terms in the SWE are different to those presented in this research. Examples of upscaling methods used previously include simple averaging of depths (Hénonin et al., 2015), modelling the distribution of velocity according to a resistance law under the assumption of constant energy slope for the friction source term (Sanders and Schubert, 2019), or a combination of both (Volp et al., 2013). In this work, we define upscaling methods for each of the terms in the SWE based on integration of the governing equations over a coarse computational cell. The effects of sub-grid modelling choices adopted here are compared against other methods in the following chapters.

Table 1.1 lists similarities and differences between the proposed model and other approaches previously introduced. Furthermore, the properties of the proposed model as a version of sub-grid approach are compared with the main properties of other sub-grid models such as the model proposed by Sanders and Schubert (2019) in Table 1.2.

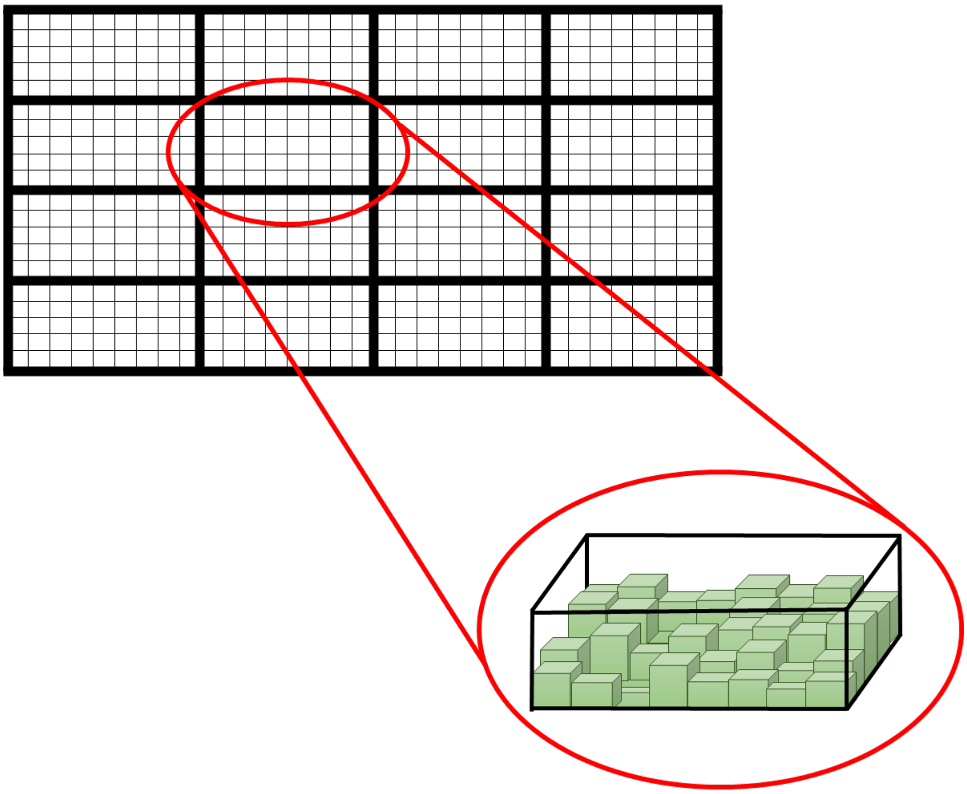


Figure 1.6: The two nested meshes of the proposed model. These meshes are related to a rectangular domain. The domain is covered by large and small cells. A large cell includes an integer number of small cells and the coarse and fine meshes share a common boundary. The refined mesh contains high-resolution topography and roughness data, which are not obtained from any hypothetical distribution function.

Table 1.1: A brief comparison between high performance approaches and the the proposed model.

Approaches	Full dynamic equations	Use of micro-topography	Partially wet cells	Further description
Simplified models	✗	✗	✗	- Simple formulation. - Unacceptable accuracy in some flow regimes and cases.
Parallel computing	✓/✗	✗	✗	- Slow simulation in certain cases - Complicated programming. - A refined mesh conventional model can be solved by parallel programming. In this case, the accuracy of the model is the highest compared to the other methods have been listed in this table.
Nested meshes	✓/✗	✓	✓/✗	- These models exploit multiple meshes with different resolutions. This provides further (relative to 1 mesh) detailed topographic data involved in the solution process. Thus, better description of the flow dynamics is achievable. - Nested meshes models use detailed topographic data for several purposes, such as calculation of net wet area of a cell blocked by obstacles. In the proposed sub-grid model, high-resolution data has further uses [e.g. micro topography is adopted in source terms of the governing equations (i.e. friction and bed slope) and flux calculations].

Approaches	Full dynamic equations	Use of micro-topography	Partially wet cells	Further description
Current porosity models	✓/✗	✓	✓	- These models are aimed at run-time reduction using grid coarsening. To avoid lack of accuracy, porosity parameters are incorporated into the governing equations. - Static porosity coefficients may not deliver a good interpretation of flow dynamics. However, recently, dynamic porosity models have attracted research interest (e.g. Guinot et al., 2018; Özgen et al., 2016a,b).
Adaptive mesh	✓/✗	✗	✗	- The efficiency is obtained using a refined mesh only at places with large changes in hydraulic properties and a coarse mesh for the rest of the domain. This leads to reduction in the number of cells in the domain. - Small cells, which appear within a small fraction of the study domain with high changes in hydraulic properties, may limit the time step (Liang and Smith, 2015).
Proposed model	✓	✓	✓	- Accuracy improvement is achieved by incorporating detailed topography and roughness data in the process of solution. - To avoid additional computational cost related to the inclusion of details, many computations are performed at the pre-processing level.

✓/✗ =Mixed versions.

Table 1.2: Comparison between the proposed model and current sub-grid (porosity and non-porosity) models.

Model	Effects of variations of topography within a computational cell on the computations of			Storage (S) and conveyance (C) changes of a computational cell in			
	Mass & momentum fluxes	Friction slope source term	Bed slope source term	Time		Space	
				S	C	S	C
Current porosity models	×[1]	✓[2]	×	✓/×	✓/×	✓/×	✓/×
Non-porosity models	✓[3]	✓[3]	✓[3]	✓	✓/×[4]	✓	✓/×[4]
Proposed sub-grid model	✓	✓	✓	✓	✓	✓	✓

1- Guinot et al. (2017) more recently has addressed this.

2- In current porosity models, friction effects are estimated for the submerged area of a cell (in a sense, the volumetric porosity computed through micro topography shows net wet area of a cell contributing in friction calculations). In the proposed model, further uses are exploited from the micro-topography, as follows. According to the Manning equation, friction is a function of water depth $\left(s_f = \frac{q^2 n_M^2}{h^{10/3}}\right)$. Current porosity models only use the average of water depth in a cell, but the proposed sub-grid model incorporates information on the distribution of water depth over all the wet small cells into the computations, which Herein, nonleads to higher accuracy in friction computations.

3- In the model of Hénonin et al. (2015), micro topography (variations of topography within a computational cell) is only used for accurate estimation of $\overline{h}$ in a computational cell. The obtained $\overline{h}$ contributes to the computations of the fluxes and friction and bed slope source terms.

4- In the model of Hénonin et al. (2015), the conveyance of a computational cell is not adopted in the process of solution.

* Herein, non-porosity model means a nested mesh model, such as Sanders and Schubert (2019) and Hénonin et al. (2015) which does not use any porosity parameter explicitly but make use of sub-grid data and have some similarities to the proposed model.

✓/× =Mixed versions.

1.3.3 Objectives

The proposed model exploits low-resolution meshes to speed up flood inundation simulations and in order to increase the accuracy of the coarse resolution model, high resolution data is included in the solution. The use of a nested mesh is a key difference with respect to the proposed solver compared to that of a conventional FV model. Then, a main requirement of this thesis is upscaling all terms of the governing equations from the high- to low- resolution mesh, on which the solution is proceed. Also, the proposed model requires methods that allow most of the computational cost associated with the inclusion of high-resolution information to be invoked solely at the processing stage. Such computations mostly relate to the estimation of parameters that depend on water surface level. The thesis objectives are therefore as follows:

- Definition of a mechanism to estimate the mass and momentum fluxes in the framework of a Godunov FV method;
- Treatment of friction slope source term;
- Treatment of bed slope source term;
- Development of an efficient pre-processing stage;
- Verification and validation of the proposed model against a number of analytical solutions and field data;
- Assessment of the performance of the proposed model in terms of accuracy and speedup.

1.4 Synopsis

Chapter 1 has reviewed the main approaches used to improve the performance of computational models applied to large-scale flood simulation, identified the research gap, and listed the aim and objectives of the thesis. Chapter 2 describes the fundamental concepts and methods behind the proposed model and/or required for understanding later chapters, including the finite volume method, the HLLC Riemann solver, wetting-drying conditions, boundary conditions, and the inverse method used to produce analytical solutions to the governing equations under 1D steady flow. Chapter 3 explains the methodology underpinning the proposed sub-grid model. This chapter presents the governing shallow water equations and the methods used to upscale different terms (including bed and friction slope source terms, and mass and momentum fluxes at cells interface). The structure of the proposed model is illustrated through a flowchart that displays the hierarchical process used to accomplish the cycle of the computations at a given time level. Chapter 4 presents and interprets the results of numerical simulations performed to verify and test the accuracy and computational performance of the model. Benchmark case studies are presented along with real-world test cases. Chapter 5 lists the main conclusions of the work and suggests further research directions arising from the thesis.

Chapter 2

Godunov-type Finite Volume Techniques for Solving the Shallow Water Equations

This chapter outlines and reviews Godunov type finite volume methods used to solve the 2D shallow water equations, and provides the context behind the model proposed in the thesis. Central to the finite volume method is the approximation of mass and momentum fluxes at the interface of each two neighbouring computational cells, which is performed through the solution of a Riemann problem (The Riemann problem is an initial value problem whereby the initial condition consists of a single discontinuity within the domain.) Although the exact solution of the Riemann problem is mathematically available within the literature, for practical purposes, approximate Riemann solvers are generally used because of their improved computational performance (see e.g. Toro, 2001; LeVeque, 2002). This chapter compares the most popular Riemann solvers within the literature, and gives specific details of the HLLC scheme used in the proposed model, along with the different types of boundary condition

For verification purposes, in Chapter 4 simulations by the proposed model are compared against analytical solutions for certain benchmark test problems. The second test (in section 4.2) comprises the propagation of a hydraulic bore from a dam break on a wet or a dry horizontal bed. These cases examine the ability of the scheme to model a rapidly evolving unsteady transcritical flow. The fifth test relates to steady state flow in a prismatic rectangular channel, for which inverse analytical solutions have been derived by MacDonald (1996). This method enables a large range of steady state conditions to be considered for key parameters including spatially varying cross-sections and bed roughness coefficients.

Many real world and even artificial problems deal with wetting-drying (i.e. the moving of a wet front over a dry bed and vice versa). Under such circumstance, computational models must take a set of measures into the consideration to prevent negative water depth or unphysically large velocities. Section 2.6 describes the wetting and drying scheme used in the proposed model.

In numerical modelling, it is required to ensure that governing equations are being solved correctly and the solution is insensitive to grid resolutions. In computational fluid dynamic grid convergence index is a popular method to determine discretisation error bands and analysing mesh dependency. In the current chapter, this method is introduced in brief.

2.1 Numerical methods for solving shallow water equations

In recent decades, the 2D shallow water equations have, for many problems in engineering and science, emerged as the best trade-off between acceptable interpretation of the physics of free surface flows and the complexity of the solver. Unsurprisingly, the SWEs have gained widespread popularity in research and practical engineering (Alcrudo and García-Navarro, 1993; Zhao et al., 1996; Zoppou and Roberts, 2000; Mignot et al., 2006; Liang et al., 2008; Wang et al., 2011b; Liang, 2012; Sanders and Schubert, 2019). In the present research, theses equations are adopted as the governing equations. In Appendix A, the derivation of 2D SWE from the Navier-Stokes equations is presented. The integral form of 2D SWE reads as

$$\frac{\partial}{\partial t}\int_{\Omega} \boldsymbol{U} \mathrm{d}\Omega + \oint_{\Gamma} [\boldsymbol{E}(\boldsymbol{U}) \cdot \boldsymbol{e}] \mathrm{d}\Gamma = \int_{\Omega} \boldsymbol{S}(\boldsymbol{U}) \mathrm{d}\Omega, \tag{2.1}$$

where,

$$\boldsymbol{U} = \begin{pmatrix} h \\ q_x \\ q_y \end{pmatrix}, \quad \boldsymbol{E} = [\boldsymbol{F}, \boldsymbol{G}], \quad \boldsymbol{F}(\boldsymbol{U}) = \begin{pmatrix} q_x \\ \frac{q_x^2}{h} + \frac{g}{2}h^2 \\ \frac{q_x q_y}{h} \end{pmatrix}, \quad \boldsymbol{G}(\boldsymbol{U}) = \begin{pmatrix} q_y \\ \frac{q_x q_y}{h} \\ \frac{q_y^2}{h} + \frac{g}{2}h^2 \end{pmatrix}, \tag{2.2}$$

$$\boldsymbol{S}(\boldsymbol{U}) = \boldsymbol{S}_b(\boldsymbol{U}) + \boldsymbol{S}_f(\boldsymbol{U}), \quad \boldsymbol{S}_b(\boldsymbol{U}) = \begin{pmatrix} 0 \\ ghs_{0_x} \\ ghs_{0_y} \end{pmatrix}, \quad \boldsymbol{S}_f(\boldsymbol{U}) = \begin{pmatrix} 0 \\ -ghs_{f_x} \\ -ghs_{f_y} \end{pmatrix}, \tag{2.3}$$

where, q_x and q_y are the x and y components of unit width discharges, respectively, $\boldsymbol{e} = \left[e_x, e_y\right]$ is the unit vector normal to boundary Γ. Since the equations are nonlinear, in order to solve the system, a robust numerical model is required. Over the past decades, different classes of numerical methods have been used to solve shallow water equations, such as finite volume, finite element and finite difference.

The main merit of the finite difference (FD) method is its relative simplicity in reaching a high order of accuracy. However, this may come at the cost of long simulation time, (Hernández, 2014). On the other hand, research has shown that the method of FD has a number of shortcomings. For instance, FD models result in complicated formulations when the domain involves complex geometries (Hernández, 2014). However, the main limitation of the finite difference method is its inability to solve discontinuous problems, or even problems where the solution displays large gradients. This issue arises from the fact that FD schemes are based on the differential form of shallow water equations and derivatives are not defined at discontinuities. However, this issue can be relatively addressed by the inclusion of a Riemann solver in a FD model. It is also worth mentioning that it is not always possible to obtain a conservative scheme by using FD models.

The Finite Volume (FV) method is straightforward to implement on complicated (i.e. unstructured) meshes. FV methods are based on closed cells of finite area (in two dimensions) which function as a control volume that exchanges mass and momentum with its neighbours. The solution evolves in time by applying principles of mass and momentum conservation to the cell, which is followed by averaging of the conserved variables. The method deals with discontinuous solutions explicitly and is able to capture shocks. In fact, it solves discontinuities (Riemann Problem) even when the solution is continuous. It also guarantees conservation, as fluxes leaving any cell go necessarily to the neighbouring cells.

The Finite Element (FE) method depends on a variational formulation (Zienkiewicz and Taylor, 2000) of the motion equation. Although the FE method has a rigorous mathematical base, which restricts error estimation, and is well suited to complex geometries and irregular meshes, its complex concepts and greater computational burden relative to the finite volume method has led to fewer researchers selecting the FE method for flooding studies (Alcrudo, 2002). It can also be proved that some finite element methods are equivalent to the FV (Achdou and Bernardi, 2001; Alcrudo, 2002).

Because of the aforementioned advantages, finite volume schemes of the Godunov type, the diversity of which occurs through the scheme dimensions (1D, 2D or 3D), order of accuracy of the scheme, type of Riemann solver (further expanded later), and treatment of source terms, have become a standard numerical tool in free surface modelling, (see for example Begnudelli and Sanders, 2007; Bradford and Sanders, 2002; Brufau et al., 2004; Hubbard and Dodd, 2002; Kesserwani and Liang, 2012b) .

For a quadrilateral cell, the integral form of the two-dimensional shallow water equations (i.e. Eqs. 2.1 to 2.3) can be discretised as below using the first order Euler method, (Soares-Frazão and Zech, 2002),

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n - \frac{\Delta t}{\Omega_i} \sum_{m=1}^{m=4} \left[\boldsymbol{E}_m(\boldsymbol{U}_i^n) \cdot \boldsymbol{e}_m \Gamma_m \right] + \Delta t \boldsymbol{S}_b(\boldsymbol{U}_i^n) \tag{2.4}$$

where subscript i denotes the cell index, n is the time level, Ω_i is area of i^{th} cell, Δt is the time step, m denotes the number of cell edges (or interfaces with other cells), Γ_m is the length of m^{th} edge of cell. According to (Godunov, 1959), the conservative variables are approximated constant (in first order models) and interface fluxes (i.e. $\boldsymbol{E}_m(\boldsymbol{U}_i^n) \cdot \boldsymbol{e}_m$ are computed by solving the Riemann problem. In the literature, one of the most popular and versatile Riemann solvers is the HLLC solver, which is detailed in the following section.

2.2 HLLC scheme

The estimation of numerical fluxes (i.e. $\boldsymbol{E}_m(\boldsymbol{U}_i^n) \cdot \boldsymbol{e}_m$ in Eq. 2.4) has considerable influence on simulation accuracy. In Godunov type finite volume schemes, the fluxes are approximated by solving the Riemann problem at each cell interface, (Godunov, 1959). By means of a simple expression, a Riemann problem consists of conservation equations with discontinuous initial conditions at a point, which can be the interface between neighbouring cells as shown in Fig. 2.1, so that before and after this point, the conservative variables are approximated as constant, (Godunov, 1959). The exact solution to the Riemann problem may impose heavy time burden on a computational model (Toro, 2001). Thus, approximate solutions for the Riemann problem have been developed.

In recent decades, significant research has been conducted in the context of approximate Riemann solvers, with considerable focus on the shallow water equations. Amongst these, Roe, HLL, Osher, and HLLC schemes have been evaluated largely using idealised benchmark test cases and field-scale shallow water simulations, (Bermudez and Vázquez-Cendón, 1994; Toro, 2001; Brufau et al., 2004; Zhou et al., 2004; Loukili and Soulaimani, 2007; Song et al., 2011; Huang et al., 2013). Erduran et al. (2002) conducted a study to compare these Riemann solvers in terms of accuracy, applicability, numerical stability, and run-time. They found that HLLC (Harten, Lax and van Leer Contact), which was developed by Toro et al. (1994) is highly recommended as a solver, especially given that it is suitable for all kinds of applications. A similar study was conducted by Kong (2011), which compares four popular Riemann solvers, namely Roe, HLL, HLLC, and Osher-Solomon. The Riemann solvers were tested against a comprehensive variety of test cases to assess the solvers' accuracy and robustness. The study does not start from the premise that any Riemann solver is perfect; however, the HLLC scheme did show an outstanding performance compared to the other schemes considered. The negative

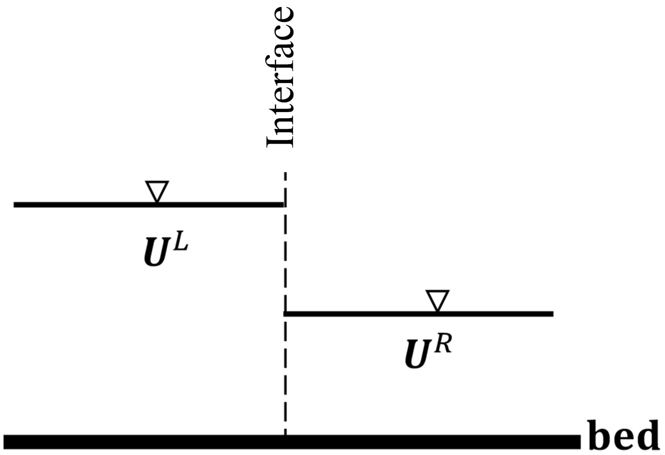


Figure 2.1: Riemann problem conditions at the interface between neighbouring cells. Conservative variables are specified by $\boldsymbol{U}^L$ and $\boldsymbol{U}^R$ for the left and right cells either side of the interface, respectively. $\boldsymbol{U}^L$ and $\boldsymbol{U}^R$ are assumed constant in the vicinity of the interface.

point reported by the study for the HLLC scheme is that in certain test cases, the HLLC results display more diffusion than those Roe's scheme. This higher diffusion may however be neglected in large-scale flood inundation problems, the subject of this thesis, given that highly accurate prediction of shocks and surges is not critical in flood simulation. Herein, the HLLC Riemann solver, which is adopted in this thesis, is detailed.

In the Riemann solver, HLLC with the solver structure shown in Fig. 2.2, term $\boldsymbol{E}_m(\boldsymbol{U}_i^n) \cdot \boldsymbol{e}_m$ of Eq. 2.4 is estimated by (see Toro et al., 1994; Hou et al., 2013),

$$\boldsymbol{E}_m(\boldsymbol{U}_i^n) \cdot \boldsymbol{e}_m = \begin{cases} \boldsymbol{E}^L & \text{if} \quad 0 \le S^L \\ \boldsymbol{E}_*^L & \text{if} \quad S^L < 0 \le S^M \\ \boldsymbol{E}_*^R & \text{if} \quad S^M < 0 \le S^R \\ \boldsymbol{E}^R & \text{if} \quad S^R < 0 \end{cases} \tag{2.5}$$

where, superscripts L and R specify the variables at the left and right sides of the m^{th} interface of cell i, respectively, $\boldsymbol{E}^L = \boldsymbol{F}(\boldsymbol{U}^L) \cdot e_{x_m} + \boldsymbol{G}(\boldsymbol{U}^L) \cdot e_{y_m}$ and $\boldsymbol{E}^R = \boldsymbol{F}(\boldsymbol{U}^R) \cdot e_{x_m} + \boldsymbol{G}(\boldsymbol{U}^R) \cdot e_{y_m}$ are defined based on Eqs2.1 and 2.2, and E_*^L and E_*^R are detailed in the following paragraphs, S^L, S^M and S^R are the left, middle (contact) and right wave speed, respectively and calculated by

$$S^L = \begin{cases} u^{\perp R} - 2\sqrt{gh^R} & \text{if} \quad h^L = 0 \\ \min\left(u^{\perp L} - \sqrt{gh^L}, u_*^{\perp} - \sqrt{gh_*}\right) & \text{if} \quad h^L > 0 \end{cases} \tag{2.6}$$

$$S^R = \begin{cases} u^{\perp L} + 2\sqrt{gh^L} & \text{if} \quad h^R = 0 \\ \max\left(u^{\perp R} - \sqrt{gh^R}, u_*^{\perp} + \sqrt{gh_*}\right) & \text{if} \quad h^R > 0 \end{cases} \tag{2.7}$$

$$S^M = \frac{S^L h^R \left(u^{\perp R} - S^R\right) - S^R h^L \left(u^{\perp L} - S^L\right)}{h^R \left(u^{\perp R} - S^R\right) - h^L \left(u^{\perp L} - S^L\right)}, \tag{2.8}$$

in which, $u^{\perp} = ue_{x_m} + ve_{y_m}$ is the normal velocity to the m^{th} interface and h_* and $u_*^{\perp}$ are expressed as

$$h_* = \frac{1}{g}\left[\frac{1}{2}\left(\sqrt{gh^L} + \sqrt{gh^R}\right) + \frac{1}{4}\left(u^{\perp L} - u^{\perp R}\right)\right]^2, \tag{2.9}$$

$$u_*^{\perp} = \frac{1}{2}\left(u^{\perp L} + u^{\perp R}\right) + \sqrt{gh^L} - \sqrt{gh^R}, \tag{2.10}$$

$\boldsymbol{E}_*^L$ and $\boldsymbol{E}_*^R$ are defined as

$$\boldsymbol{E}_*^L = \begin{pmatrix} \boldsymbol{E}_{*_1} \\ \boldsymbol{E}_{*_2} e_x - u^{\parallel L} \boldsymbol{E}_{*_1} e_y \\ \boldsymbol{E}_{*_2} e_y + u^{\parallel L} \boldsymbol{E}_{*_1} e_x \end{pmatrix}, \tag{2.11}$$

$$\boldsymbol{E}_*^R = \begin{pmatrix} \boldsymbol{E}_{*_1} \\ \boldsymbol{E}_{*_2} e_x - u^{\parallel R} \boldsymbol{E}_{*_1} e_y \\ \boldsymbol{E}_{*_2} e_y + u^{\parallel R} \boldsymbol{E}_{*_1} e_x \end{pmatrix}, \tag{2.12}$$

where, $u^{\parallel} = -ue_{y_m} + ve_{x_m}$ is the tangential velocity to the m^{th} interface and $\boldsymbol{E}_* = \left[\boldsymbol{E}_{*_1}, \boldsymbol{E}_{*_2}\right]$ is

$$\boldsymbol{E}_* = \frac{S^R \boldsymbol{E}\left(\boldsymbol{U}^{\perp L}\right) - S^L \boldsymbol{E}\left(\boldsymbol{U}^{\perp R}\right) + S^R S^L \left(\boldsymbol{U}^{\perp R} - \boldsymbol{U}^{\perp L}\right)}{S^R - S^L}, \tag{2.13}$$

in which, $\boldsymbol{U}^{\perp R} = \left[h, q_x e_{x_m} + q_y e_{y_m}\right]$ and

$$\boldsymbol{E}\left(\boldsymbol{U}^{\perp}\right) = \begin{pmatrix} hu^{\perp} \\ u^{\perp}\left(q_x e_{x_m} + q_y e_{y_m}\right) + \frac{1}{2} g h^2 \end{pmatrix}, \tag{2.14}$$

2.3 Boundary conditions

In a finite volume scheme, boundary conditions may be implemented through ghost cells, defined adjacent to actual boundary cells. In the light of this, reflective (closed) and transmissive (open

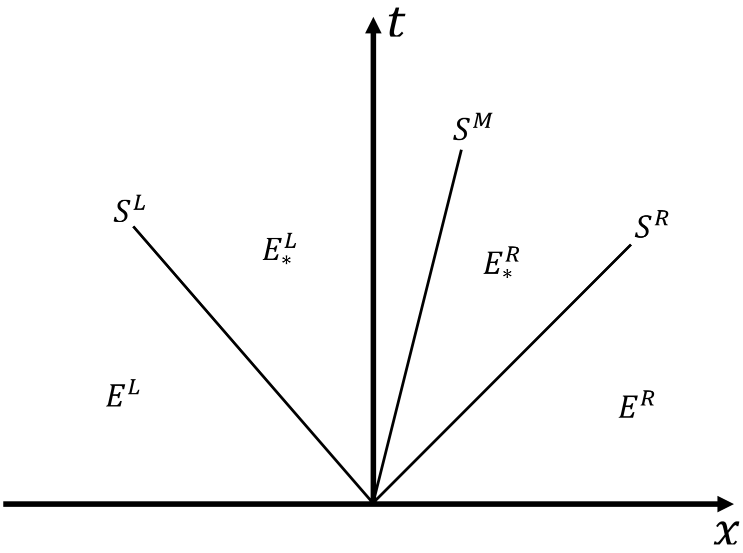


Figure 2.2: The structure of HLLC Riemann solver.

inflow/outflow) boundary conditions may be expressed as follows. Assume a typical cell of a 2D Cartesian mesh having four edges labelled Γ_1 to Γ_4 anticlockwise from the southern edge; for each of which at the boundary, it is defined what values of conservative variables are assigned to the ghost cells (i.e. h^G, q_x^G and q_y^G) based on the known variables h^B, q_x^B and q_y^B, where superscripts G and B denote the variables values at ghost and inner boundary cells, respectively.

a) Reflective boundary conditions

This boundary condition applies to reflection at a solid barrier (e.g. solid wall, where the boundary acts as a mirror. This means that the velocity component normal to the boundary is reflected and water depth and the velocity component tangential to the boundary is maintained unchanged, as indicated in Table2.1.

Table 2.1: Values of conservative variables in a ghost cell in reflective boundary conditions.

Edge	h^G	q_x^G	q_y^G
Γ_1 & $\Gamma 3$	h^B	q_x^B	$-q_y^B$
Γ_2 & $\Gamma 4$	h^B	$-q_x^B$	q_y^B

b) Transmissive boundary conditions

Transmissive boundary conditions are implemented through Riemann invariants depending on the flow direction (i.e. inflow/outflow) and regime (supercritical, critical and subcritical) of flow passing into the boundary that can be described in three ways, namely: (b1) Subcritical inflow/outflow and critical outflow, (b2) Supercritical and critical inflow and (b3) Supercritical outflow.

b1) Subcritical inflow/outflow and critical outflow

Table 2.2 describes the transmissive boundary condition for the case of Subcritical inflow/outflow and critical outflow in detail.

Table 2.2: Values of conservative variables in a ghost cell under the transmissive boundary conditions: the case of subcritical inflow/outflow and critical outflow

Edge	Theory of characteristics	Subcritical inflow			Subcritical and critical outflow		
		q_x^G	q_y^G	h^G	q_x^G	q_y^G	h^G
Γ_1	$\frac{q_y^G}{h^G} - 2\sqrt{gh^G} = \frac{q_y^B}{h^B} - 2\sqrt{gh^B}$	$\frac{q_x^B}{h^B}h^G$	PSCR	*	$\frac{q_x^B}{h^B}h^G$	**	PSCR
Γ_2	$\frac{q_x^G}{h^G} + 2\sqrt{gh^G} = \frac{q_x^B}{h^B} + 2\sqrt{gh^B}$	PSCR	$\frac{q_y^B}{h^B}h^G$	*	**	$\frac{q_y^B}{h^B}h^G$	PSCR
Γ_3	$\frac{q_y^G}{h^G} + 2\sqrt{gh^G} = \frac{q_y^B}{h^B} + 2\sqrt{gh^B}$	$\frac{q_x^B}{h^B}h^G$	PSCR	*	$\frac{q_x^B}{h^B}h^G$	**	PSCR
Γ_4	$\frac{q_x^G}{h^G} - 2\sqrt{gh^G} = \frac{q_x^B}{h^B} - 2\sqrt{gh^B}$	PSCR	$\frac{q_y^B}{h^B}h^G$	*	**	$\frac{q_y^B}{h^B}h^G$	PSCR

* Obtained by solving the equations written in the table via methods such as Newton-Raphson.

** Obtained by solving the equations written in the table.

PSCR=prescribed.

b2) Supercritical and critical inflow

All conservative variables h^G, q_x^G and q_y^G are prescribed.

b3) Supercritical outflow

Values of the real boundary cells are duplicated for ghost cells (i.e. $h^G = h^B, q_x^G = q_x^B, q_y^G = q_y^B$).

Reconstructing conservative variables at ghost cells, the conservative variables are updated using Eq. 2.4 with the following consideration. At reflective boundaries, the boundary is treated as a normal edge within the domain (i.e. treated as an edge, which is far from is the boundary) and fluxes (i.e. term $\boldsymbol{E}_m\left(\boldsymbol{U}_i^n\right) \cdot \boldsymbol{e}_m$ of Eq. 2.4) are computed according to Eq. 2.5, so that conservative variable values at the both sides of the edge (interface) are those assigned to the ghost cell (h^G, q_x^G and q_y^G) and interior cell (h^B, q_x^B and q_y^B). For transmissive boundaries, term $\boldsymbol{E}_m\left(\boldsymbol{U}_i^n\right) \cdot \boldsymbol{e}_m$ of Eq. 2.4 instead of Eq. 2.5 is computed based only on the ghost cells as follows,

$$\boldsymbol{E}_m\left(\boldsymbol{U}_i^n\right)\cdot\boldsymbol{e}_m = e_{x_m}\begin{pmatrix} q_x^G \\ \frac{\left(q_x^G\right)^2}{h^G}+\frac{1}{2}g\left(h^G\right)^2 \\ \frac{q_x^G q_y^G}{h^G}\end{pmatrix} + e_{y_m}\begin{pmatrix} q_y^G \\ \frac{q_x^G q_y^G}{h^G} \\ \frac{\left(q_y^G\right)^2}{h^G}+\frac{1}{2}g\left(h^G\right)^2\end{pmatrix}, \quad m = 1,2,3,4. \tag{2.15}$$

2.4 Dam-break problem

Dam-break is a well-established benchmark problem in open channel computational hydraulics and is important for the following reasons. First, shock and rarefaction waves are a part of the solution of the dam-break problem that can be only captured accurately by robust and sophisticated computational models. Second, the dam-break problem is an unsteady test case, for which there exists an analytical solution to the shallow water equations. Consequently, many SWE-based computational models have been verified against solutions of this benchmark test. The literature reports many dam-break studies in which different factors are analysed such as the dimension of the problem (1D, 2D and 3D), friction (flow propagation over the frictionless or frictional bed), bed form (flat, even or discontinuous bed), and the downstream conditions in terms of being wet or dry (see e.g. Alcrudo and Benkhaldoun, 2001; García-Navarro and Vázquez-Cendón, 2000; Brufau and García-Navarro, 2000; Benkhaldoun and Seaïd, 2010; Liu et al., 2018; Hu et al., 2018; Khoshkonesh et al., 2019). Herein, the widely used form of the dam-break problem (i.e. flow propagation over a flat, frictionless bed, where the flow conditions are initially still) is reviewed for both wet and dry initial downstream bed conditions.

a) Initially wet downstream bed

For the assumed dam-break problem, based on the characteristic curves method, the SWE's solution includes a downstream shock wave and upstream rarefaction wave. Fig. 2.3 illustrates the initial conditions and a schematic solution to the governing equations. According to Fig. 2.3, the domain of study may be split into four zones (I to IV). Zones I and IV are those have not been yet influenced by the waves for which the initial conditions are valid. Zone II relates to a part of the domain, which is experiencing the rarefaction wave and in zone III, the waves have passed the zone and steady flow conditions have been set.

The velocity of the steady state flow at the moving front (i.e. zone III) is a function of its water depth h, and the downstream initial water depth h^R and written as (LeVeque, 2002)

$$u = \sqrt{\frac{g}{2}\left(h - h^R\right)^2\left(\frac{1}{h}+\frac{1}{h^R}\right)}, \tag{2.16}$$

Identically, the flow velocity at each point on the rarefaction wave (i.e. zone II) relies on the two variables h^L (upstream initial water depth) and h as (LeVeque, 2002)

$$u = 2\left(\sqrt{gh^L} - \sqrt{gh}\right). \tag{2.17}$$

The solution of simultaneous equations 2.16 and 2.18 gives the water depth and velocity in the zone II. The positions of the waves at any time depends on their wave speeds. The speed of the shock front is

$$V_s = \sqrt{\frac{g}{2}\left(h + h^R\right)\left(\frac{h}{h^R}\right)}. \tag{2.18}$$

The speed of the smoothly, curving rarefaction wave is $\sqrt{gh}$. This means that the wave's speed is not constant along the rarefaction wave as h varies, so that the highest speed belongs to the farthest upstream point of the rarefaction wave moving at the speed of $\sqrt{gh^L}$.

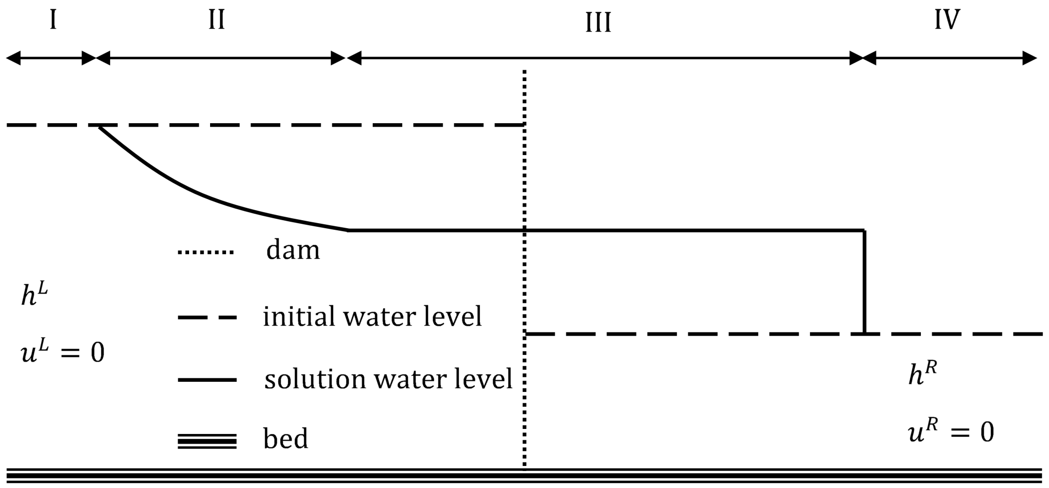


Figure 2.3: The solution mechanism of the dam-break problem for an initially wet downstream bed.

b) Initially dry downstream bed

In the case of an initially dry downstream bed, the solution to the dam-break problem contains a rarefaction wave propagating upstream and downstream as shown in Fig. 2.4. The first analytical solution to this type dam-break problem (i.e. initially dry downstream bed) was first presented by Ritter (1892), in which the water depth and flow velocity at each point of the rarefaction wave (i.e. zone II) are obtained as

$$h = \frac{1}{9g}\left(2\sqrt{gh^L} - \frac{x}{t}\right)^2, \tag{2.19}$$

$$u = \frac{2}{3}\left(\sqrt{gh^L} + \frac{x}{t}\right). \tag{2.20}$$

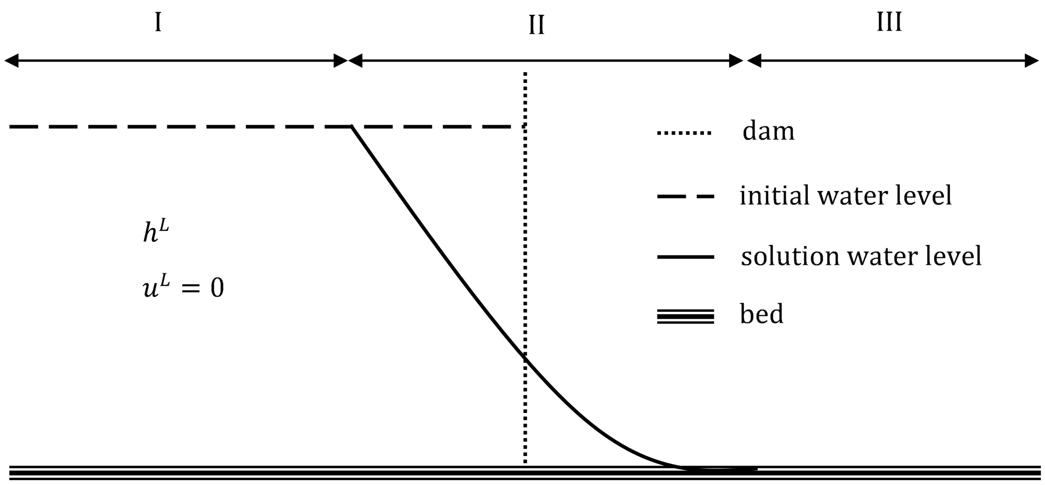


Figure 2.4: The solution mechanism of the dam-break problem for an initially dry downstream bed.

2.5 Inverse method for a prismatic rectangular channel

In many free surface flow problems involving the shallow water equations, there is no exact solution and numerical modelling is the only tool to approximate the exact solution for these problems. Thus, analytical solutions of standard benchmark tests are very useful to evaluate the accuracy of numerical models.

MacDonald (1996) presented a simple method to design 1D steady free surface flow problems whereby their analytical solutions are derived from the shallow water equations. The method is called an inverse method because, compared to typical free surface flow problems, the reverse procedure is followed to find the solution. In typical hydraulic channel problems, the channel bed characteristics such as bed slope and cross-section geometry are already known, and the water surface profile is computed from inflow/outflow discharges. In MacDonald's inverse method, first the water depth solution, flow discharge (constant), and channel cross-section geometry (varied or fixed) are assumed for the channel and then the corresponding longitudinal bed profile is computed. Herein, an inverse method for prismatic rectangular channels is presented.

According to the literature (Henderson, 1966), the following expression may be derived for gradually varied flow in an irregular prismatic channel [the following expression is a special case of the general expression presented and derived in Appendix B, corresponding to 1D steady free surface flow channels (non-prismatic channels)],

$$\frac{\mathrm{d}h}{\mathrm{d}x} = \frac{s_0 - s_f}{1 - Fr^2}, \tag{2.21}$$

where, $Fr = \sqrt{\frac{Q^2 T}{gA^3}}$ is the Froude number and Q, T and A are respectively, discharge, channel width and cross-section area.

According to Eq. 2.21, the bed profile is obtained as

$$s_0 = \left[1 - Fr^2\right] \frac{\mathrm{d}h}{\mathrm{d}x} + s_f, \tag{2.22}$$

where, s_f (friction slope) can be approximated using the Manning coefficient,

$$s_f = \frac{Q^2 n_M^2}{A^2 R^{\frac{4}{3}}}, \tag{2.23}$$

in which, R is the hydraulic radius and equals the ratio of cross-section area to cross-section wet perimeter.

For a wide rectangular channel, where $R \cong h$ is assumed, Eq. 2.22 is converted into the following formulation,

$$s_0(x) = \left[1 - \frac{Q^2}{gh(x)^3 T^2}\right] \frac{\mathrm{d}h}{\mathrm{d}x} + \frac{Q^2 n_M^2}{h(x)^{\frac{10}{3}} T^2}. \tag{2.24}$$

Since by definition $s_0(x) = -\frac{\mathrm{d}z}{\mathrm{d}x}$, then the bed profile for the assumed water depth solution and given discharge is easily obtained as

$$z = z_0 - \int_0^x s_0(x)\mathrm{d}x, \tag{2.25}$$

where, z_0 is the channel bed level at $x = 0$.

2.6 Wetting and drying

To simulate an actual flood inundation problem, a computational model should deal with wetting and drying conditions. In practice, an inaccurate representation of the wet-dry problem may lead to the appearance of a negative water depth and unphysical high velocities around a wet-dry front. This might smear out the model performance and even result in numerical instability, (Wang, 2011).

Moreover, mass and momentum conservation is (both locally and globally) central in addressing the wetting-drying problem.

According to Medeiros and Hagen (2013) [similar discussion is also available in Le et al. (2020) and Martins et al. (2018)], wetting-drying algorithms can be categorised in four classes, namely element removal, thin layer, depth extrapolation, and negative depth so-called artificial porosity.

In the *element removal* method, wet and dry cells are detected. The dry cells are excluded from the computation process and the solution is only performed for the wet cells. This algorithm may involve the failure of mass and momentum conservation, and introduce numerical instability (Kärnä et al., 2011).

The *thin layer* method comprises a wetting-drying algorithm in which cell water depths are compared against a pre-defined threshold water depth. Any cell with a water depth less than the threshold is assumed dry, and its velocity is then set to zero. The main advantage of the algorithm is that it is easily applied to 2D and 3D problems (Gourgue et al., 2009). Two major shortcomings are reported for this method within the literature. The threshold depth may manipulate the water volume in the domain and affect mass conservation. Around the wetting front, the algorithm requires rapid transfer of cell states from dry to wet, restricting the time step in an explicit scheme and lowering the accuracy of non-linear systems solved by an implicit scheme (Le et al., 2020).

In *depth extrapolation* algorithms, a water depth is attributed to a dry cell, the value of which is obtained from an extrapolation implemented using the water depths of the neighbouring wet cells. The attributed water depth is then adopted in evaluating the velocity in the cell. The dry cell is now ready to be integrated into the wet domain. The artificial wetting imposed on dry cells is followed by artificial leakage between adjacent cells (Bradford and Sanders, 2002) that affects mass conservation, the control of which requires corrective schemes (Begnudelli and Sanders, 2006).

Negative depth algorithms are allowed to have cells with negative water depth controlled through porosity terms. In this method, wet and dry cells are not separately treated (Gourgue et al., 2009) and wet-dry transitions are smooth (Kärnä et al., 2011). However, the method has several disadvantages, such as the violation of mass conservation stemming from mass transferring from dry to wet areas (Martins et al., 2018; Heniche et al., 2000), and appearance of unphysical properties in the fictitious porous layers.

Herein, some important wetting-drying numerical treatments are reviewed. Toro et al. (1994) introduced the *modified wave speed* approach, in which the wave speed calculated by the HLL approximate Riemann solver, is modified at cell interfaces near the dry front. The method was adopted by Fraccarollo and Toro (1995) and evaluated against an experimental dam-break case, the results of which were very promising. Hu et al. (2000) later presented a simpler approach, referred to as minimum wet depth in the literature, which redefines the dry state when water depth is less than a tolerance depth (10^{-6}), in which wave speed modification is prevented. This method has been applied successfully in simulations of wave overtopping of coastal structures (Wang, 2011).

A wetting-drying treatment was presented by Brufau et al. (2002) and Brufau et al. (2004) for steady and unsteady flows, respectively, based on local bed modification, and the technique is implemented using the upwind discretisation of bed slope terms with a particular focus on well-balancing. In the proposed method, negative water depth is unavoidable; however a controlling approach is adopted to eliminate the mass error. A drawback of the technique may be the violation of momentum conservation.

Audusse et al. (2004) presented a computational model that successfully handles wetting-drying problem using a well-balanced hydrostatic reconstruction in which non-negative water depth is guaranteed without artificially imposing zero velocity and water depth in the model. The method was later followed by Marche et al. (2007)) and outstanding results were obtained. In the light of Audusse et al. (2004), and (Marche et al., 2007), Liang and Marche (2009) developed a sophisticated technique to maintain the numerical stability, while prevents non-negative water depth reconstruction at wetting-drying interface. The main features of the method are efficiency and robustness in a wide range of applications. In this PhD research this set of methods, which are widely used (see e.g. Wang et al., 2011a; Kesserwani and Liang, 2010; Wu et al., 2018) is adopted as follows.

Denoting the hydraulic and geometric variables of two cells at both sides of an interface by superscripts L and R (see Fig. 2.5), the following procedure comprises the wetting-drying treatment. In the following steps, the variables, such as η, q_x and q_y, at the interface are the reconstructed by a finite volume scheme. The new modified variables arising from the wet-dry procedure are adopted to estimate the Riemann state of numerical fluxes.

$$
\begin{aligned}
&\text{a)} \quad z^{\wedge} = \max\left(z^{L}, z^{R}\right), \\
&\text{b)} \quad \begin{cases} u^{L} = \dfrac{q_x^L}{\max\left[10^{-5},\left(\eta^L - z^L\right)\right]}, \quad \upsilon^{L} = \dfrac{q_y^L}{\max\left[10^{-5},\left(\eta^L - z^L\right)\right]} \\ u^{R} = \dfrac{q_x^R}{\max\left[10^{-5},\left(\eta^R - z^R\right)\right]}, \quad \upsilon^{R} = \dfrac{q_y^R}{\max\left[10^{-5},\left(\eta^R - z^R\right)\right]} \end{cases}, \\
&\text{c)} \quad \begin{cases} h^{\wedge L} = \max\left[0, \left(\eta^{L} - z^{\wedge LR}\right)\right] \\ h^{\wedge R} = \max\left[0, \left(\eta^{R} - z^{\wedge LR}\right)\right] \end{cases}, \\
&\text{d)} \quad \begin{cases} \eta^{\wedge L} = h^{\wedge L} + z^{\wedge LR} \\ \eta^{\wedge R} = h^{\wedge R} + z^{\wedge LR} \end{cases}, \\
&\text{e)} \quad \begin{cases} q_x^{\wedge L} = u^{L} h^{L} \quad q_y^{\wedge L} = \upsilon^{L} h^{L} \\ q_x^{\wedge R} = u^{R} h^{R} \quad q_y^{\wedge R} = \upsilon^{R} h^{R} \end{cases},
\end{aligned}
\tag{2.26}
$$

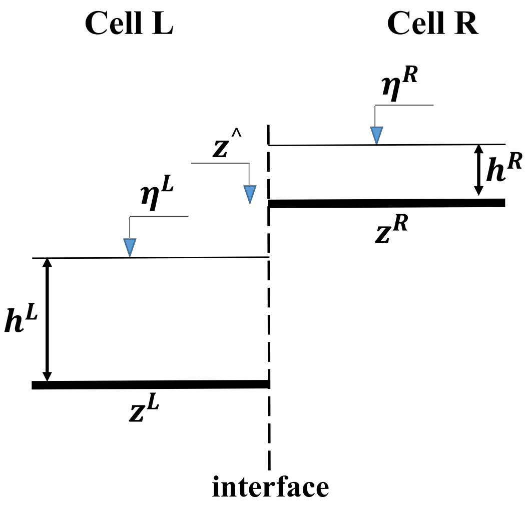


Figure 2.5: Two neighbour cells. This figure defines the variables used in the introduced wetting-drying algorithm.

2.7 Grid convergence assessment

In computational modelling, the ordered discretisation error may be assessed through the examination of the spatial convergence. This assessment is preceded from an error analysis, which is based on the numerical solutions of a model obtained at different (two or more) grid (or mesh) resolutions. Herein, the grid convergence assessment is established via the GCI (grid convergence index) criterion proposed by Roache (1994). The GCI criterion relies on the old concept of Richardson extrapolation (Richardson, 1911).

Using the Richardson extrapolation at a certain point of the domain of study, the discretised solution f can be written as a series via the exact solution (f_e) at that point as:

$$f = f_e + C_1(\Delta x) + C_2(\Delta x)^2 + C_3(\Delta x)^3 + ..., \tag{2.27}$$

where the functions C_1, C_2, etc. are defined in the continuum and are independent of any discretisation, Δx denotes the used grid resolution at which the solution f has been performed. By neglecting the terms of order higher than p, Eq. 2.27 leads to

$$f = f_e + C_p(\Delta x)^p + \text{H.O.T}, \tag{2.28}$$

where, H.O.T. are higher-order terms, and p is the order of accuracy. Order of accuracy is an important property, which defines the pace at which a numerical solution is converged, as the grid is refined, to the exact solution.

Assume that at the point we are discussing, there are three solutions f_1, f_2, and f_3 which have been obtained at the grid size $\Delta x_1, \Delta x_2$, and Δx_3, respectively, so that $\frac{\Delta x_2}{\Delta x_1} = \frac{\Delta x_3}{\Delta x_2} = r$ and Δx_1 is the highest grid resolution adopted. Then Eq. 2.28 gives

$$f_i \cong f_e + C_p(\Delta x_i)^p, i = 1, 2, 3. \tag{2.29}$$

Using Eq. 2.29 and through certain algebraic operations, the order of accuracy can be approximated as:

$$p \cong \frac{\ln\left(\frac{f_3-f_2}{f_2-f_1}\right)}{\ln(r)}, \tag{2.30}$$

and f_e can be estimated by

$$\begin{cases} f_e = f_1 + \frac{(f_1-f_2)}{r^p-1} \\ f_e = f_2 + \frac{(f_2-f_3)}{r^p-1} \end{cases} \tag{2.31}$$

Using Eq. 2.31 and based on the solutions f_1, f_2, and f_3, the estimated fractional error (E_e) is expressed as:

$$\begin{cases} E_{e_1} = \frac{f_e - f_1}{f_1} \times 100 \cong \frac{\theta_{21}}{r^p - 1} \\ E_{e_2} = \frac{f_e - f_2}{f_2} \times 100 \cong \frac{\theta_{32}}{r^p - 1} \end{cases}, \tag{2.32}$$

where, θ is the percentage relative error, which defined as:

$$\theta_{21} = \frac{f_1 - f_2}{f_1} \times 100, \qquad \theta_{32} = \frac{f_2 - f_3}{f_2} \times 100. \tag{2.33}$$

GCI is a criterion, which is conceptually similar to the estimated fractional error, is defined as follows:

$$\mathrm{GCI}_{21} = F_s \frac{\mid \theta_{21} \mid}{r^p - 1}, \qquad \mathrm{GCI}_{32} = F_s \frac{\mid \theta_{32} \mid r^p}{r^p - 1}, \tag{2.34}$$

where, F_s is the factor of safety and it is recommended to be set to 3.0, when the solutions f_1 and f_2 are only available, and $F_s = 1.25$ when the solution f_3 is also incorporated in the process of analysis. The GCI criterion can ensure us that grids are in the asymptotic range of convergence by checking:

$$\frac{\mathrm{GCI}_{32}}{r^p \mathrm{GCI}_{32}} \cong 1. \tag{2.35}$$

2.8 Summary and conclusions

In this chapter, the integral form of the 2D shallow water equations as the governing equations adopted in this research was introduced. Numerical methods in the literature widely used to solve the 2D SWEs were also reviewed, and their advantages and disadvantages discussed. Based on the resulting findings, the finite volume method with an HLLC approximate Riemann solver was chosen for the current research

Inaccurate application of the boundary conditions smears the solution. In this chapter, all possible boundary conditions states (i.e. reflective and transmissive boundary conditions under different flow regimes) for a Cartesian mesh, which is adopted herein, were explained.

The proposed model in this thesis requires to be verified against a set of analytical test cases. Herein, the dam-break problem and inverse method were introduced, according to which a large wide range of artificial tests with exact solutions to the SWE under the steady and unsteady flow regimes are created. To prevent any unrealistic and unphysical solutions in water depth and velocity, the measures were recommended by Liang and Marche (2009) were reviewed and implemented in the current study. In this chapter, the concept of GCI is also reviewed to be utilised in providing a formal grid convergence analysis of test cases presented in Chapter 4.

Chapter 3

Model development

This chapter presents the development of the two nested mesh computational model, a sub-grid approach developed based on the integral form of 2D shallow water equations. The governing equations are solved on the high-resolution mesh and upscaled to the computational (low-resolution) mesh. The solution is performed in the framework of an HLLC approximate Riemann solver. Herein, the numerical treatment of all terms in the governing equations, including the homogenous shallow water equations and friction and bed slope source terms is described in detail. In Chapter 1, different models based on the 2D shallow water equations and their simplified versions were explained. To avoid any ambiguity, the governing equations adopted here, which have small differences compared to the formulation previously displayed (i.e. Eqs. 2.1 to 2.3), are presented in this chapter.

3.1 The nested meshes sub-grid model approach

The governing equations are solved on two nested meshes, coarse and fine, as shown in Fig. 3.1. Cells defined on the coarse and fine meshes are hereafter referred to as large and small cells, respectively. Large cells are the computational cells and contain several small cells where information such as bed elevation and roughness coefficient are defined as constant. Within a large cell, topography and roughness are piecewise constant. In this work, only grids of rectangular cells are considered, although the modelling framework presented here may be adapted to other mesh types. This type of nested computational grid has also been previously used by other sub-grid models (see e.g. Stelling, 2012; Volp et al., 2013; Platzek et al., 2016; Duan et al., 2017; Sanders and Schubert, 2019)

Fig. 3.1 illustrates the nested mesh and the main symbols used to describe it. The edges of the large cell are denoted as Γ_1 to Γ_4 anticlockwise and W_d and L_d denote the overall dimension of the rectangular computational domain. $\Delta X, \Delta Y, \delta x$ and δy are the dimensions of the large and small cells, respectively and z is the elevation of the bed (hereafter often also referred to as micro-topography). J and K are the number of columns and rows of the small cells in a large cell, respectively. Hydraulic and geometric variables defined at large and small cells are labelled as follows. Large cells are denoted by subscript i, whereas the position of a small cell inside this large cell is represented by subscripts j and k (column and row, respectively), (see Fig. 3.1). As an example $z_i\ |_{j,k}$ means bed level at j^{th} column and k^{th} row of small cells inside the i^{th} large cell.

3.2 Governing equations and their time integration

The proposed model solves the integral form of the two-dimensional shallow water equations the derivation of which is available in Appendix A.

$$\frac{\partial}{\partial t}\int_{\Omega} \boldsymbol{U} d\Omega + \oint_{\Gamma} [\boldsymbol{E}(\boldsymbol{U}) \cdot \boldsymbol{e}] d\Gamma = \int_{\Omega} \boldsymbol{S}(\boldsymbol{U}) d\Omega, \tag{3.1}$$

where,

$$\boldsymbol{U} = \begin{pmatrix} \eta \\ q_x \\ q_y \end{pmatrix}, \quad \boldsymbol{E} = [\boldsymbol{F}, \boldsymbol{G}], \quad \boldsymbol{F}(\boldsymbol{U}) = \begin{pmatrix} q_x \\ \frac{q_x^2}{h} + \frac{g}{2}h^2 \\ \frac{q_x q_y}{h} \end{pmatrix}, \quad \boldsymbol{G}(\boldsymbol{U}) = \begin{pmatrix} q_y \\ \frac{q_x q_y}{h} \\ \frac{q_y^2}{h} + \frac{g}{2}h^2 \end{pmatrix}, \tag{3.2}$$

$$\boldsymbol{S}(\boldsymbol{U}) = \boldsymbol{S}_b(\boldsymbol{U}) + \boldsymbol{S}_f(\boldsymbol{U}), \quad \boldsymbol{S}_b(\boldsymbol{U}) = \begin{pmatrix} 0 \\ ghs_{0_x} \\ ghs_{0_y} \end{pmatrix}, \quad \boldsymbol{S}_f(\boldsymbol{U}) = \begin{pmatrix} 0 \\ -ghs_{f_x} \\ -ghs_{f_y} \end{pmatrix}, \tag{3.3}$$

in which Ω is area of the domain over which the equation is solved (i.e. the area of a large cell), Γ represents the boundary of this domain, h denotes water depth, η is water surface elevation, $\boldsymbol{e} = \left[e_x, e_y\right]$ is the unit vector normal to Γ, $s_{f_x}, s_{f_y}, s_{0_x}, s_{0_y}, q_x$ and q_y are the x and y components of frictional and bed slopes, and unit width discharges, respectively. Friction slopes are modelled using Manning's expression $s_{f_x} = \frac{u\|V\|n_M^2}{h^{4/3}}, s_{f_y} = \frac{v\|V\|n_M^2}{h^{4/3}}$, where u and v are x and y velocity components and $\| \boldsymbol{V} \| = \sqrt{u^2 + v^2}$ is the magnitude of the velocity vector. It should be noted that the conserved variable h in

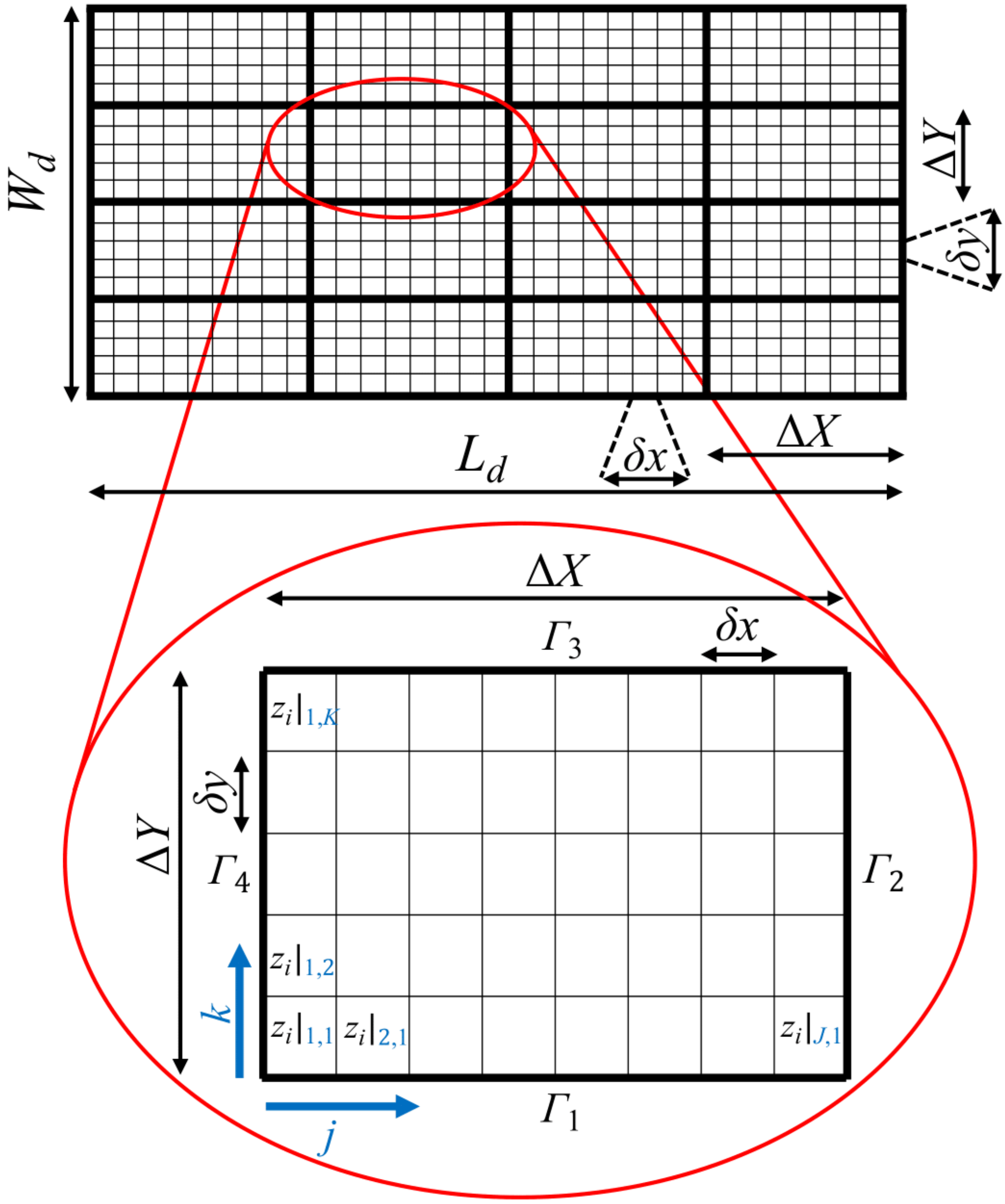


Figure 3.1: Nested meshes adopted in the proposed sub-grid.

the vector $\boldsymbol{U}(x, y, t)$ is replaced with η given that $\eta = h + z$ and a non-erodible bed is assumed (i.e. $\frac{\partial z}{\partial t} = 0$).

Spatial averages over computational cells are hereafter denoted by the overbar, e.g.

$$\overline{\boldsymbol{U}} = \frac{1}{\Omega} \int_{\Omega} \boldsymbol{U} \mathrm{d}\Omega. \tag{3.4}$$

The solution at time level $n + 1$ is obtained from values of the conserved variables at n via an intermediate state usually referred to as the fractional step (LeVeque, 2002). In the proposed model, the intermediate state is obtained by solving the equations including all terms except $\boldsymbol{S}_f$. This time integration is performed over a large cell by the first order Euler method as follows,

$$\overline{\boldsymbol{U}}_i^* = \overline{\boldsymbol{U}}_i^n - \frac{\Delta t}{\Omega_i} \left\{ \sum_{m=1}^{m=4} \left[\oint_{\Gamma_m} \boldsymbol{E}_m(\boldsymbol{U}_i^n) \cdot \boldsymbol{e}_m \mathrm{d}\Gamma_m \right] + \int_{\Omega_i} \boldsymbol{S}_b(\boldsymbol{U}_i^n) \mathrm{d}\Omega_i \right\}, \tag{3.5}$$

where, $\overline{\boldsymbol{U}}_i^*$ is the average of the conservative variable vector at the intermediate state. The solution at time level $n + 1$ is then obtained by introducing the friction term:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_{\Omega_i} \boldsymbol{U}_i^* \mathrm{d}\Omega = \int_{\Omega_i} \boldsymbol{S}_f \left(\boldsymbol{U}_i^{*,n+1} \right) \mathrm{d}\Omega, \tag{3.6}$$

which in differential form reads as

$$\frac{\mathrm{d}}{\mathrm{d}t} \boldsymbol{U}_i^* = \boldsymbol{S}_f \left(\boldsymbol{U}_i^{*,n+1} \right). \tag{3.7}$$

Godunov type finite volume techniques for the solution of Eq. 3.5 estimate the first integral using Riemann solvers with values of $\overline{\boldsymbol{U}}_i^n$ reconstructed on each side of the edges of the cell, while the second integral is typically solved as $\boldsymbol{S}_b \left(\overline{\boldsymbol{U}}_i^n \right) \mathrm{d}\Omega_i$, (e.g. Valiani and Begnudelli, 2006; Liang and Marche, 2009; Liang, 2010). At fine grid resolutions, this approach typically results in good accuracy, but since the objective of the current work is to solve Eqs. 3.5 and 3.6 at coarse resolution, an improved method needs to be adopted. The following sections describe the methods proposed in this research to approximate each of the integrals in Eqs. 3.5 and 3.6.

To prevent numerical instabilities, the time step used in the solution of Eq. 3.5 must be restricted by the Courant–Friedrichs–Lewy (*CFL*) condition (see e.g. Toro, 2001; LeVeque, 2002),

$$\Delta t = \min \left(\frac{\Delta X \; CFL}{\frac{\left| \overline{q}_{x_i}^n \right|}{\overline{h}_i^n} + \sqrt{g \overline{h}_i^n}}, \frac{\Delta Y \; CFL}{\frac{\left| \overline{q}_{y_i}^n \right|}{\overline{h}_i^n} + \sqrt{g \overline{h}_i^n}} \right)_{i=1}^{i=N} \qquad CFL < 1, \qquad N = \text{number of large cells}. \tag{3.8}$$

3.3 Upscaling the shallow water equations

3.3.1 Solving the homogeneous part of the governing equations

The proposed model uses a Godunov-type finite volume method to solve the homogeneous part of equations. The procedure is similar to other finite volume models, except for one key difference. In order to improve the accuracy of fluxes computed at each interface of a large cell, the model solves a Riemann problem at each interface between small cells located at the edges of large cells, whereas in traditional finite volume models, only one solution is performed at each large edge. This approach is similar to the recently published method of Sanders and Schubert (2019). Fig. 3.2 shows an example of two neighbouring large cells. At the common interface, there are four small cells on each side, and the Riemann problem would need to be solved four times. Fluxes at each small cell interface are solved by the Harten, Lax and van Leer Contact (HLLC) Riemann solver (Toro et al., 1994) with values of depths and velocities reconstructed on both sides of each small cells.

Reconstructed depths vary along a cell interface according to micro topography as $h = \max\,(\eta - z, 0)$, while the free surface elevation η is assumed constant across the large cell. It is assumed that this simple reconstruction of η at the edges of a large cell provides an accurate approximation, given that in many problems of flood propagation over complex topography (e.g. river and coastal flooding), changes in the free surface elevation are considerably milder than variations in the bed elevation. Begnudelli et al. (2008) compared the results of two Godunov type FV models: (i) a second order accurate scheme for the solution of the homogeneous equations, (ii) a first order accurate scheme adopting a second order topographic model (linear variation of bed elevation inside each cell). They concluded that in practical cases, the latter can be more accurate, efficient and robust than the former. The piecewise constant reconstruction also simplifies the equations representing the effects of micro topography in the SWE and the corresponding solution. The following section describes a new methodology to model the distribution of the remaining conserved variables (q_x, q_y) at the interface between two large cells, which are also used in the computation of fluxes through a Riemann solver.

3.3.2 Flow reconstruction at the edges of two neighbouring large cells

Eq. 3.5 computes the spatially averaged values of $\overline{\boldsymbol{U}}^*$ from the values of $\overline{\boldsymbol{U}}^n$. In order to approximate the fluxes across large cells accurately, values of $\overline{\boldsymbol{U}}^n$ need to be downscaled to define the values of η and $\boldsymbol{q} = [q_x, q_y]$ at interfaces between the small cells. In this work, we propose an heuristic method to downscale spatially averaged conserved variables $\overline{q}_x^n$ and $\overline{q}_y^n$ to the fine resolution grid at the interface

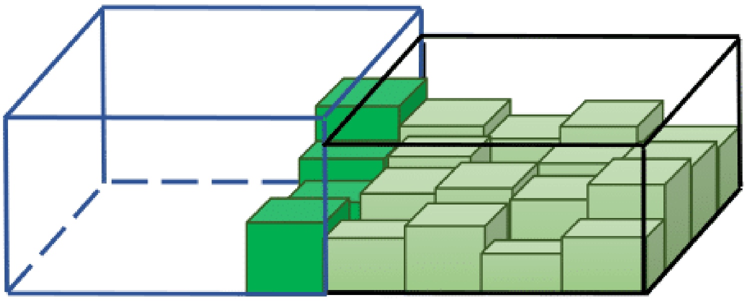


Figure 3.2: Example of the distribution of bed elevation within and along the interface between two large cells.

between two large cells.

Fig. 3.3 shows an example of a cross-section along the edge Γ_4 of i^{th} large cell. The values of depth and Manning's roughness coefficient of the small cells vary along the cross-section. The model distributes the values of $\overline{q}^n_y$ and $\overline{q}^n_y$ along the edge Γ_4 solely for the wet small cells, the number of which $\left(N_{\Gamma_4}\right)$ is a function of η^n_i. This distribution is performed based on the assumption of a constant friction slope at the edges (e.g. Chow, 1959; Cunge et al., 1980; Burguete et al., 2008). Using Manning's relation for the x component of the unit discharge,

$$q^n_{x_i}\Big|_{1,k} = \left[\left(\frac{1}{n_M} h^{\frac{5}{3}}\right)_{1,k} \left(\sqrt{s_f}\varepsilon_x\right)_{\Gamma_4}\right]^n_i, \qquad 1 \le k \le N^n_{\Gamma_4}, \tag{3.9}$$

where, $\varepsilon^n_{x_i}$ is the x component of the unit vector parallel to $\boldsymbol{q}$ at i^{th} large cell and time level n, and $s^n_{f_i}\Big|_{\Gamma_4}$ represents the (constant) magnitude of the friction gradient at all small cells adjacent to the Γ_4. The cross-section average of the x component of unit width discharge at the edge Γ_4 (here denoted by $\tilde{q}^n_{x_i}\Big|_{\Gamma_4}$) is by definition given by

$$\tilde{q}^n_{x_i}\Big|_{\Gamma_4} = \left[\frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left(q_x|_{1,k}\right)\right]^n_i, \tag{3.10}$$

or alternatively,

$$\tilde{q}^n_{x_i}\Big|_{\Gamma_4} = \left[\left(\psi_e \sqrt{s_f}\varepsilon_x\right)_{\Gamma_4}\right]^n_i, \tag{3.11}$$

where, ψ_e is the cross-section's conveyance,

$$\psi^n_{e_i}\Big|_{\Gamma_4} = \left[\frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left(\frac{h^{\frac{5}{3}}}{n_M}\right)_{1,k}\right]^n_i. \tag{3.12}$$

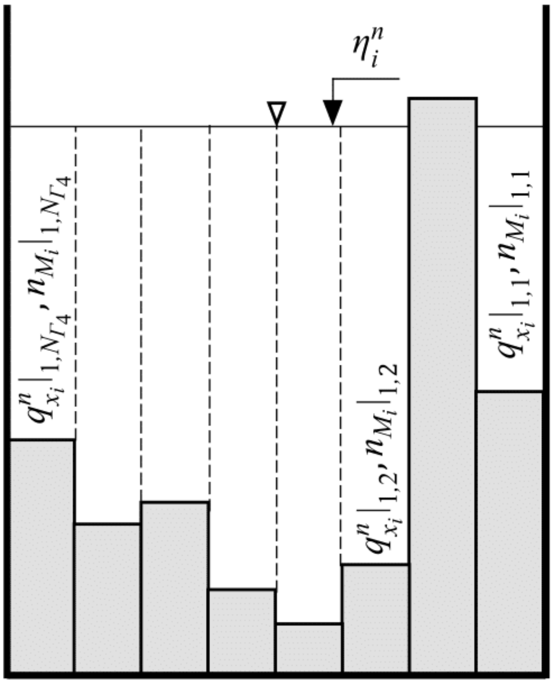

η^n_i
$q^n_{x_i}|_{1,N_{\Gamma_4}}, n_{M_i}|_{1,N_{\Gamma_4}}$
$q^n_{x_i}|_{1,2}, n_{M_i}|_{1,2}$
$q^n_{x_i}|_{1,1}, n_{M_i}|_{1,1}$

Figure 3.3: Unit width discharge distribution across the edge Γ_4 of i^{th} large cell of the domain.

Downscaling the values of q through Eq. 3.10 requires the computation of ψ_e through Eq. 3.12, which can be computationally inefficient at fine grid resolution. Much of this cost can be reduced by computing the values of ψ_e for different values of η_i at pre-processing level and recovering them from tables at run time. Such method requires knowledge of the maximum value of η^n_i attained during simulations, which is unknown *a priori*. Setting a maximum η^n_i that is too high may lead to increased computational cost, because of the need to search over a large list during runtime. We here propose an alternative method, which uses a Taylor series expansion centred about a point $\tilde{h}^n_i|_{\Gamma_4}$ (the average of h at the wet small cells adjacent to the edge Γ_4) to obtain an approximation of ψ_e, as follows. First, the term $\left(h^n_i\right)^{\frac{5}{3}}_{1,k}$ is written as

$$\left(h^n_i\right)^{\frac{5}{3}}_{1,k} = \left[\left(\tilde{h}\right)^{\frac{5}{3}}_{\Gamma_4} + \frac{5}{3}\left(\tilde{h}\right)^{\frac{2}{3}}_{\Gamma_4}(\Delta h)_{1,k} + \frac{5}{9}\left(\tilde{h}\right)^{\frac{-1}{3}}_{\Gamma_4}(\Delta h)^2_{1,k} + \mathcal{O}(\Delta h)^3_{1,k}\right]^n_i, \tag{3.13}$$

where, $\left(\Delta h_i\right)^n_{1,k} = \left(h|_{1,k} - \tilde{h}|_{\Gamma_4}\right)^n_i$ and $\tilde{h}_i|_{\Gamma_4}$ the depth averaged over the wet cells adjacent to Γ_4. Neglecting the terms of order higher than two, ψ_e can be approximated by ψ_a as

$$\psi^n_{a_i}\Big|_{\Gamma_4} = \left\{\frac{1}{N_{\Gamma_4}}\sum_{k=1}^{N_{\Gamma_4}}\frac{1}{n_M|_{1,k}}\left[\left(\tilde{h}\right)^{\frac{5}{3}}_{\Gamma_4} + \frac{5}{3}\left(\tilde{h}\right)^{\frac{2}{3}}_{\Gamma_4}(\Delta h)_{1,k} + \frac{5}{9}\left(\tilde{h}\right)^{\frac{-1}{3}}_{\Gamma_4}(\Delta h)^2_{1,k}\right]\right\}^n_i, \tag{3.14}$$

given that, $\eta^n_i = h_i|^n_{j,k} + z_i|_{j,k}$ is assumed constant, then $\left(\Delta h_i\right)^n_{1,k} = \left(h|_{1,k} - \tilde{h}|_{\Gamma_4}\right)^n_i = \left(\tilde{z}^n|_{\Gamma_4} - z|_{1,k}\right)_i = \left(\Delta z_i\right)^n_{1,k}$ where $\tilde{z}^n_i|_{\Gamma_4}$ is the average of wet small cells bed levels at the cross-section and time level n.

Then,

$$\psi_{a_i}^n\Big|_{\Gamma_4} = \left\{ \frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \frac{1}{n_M|_{1,k}} \left[(\eta - \tilde{z})_{\Gamma_4}^{\frac{5}{3}} + \frac{5}{3} (\eta - \tilde{z})_{\Gamma_4}^{\frac{2}{3}} (\Delta z)_{1,k} + \frac{5}{9} (\eta - \tilde{z})_{\Gamma_4}^{\frac{-1}{3}} (\Delta z)_{1,k}^2 \right] \right\}_i^n, \quad (3.15)$$

$$\psi_{a_i}^n\Big|_{\Gamma_4} = \frac{1}{N_{\Gamma_4}} \left\{ (\eta - \tilde{z})_{\Gamma_4}^{\frac{5}{3}} \sum_{k=1}^{N_{\Gamma_4}} \left(\frac{1}{n_M} \right)_{1,k} + \frac{5}{3} (\eta - \tilde{z})_{\Gamma_4}^{\frac{2}{3}} \sum_{k=1}^{N_{\Gamma_4}} \left(\frac{\Delta z}{n_M} \right)_{1,k} + \frac{5}{9} (\eta - \tilde{z})_{\Gamma_4}^{\frac{-1}{3}} \sum_{k=1}^{N_{\Gamma_4}} \left[\frac{(\Delta z)^2}{n_M} \right]_{1,k} \right\}_i^n, \quad (3.16)$$

or

$$\psi_{a_i}^n\Big|_{\Gamma_4} = \left\{ \left[(\eta - \tilde{z})^{\frac{5}{3}} T_a + \frac{5}{3} (\eta - \tilde{z})^{\frac{2}{3}} T_b + \frac{5}{9} (\eta - \tilde{z})^{\frac{-1}{3}} T_c \right]_{\Gamma_4} \right\}_i^n, \quad (3.17)$$

where, $T_{a_i}^n\Big|_{\Gamma_4}$, $T_{b_i}^n\Big|_{\Gamma_4}$ and $T_{c_i}^n\Big|_{\Gamma_4}$ are defined as

$$T_{a_i}^n\Big|_{\Gamma_4} = \left\{ \frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left(\frac{1}{n_M} \right)_{1,k} \right\}_i^n, \quad (3.18)$$

$$T_{b_i}^n\Big|_{\Gamma_4} = \left\{ \frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left[\frac{\left(\tilde{z}|_{\Gamma_4} - z \right)}{n_M} \right]_{1,k} \right\}_i^n, \quad (3.19)$$

$$T_{c_i}^n\Big|_{\Gamma_4} = \left\{ \frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left[\frac{\left(\tilde{z}|_{\Gamma_4} - z \right)^2}{n_M} \right]_{1,k} \right\}_i^n. \quad (3.20)$$

Eq. 3.11 and Eq. 3.12, and the assumptions that $\tilde{q}_{x_i}^n|_{\Gamma_4} = \overline{q}_{x_i}^n$ and $\psi_{a_i}^n|_{\Gamma_4} \cong \psi_{e_i}^n|_{\Gamma_4}$ lead to the following approximate expression for the friction slope:

$$\left[\left(\sqrt{s_f} \varepsilon_x \right)_{\Gamma_4} \right]_i^n = \left(\frac{\bar{\boldsymbol{q}}_x}{\psi_a|_{\Gamma_4}} \right)_i^n. \quad (3.21)$$

The downscaled value of the unit discharge is then obtained by

$$q_{x_i}^n\Big|_{1,k} = \left[\left(\frac{1}{n_M} h^{\frac{5}{3}} \right)_{1,k} \frac{\bar{\boldsymbol{q}}_x}{\left(\psi_a \right)_{\Gamma_4}} \right]_i^n, \quad 1 \leq k \leq N_{\Gamma_4}^n. \quad (3.22)$$

The same method is used to downscale the y component of the unit discharge.

A few important remarks must be made about Eq. 3.22. First, the parameters T_a to T_c are computed based on the values of bed elevation and roughness of wet small cells only. Therefore, these values are not constant in a simulation because the number of wet cells depends on η_i^n. These parameters are computed as a function of η_i at pre-processing for all elevations of small cells within a large cell, and stored in a sorted vector for efficient runtime search. This reduces the computational cost considerably, because the corresponding values are obtained only once per large cell. A comparison of numerical values of ψ_e and ψ_a was performed for different cross-section shapes, concluding that these two equations generally yield very close results (i.e. typically, less than 1% error. See Appendix C).

It should be noted that in the proposed model, to follow the Godunov type finite volume technique, first the conserved variables are constructed based on the method detailed above. Then, the wetting and drying algorithm described in Section 2.1 is applied to preserve the positivity of water depth at small cells next to the large cell interfaces and finally, the Riemann problem is solved in accordance with the procedure explained in Chapter 2.

3.3.3 Source terms

Bed slope source term

The model approximation of the bed slope source term in Eq. 3.5 is described here for the x component only. Valiani and Begnudelli (2006) showed that for a differential volume of fluid, if the water level is assumed constant across the cell, the bed slope term can be approximated as (the derivation is available in Appendix D)

$$ghs_{0_x} = \frac{\partial}{\partial x}\left(\frac{1}{2}gh^2\right). \tag{3.23}$$

Substituting Eq. 3.23 and the expression $h(x, y) = \eta - z(x, y)$ into the last term of Eq. 3.5 yields:

$$\begin{aligned}
-\tfrac{\Delta t}{\Omega_i}\left(\int_{\Omega_i} ghs_{0_x}\mathrm{d}\Omega\right)_i^n &= -\tfrac{\Delta t}{\Omega_i}\left[\int_{y|_{\Gamma_1}}^{y|_{\Gamma_3}}\int_{x|_{\Gamma_4}}^{x|_{\Gamma_2}}\tfrac{\partial}{\partial x}\left(\tfrac{1}{2}gh^2\right)\mathrm{d}x\mathrm{d}y\right]_i^n \\
&= -\tfrac{\Delta t}{\Omega_i}\left\{\int_{y|_{\Gamma_1}}^{y|_{\Gamma_3}}\left[\int_{x|_{\Gamma_4}}^{x|_{\Gamma_2}}\mathrm{d}\left(\tfrac{1}{2}gh^2\right)\right]\mathrm{d}y\right\}_i^n \\
&= -\tfrac{\Delta t}{\Omega_i}\left[\int_{y|_{\Gamma_1}}^{y|_{\Gamma_3}}\left(\tfrac{1}{2}gh^2|_{\Gamma_2}-\tfrac{1}{2}gh^2|_{\Gamma_4}\right)\mathrm{d}y\right]_i^n \\
&= -\tfrac{g\Delta t}{2\Omega_i}\left\{\int_{y|_{\Gamma_1}}^{y|_{\Gamma_3}}\left[\left(\eta - z|_{\Gamma_2}\right)^2 - \left(\eta - z|_{\Gamma_4}\right)^2\right]\mathrm{d}y\right\}_i^n.
\end{aligned} \tag{3.24}$$

Central to the accuracy of any sub-grid formulation is the ability to capture the condition of a large computational cell that is only partially submerged (i.e. partially wet cells). Since the governing equations are only defined at wet regions of the domain, such a condition can be modelled by considering only the wet sub-region of the cell. This is achieved in the model by substituting the variable $\acute{z} = \min(z, \eta)$ for z in Eq. 3.24. Fig. 3.4 illustrates how this change of variable translates into integration over the wet sub-domain of a large cell only. In Fig. 3.4a, the cell is fully wet and $\acute{z} = z$. On the other hand, the left edge of the cell shown in Fig. 3.4b is unsubmerged, and the change to $\acute{z} = \eta$ effectively moves the boundary of integration to the wet/dry front.

Using $\acute{z}$ in Eq. 3.24, expanding the quadratic terms and substituting $K = \frac{\Delta Y}{\delta y}$ yields:

$$\begin{aligned}
-\tfrac{\Delta t}{\Omega_i}\left(\textstyle\int_{\Omega_i} g h s_{0_x} \mathrm{d}\Omega\right)_i^n &= -\tfrac{g\Delta t}{2\Omega_i}\left\{\textstyle\int_{y|_{\Gamma_1}}^{y|_{\Gamma_3}} \left[\left(\eta - \acute{z}|_{\Gamma_2}\right)^2 - \left(\eta - \acute{z}|_{\Gamma_4}\right)^2\right] \mathrm{d}y\right\}_i^n \\
&= -\tfrac{g\Delta t}{2\Omega_i}\left\{\textstyle\int_{y|_{\Gamma_1}}^{y|_{\Gamma_3}} \left[\left(\acute{z}|_{\Gamma_4} - \acute{z}|_{\Gamma_2}\right)\left(2\eta - \acute{z}|_{\Gamma_4} - \acute{z}|_{\Gamma_2}\right)\right] \mathrm{d}y\right\}_i^n \qquad (3.25) \\
&= \tfrac{-g\Delta t}{2\Delta X K} \textstyle\sum_{k=1}^{K} \left[2\Delta \acute{z}_x|_k \left(\eta - \hat{\acute{z}}|_k\right)\right]_i^n ,
\end{aligned}$$

where $\Delta \acute{z}_{x_i}|_k^n = \left(\acute{z}|_{1,k} - \acute{z}|_{N_x,k}\right)_i^n$ and $\hat{\acute{z}}_i|_k^n = \frac{1}{2}\left(\acute{z}|_{1,k} + \acute{z}|_{N_x,k}\right)_i^n$.

Eq. 3.25 shows that the bed slope source term is a function of water surface (η), cell sizes and topographic data at the edges of a large cell only. The topographic data of other internal small cells do not contribute to the bed slope computations. This is a convenient property of the equation; as time-consuming computations at fine resolution can be avoided. An identical procedure is used by the model to compute the bed slope source term in the y direction.

Friction source term

Eq. 3.7 is discretised using a widely tested semi-implicit scheme (e.g. Liang and Marche, 2009; Kesserwani and Liang, 2012b; Cea and Bladé, 2015; de Almeida et al., 2018). The scheme is adapted here to include the upscaling of the non-uniform distribution of variables at sub-grid scale, as follows. Substituting Manning's equation into Eq. 3.7, and ignoring the first (zero-valued) component of the friction source term vector results in:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\boldsymbol{q}^*_{j,k}\right)_i = \left\{-g\left[\frac{n_M^2 \boldsymbol{q}^{n+1} \parallel \boldsymbol{q}^* \parallel}{(h^*)^{\frac{7}{3}}}\right]_{j,k}\right\}_i , \qquad (3.26)$$

where j, k represent the location of a small cell inside the i^{th} large cell (see Fig. 3.1) and $\parallel \boldsymbol{q}^*_{j,k} \parallel$ is the magnitude of $\boldsymbol{q}^*_{j,k}$. It is assumed here that while $\parallel \boldsymbol{q}^*_{j,k} \parallel$ varies over a large cell, the flow direction is constant. Therefore, $\boldsymbol{q}_{j,k}(x,y)|_i = \left[\varphi_{j,k}(x,y)\,\overline{\boldsymbol{q}}\right]_i$, where, $\varphi_i(x,y)$ is the coefficient of unit width

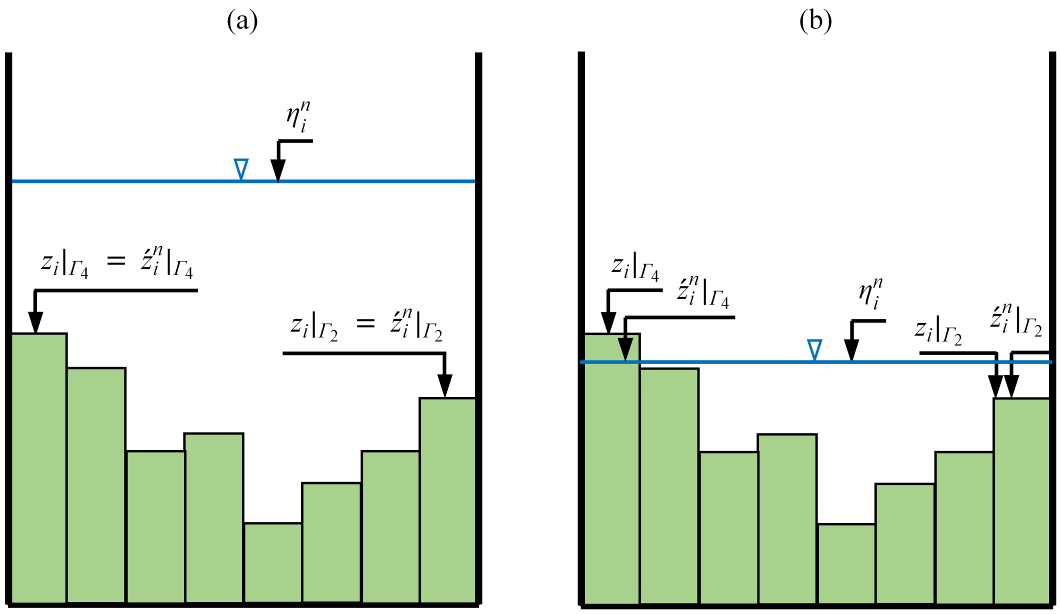

(a)
η_i^n
$z_i|_{\Gamma_4} = \acute{z}_i^n|_{\Gamma_4}$
$z_i|_{\Gamma_2} = \acute{z}_i^n|_{\Gamma_2}$
(b)
$z_i|_{\Gamma_4}$
$\acute{z}_i^n|_{\Gamma_4}$
η_i^n
$z_i|_{\Gamma_2}$
$\acute{z}_i^n|_{\Gamma_2}$

Figure 3.4: Cross section of an arbitrary large cell illustrating the definition of $\acute{z}$ used by the model to compute bed slope source term. (a) In fully wet cells, $\acute{z}$ is simply equal to z but (b) in partially wet cells, $\acute{z}$ in unsubmerged small cells is lower than z (i.e. $\acute{z} = \eta$).

discharge distribution inside the large cell. Substituting this relation into Eq. 3.26 yields:

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\varphi^*|_{j,k}\bar{\boldsymbol{q}}^*\right)_i = \left\{-g\bar{\boldsymbol{q}}^{n+1} \parallel \bar{\boldsymbol{q}}^* \parallel \left[\frac{\varphi^*\varphi^{n+1}n_M^2}{(h^*)^{\frac{7}{3}}}\right]_{j,k}\right\}_i . \tag{3.27}$$

Further, the simplifying assumption that $\frac{\mathrm{d}}{\mathrm{d}t}\left(\varphi^*|_{j,k}\right)_i$ is negligible is introduced here. This assumes that rate of change of the temporal distribution of flow is small compared to the other terms in the equation. Under this assumption, $\left[\bar{\boldsymbol{q}}^* \frac{\mathrm{d}}{\mathrm{d}t}(\varphi^*)_{j,k}\right]_i \ll \left[(\varphi^*)_{j,k}\frac{\mathrm{d}\bar{\boldsymbol{q}}^*}{\mathrm{d}t}\right]_i$ and $\left(\varphi^*|_{j,k}\right)_i \approx \left(\varphi^{n+1}|_{j,k}\right)_i$, which yields:

$$\left[\frac{(h^*)^{\frac{7}{3}}}{n_M^2}\bigg|_{j,k}\right]_i \frac{\mathrm{d}}{\mathrm{d}t}\left(\bar{\boldsymbol{q}}_i^*\right) = \left[-g\bar{\boldsymbol{q}}^{n+1} \parallel \bar{\boldsymbol{q}}^* \parallel \left(\varphi^{n+1}\right)_{j,k}\right]_i, \tag{3.28}$$

Integrating Eq. 3.28 over the wet fraction of the large cell:

$$\int_{\Omega_{w_i}} \left[\frac{(h^*)^{\frac{7}{3}}}{n_M^2}\right]_{j,k} \frac{\mathrm{d}}{\mathrm{d}t}\left(\bar{\boldsymbol{q}}^*\right) \mathrm{d}\Omega_w = \int_{\Omega_{w_i}} -g\bar{\boldsymbol{q}}^{n+1} \parallel \bar{\boldsymbol{q}}^* \parallel \left(\varphi^{n+1}\right)_{j,k} \mathrm{d}\Omega_w, \tag{3.29}$$

where, Ω_w is the wet area of large cell. Since $\bar{\boldsymbol{q}}^*$ and $\bar{\boldsymbol{q}}^{n+1}$ are by definition constant within a large cell,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\bar{\boldsymbol{q}}_i^*\right) \int_{\Omega_{w_i}} \left[\frac{(h^*)^{\frac{7}{3}}}{n_M^2}\right]_{j,k} \mathrm{d}\Omega_w = -g\bar{\boldsymbol{q}}^{n+1} \parallel \bar{\boldsymbol{q}}^* \parallel \int_{\Omega_{w_i}} \left(\varphi^{n+1}\right)_{j,k} \mathrm{d}\Omega_w. \tag{3.30}$$

Given that $\overline{\boldsymbol{q}}_i^{n+1} = \frac{\int_{\Omega_{w_i}} \boldsymbol{q}_{j,k}^{n+1} \mathrm{d}\Omega_w}{\Omega_{w_i}} = \frac{\int_{\Omega_{w_i}} (\varphi^{n+1})_{j,k} \overline{\boldsymbol{q}}^{n+1} \mathrm{d}\Omega_w}{\Omega_{w_i}}$, it follows that $\int_{\Omega_{w_i}} \left(\varphi^{n+1}\right)_{j,k} \mathrm{d}\Omega_w = \Omega_{w_i}$, and

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\overline{\boldsymbol{q}}_i^*\right) = \left[\frac{-g\overline{\boldsymbol{q}}^{n+1} \parallel \overline{\boldsymbol{q}}^* \parallel}{\overline{\left(\frac{h^{\frac{7}{3}}}{n_M^2}\right)}^*}\right]_i, \tag{3.31}$$

where

$$\overline{\left(\frac{h^{\frac{7}{3}}}{n_M^2}\right)}^*_i = \frac{1}{\Omega_{w_i}} \int_{\Omega_{w_i}} \left[\frac{(h^*)^{\frac{7}{3}}}{n_M^2}\right]_{j,k} \mathrm{d}\Omega_w = \left[\frac{1}{N_w}\sum_{j=1}^{J}\sum_{k=1}^{K}\left(\frac{h^{\frac{7}{3}}}{n_M^2}\right)_{j,k}\right]_i^*, \tag{3.32}$$

in which, N_w is the number of submerged small cells in the large cell.

Discretising the derivative in Eq. 3.31 using first-order forward-time finite-differences yields:

$$\overline{\boldsymbol{q}}_i^{n+1} = \left[\frac{\overline{\boldsymbol{q}}}{1 + \frac{g\|\overline{\boldsymbol{q}}\|\Delta t}{\overline{\left(\frac{h^{\frac{7}{3}}}{n_M^2}\right)}}}\right]_i^*. \tag{3.33}$$

Eq. 3.32 requires expensive fine-resolution scale computations to be performed at each time step. In order to limit this computational burden, an approximation to Eq. 3.32 is derived here which follows an approach similar to that described in Section 3.3.2 for the reconstruction of $\boldsymbol{q}$ at the edges of large cells. First, $\left(h_i^*\right)^{\frac{7}{3}}$ is approximated using Taylor series centred about $\overline{h}_i^*$ (the mean water depth over the submerged area of the large cell):

$$\left(h_i^*\right)^{\frac{7}{3}} = \left[\left(\overline{h}\right)^{\frac{7}{3}} + \frac{7}{3}\left(\overline{h}\right)^{\frac{4}{3}}\left(h - \overline{h}\right) + \frac{14}{9}\left(\overline{h}\right)^{\frac{1}{3}}\left(h - \overline{h}\right)^2 + \mathcal{O}\left(h - \overline{h}\right)^3\right]_i^*. \tag{3.34}$$

Neglecting terms of order higher than two and substituting Eq. 3.34 into Eq. 3.32 yields:

$$\overline{\left(\frac{h^{\frac{7}{3}}}{n_M^2}\right)}^*_i = \left\{\frac{1}{N_w}\sum_{j=1}^{J}\sum_{k=1}^{K}\left[\frac{\left(\overline{h}\right)^{\frac{7}{3}} + \frac{7}{3}\left(\overline{h}\right)^{\frac{4}{3}}\left(h|_{j,k} - \overline{h}\right) + \frac{14}{9}\left(\overline{h}\right)^{\frac{1}{3}}\left(h|_{j,k} - \overline{h}\right)^2}{n_M^2|_{j,k}}\right]\right\}_i^*. \tag{3.35}$$

Substituting $h_i^*|_{j,k} = \left(\eta^* - z|_{j,k}\right)_i$, and $\overline{h}_i^* = \eta_i^* - \overline{z}_i^*$ (where $\overline{z}_i^*$ is the mean bed level of wet area of the large cell) into Eq. 3.35 leads to

$$\overline{\left(\frac{h^{\frac{7}{3}}}{n_M^2}\right)}^*_i = \left\{\frac{1}{N_w}\sum_{j=1}^{J}\sum_{k=1}^{K}\left[\frac{\left(\eta - \overline{z}\right)^{\frac{7}{3}} + \frac{7}{3}\left(\eta - \overline{z}\right)^{\frac{4}{3}}\left(\overline{z} - z|_{j,k}\right) + \frac{14}{9}\left(\eta - \overline{z}\right)^{\frac{1}{3}}\left(\overline{z} - z|_{j,k}\right)^2}{n_M^2|_{j,k}}\right]\right\}_i^*, \tag{3.36}$$

or alternatively:

$$\overline{\left(\frac{h^{\frac{7}{3}}}{n_M^2}\right)}_i^* = \left[\left(\eta - \overline{z}\right)^{\frac{7}{3}} T_d + \frac{7}{3}\left(\eta - \overline{z}\right)^{\frac{4}{3}} T_e + \frac{14}{9}\left(\eta - \overline{z}\right)^{\frac{1}{3}} T_f\right]_i^*, \tag{3.37}$$

where

$$T_{d_i}^* = \left\{\frac{1}{N_w}\sum_{j=1}^{J}\sum_{k=1}^{K}\left[\frac{1}{n_M^2}\right]_{j,k}\right\}_i^*, \tag{3.38}$$

$$T_{e_i}^* = \left\{\frac{1}{N_w}\sum_{j=1}^{J}\sum_{k=1}^{K}\left[\frac{\left(\overline{z} - z\right)}{n_M^2}\right]_{j,k}\right\}_i^*, \tag{3.39}$$

$$T_{f_i}^* = \left\{\frac{1}{N_w}\sum_{j=1}^{J}\sum_{k=1}^{K}\left[\frac{\left(\overline{z} - z\right)^2}{n_M^2}\right]_{j,k}\right\}_i^*. \tag{3.40}$$

The parameters T_d, T_e and T_f are computed at pre-processing (for the range of values of z within the submerged part of the large cell) and stored in sorted tables from which they are obtained during runtime as a function of η. The final relation for friction updating is obtained by substituting Eq. 3.37 into Eq. 3.35, which yields:

$$\overline{\boldsymbol{q}}_i^{n+1} = \frac{\overline{\boldsymbol{q}}_i^*}{\sigma_i^*}, \qquad \sigma_i^* = \left[1 + \frac{g \parallel \overline{\boldsymbol{q}} \parallel \Delta t}{\left(\eta - \overline{z}\right)^{\frac{7}{3}} T_d + \frac{7}{3}\left(\eta - \overline{z}\right)^{\frac{4}{3}} T_e + \frac{14}{9}\left(\eta - \overline{z}\right)^{\frac{1}{3}} T_f}\right]_i^*. \tag{3.41}$$

3.4 Model structure

The upscaling of the equations proposed in the previous sections results in a number of parameters that need to be assessed at fine resolution. However, since these parameters (namely $\overline{z}, \tilde{z}, T_a, T_b, T_c, T_d, T_e$ and T_f) depend only on the fine resolution topography and roughness, they can be determined at pre-processing level. This saves substantial runtime computational effort and is central to obtaining high computational performance. Herein, a more detailed explanation of the pre-processing and the overall model structure are provided.

Since the proposed sub-grid model allows the large cells to be partially wet, the number of submerged small cells N_w involved in the estimation of the above parameters changes dynamically during the simulation (i.e. since N_w depends on η when the large cell is only partially wet). During pre-processing, the values of the aforementioned parameters are computed for different values of η based on the elevation and roughness of the corresponding submerged small cells only, and the results are stored in tables sorted by η. During runtime, the values of the parameters are simply retrieved from the table using the value of η at each time step. Since only small changes in η occur from one time-step to another, a relatively expensive search is avoided by storing the index of the table row found in the previous time level, which is used as the initial guess in the search. The computational effort associated with the model's pre-processing is in general negligible compared with the simulation time, since it only needs to be performed once. For example, in all simulations presented the next chapter, pre-processing took no longer than 0.1% of the simulation time. Fig. 3.5 represents the structure of the model, illustrating how the pre-processing links to other components of the model. The algorithm was implemented in C++ programming language and the executable used in this research was compiled with Intel icc compiler version 14.0.2.

3.5 How does the proposed sub-grid model compare with other similar models recently proposed?

In recent years, over the course of this PhD, a few works have been published that that bear similarity to the proposed model. A brief description of these models was provided in Chapter 2. Among them, the methods proposed by Hénonin et al. (2015) and Sanders and Schubert (2019) are the most closely related to the research presented in this thesis. This section describes how the model proposed in this PhD differs from these two models, and why the approach adopted here represents an advance compared to them.

Hénonin et al. (2015) presents a finite difference model for overland flow problems, which, like the approach taken in this thesis, used a nested computational mesh. In Chapter 1, the model of Hénonin et al. (2015) was discussed. In the model proposed by Hénonin et al. (2015), the coarse mesh represents the computational cells, which can be partially wet. Small cells nested in the computational cells are used to incorporate high resolution topography data within the model solution procedure. Water surface elevation (η) in the computational cell is assumed constant whereas water depth (h) varies as $h = \eta - z$. While this modelling conceptualisation is similar to the one adopted in the present thesis, the approach proposed here has a number of characteristics that make it unique and provide a more accurate representation of sub-grid scale processes. Namely, Hénonin et al. (2015) uses high-resolution topography data in a rather simplistic way through the averaging of the water depth in computational cells. The averaging approach lacks a strong theoretical basis and has important limitations, as will be made explicit in the following section. In the model proposed in this PhD thesis the inclusion of fine resolution topography and roughness is performed through integration of the

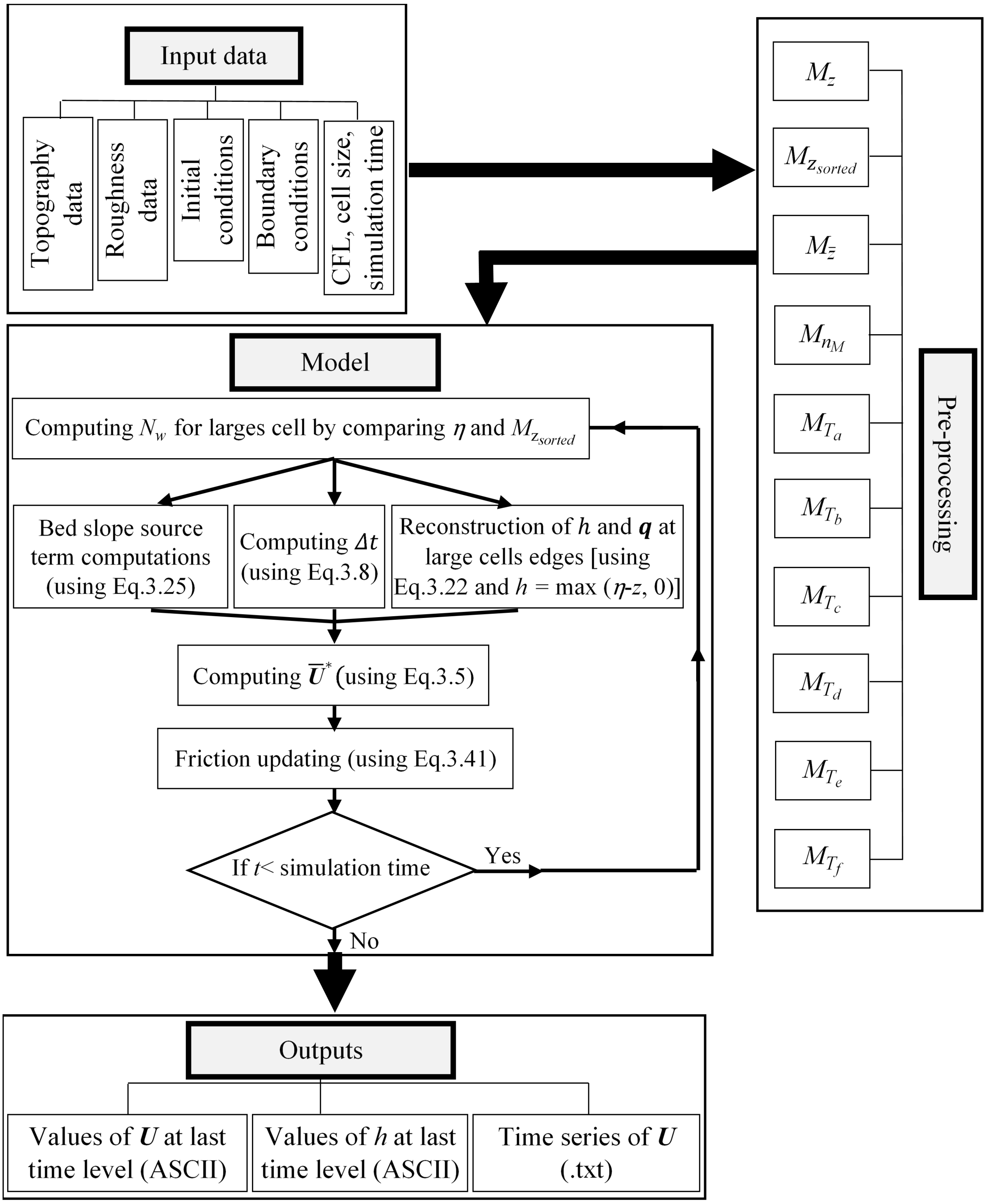

Input data
Topography data
Roughness data
Initial conditions
Boundary conditions
CFL, cell size, simulation time
M_z
$M_{z_{sorted}}$
$M_{\bar{z}}$
M_{n_M}
M_{T_a}
M_{T_b}
M_{T_c}
M_{T_d}
M_{T_e}
M_{T_f}
Pre-processing
Model
Computing N_w for larges cell by comparing η and $M_{z_{sorted}}$
Bed slope source term computations (using Eq.3.25)
Computing Δt (using Eq.3.8)
Reconstruction of h and $\boldsymbol{q}$ at large cells edges [using Eq.3.22 and h = max (η-z, 0)]
Computing $\overline{\boldsymbol{U}}^*$ (using Eq.3.5)
Friction updating (using Eq.3.41)
If t< simulation time
Yes
No
Outputs
Values of $\boldsymbol{U}$ at last time level (ASCII)
Values of h at last time level (ASCII)
Time series of $\boldsymbol{U}$ (.txt)

Figure 3.5: General structure of the proposed model. M_z stores the topographic data of small cells within each large cell. $M_{z_{stored}}$ stores the same topographic information sorted by bed elevation. $M_{\bar{z}}$ stores the values of $\bar{z}$ corresponding to the number of submerged small cells. M_{n_M} stores the fine resolution roughness data within each large cell. M_{T_a} to M_{T_f} store the values of T_a to T_f for large cells.

governing equations, combined with a set of simplifying assumptions that are made explicit during the derivation. These high-resolution roughness data are used in the computation of bed and friction slope source terms and reconstruction of conservative variables at the computational edges for solving the Riemann problem.

Also similar to the work in this thesis, as previously detailed in Chapter 1, Sanders and Schubert (2019) makes use of two Cartesian nested meshes. The model is based on the integral form of shallow water equations, which are solved using the finite volume method over the computational cells (large cells). The small cells provide topography and roughness data. The model also uses high-resolution data to estimate the source terms and fluxes between neighbouring computational cells. Sanders and Schubert (2019) modelling strategy, which has been developed independently from the model proposed in the present thesis, is the same as the one described in the previous section. However, there are important differences between the two models, which are related to the reconstruction of the conservative variables for the solution to the Riemann problem and the approach used to include fine resolution data in the computation of friction within large cells, which are described next.

Reconstruction of the conservative variables at the edges of computational cells

In a Godunov type model, the conservative variables are reconstructed at the cell edges and then used in the solution of the Riemann problem. Typically, this reconstruction is uniform along an edge (i.e. variables on both sides of the inter-cell boundary are constant). In the nested mesh models proposed in this thesis, as well in Sanders and Schubert (2019), a non-uniform reconstruction of conservative variables is used to account for the variation of variables along the cell edges. The reconstruction of h is identical in both models (η is assumed constant, thus $h = \eta - z$). However, Sanders and Schubert (2019) reconstruct $\boldsymbol{q} = [q_x, q_y]$ uniformly on the both sides of edges of the large cell. Given that $\boldsymbol{V} = \boldsymbol{q}/h$, this reconstruction implies an unphysical distribution of flow velocities with values that are high at shallow parts of the edge (indeed, as $\eta \rightarrow z, \boldsymbol{V} \rightarrow \infty$) and low at deep areas. This is contrary to what is commonly observed for shallow water flows (e.g. floodplains or submerged fluvial islands typically exhibit velocities that are lower than the velocities observed in the main channel). In the proposed model, velocity is distributed along the edges of adjacent cells based on the simplifying assumption of a constant energy slope (see Section 3.3.2). The values of $\boldsymbol{q}$ reconstructed in this way display a distribution that more closely resembles flows in open channels.

Friction source term

In Sanders and Schubert (2019), the upscaling of the friction source terms derived are based on the assumption that friction slope ($\boldsymbol{s}_f$) and water surface level (η) are constant over a computational cell [it should be noted that the model of Sanders and Schubert (2019) is second order accurate (piecewise linear reconstruction) for η and first order (reconstructed piecewise constant) for q_x and q_y, but in the source term computations (bed and friction slope), η is also reconstructed piecewise constant]. While the assumption of a constant $\boldsymbol{s}_f$ to approximate the distribution of flow at given cross-sections has been widely used (e.g. Hec-Ras), extending this idea to assume that $\boldsymbol{s}_f$ is constant over an area of flow is questionable and would require further examination. For instance, let us assume a 1D prismatic rectangular channel over an irregular bed topography and under steady state flow. Let us also assume, for simplicity, a constant value of Manning's coefficient (n_M) along the channel. Fig. 3.6 shows a sketch of the bed and free surface profiles in this hypothetical channel. Since Manning's equation is used to estimate the friction slope, $\left(\boldsymbol{s}_f\right)^{\frac{1}{2}} = \frac{q n_M}{h^{\frac{5}{3}}}$, and given that both $\boldsymbol{q}$ and $\boldsymbol{s}_f$ are assumed constant ($\boldsymbol{q}$ =const since the flow is steady) then it follows that h should be constant along the channel, which contradicts the very assumption of a horizontal water surface over a varied topography. In general, the assumption of a constant value of $\boldsymbol{s}_f$ over a large computational domain/cell is likely to be inaccurate because the friction slope may change considerably due to the acceleration of the flow induced by the fine resolution topography. Since the objective of upscaling is to increase the model's accuracy by accounting for sub-grid scale variations in the flow, in this research, $\boldsymbol{s}_f$ is not assumed uniform within a computational cell.

Another consideration that highlights an important difference between the model proposed in this thesis and Sanders and Schubert (2019) relates to how the integral equations are upscaled. To compute the friction source term, the following integral needs to be approximated

$$\frac{1}{\Omega_i}\left(\int_{\Omega_i} g h \boldsymbol{s}_f \mathrm{d}\Omega\right)_i^{*,n+1} = \left(\overline{g h \boldsymbol{s}_f}\right)_i^{*,n+1}, \tag{3.42}$$

Sanders and Schubert (2019) approximate the term $\overline{g h \boldsymbol{s}_f}$ as $\overline{g h \boldsymbol{s}_f} = g\bar{h}\bar{\boldsymbol{s}}_f$, which further reduces the accuracy of the approximation. In the present work, the term $\overline{g h \boldsymbol{s}_f}$ is computed instead.

3.6 Summary and conclusions

In this chapter, the proposed sub-grid model was described in detail. The model makes use of two nested meshes. The coarse mesh is the computational mesh and the fine mesh provides high-resolution

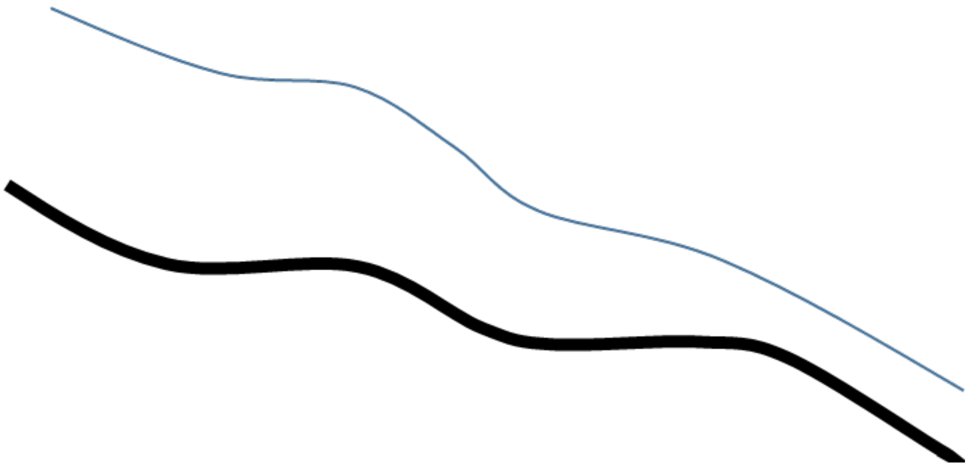


Figure 3.6: Longitudinal bed and surface profiles for an arbitrary 1D prismatic rectangular channel with constant roughness and flow rate.

topography and roughness data that are incorporated into the solution procedure. The governing equations are the 2D shallow water equations. The equations are in integral form and in the vector of conservative variables, with η used instead of h. The governing equations are solved in the framework of Godunov type finite volume method, and the HLLC Riemann solver is used to estimate fluxes. The η reconstruction is piecewise constant at low-resolution mesh, while $\boldsymbol{q}$ is distributed at the edges using a new method, which is a function of the local (i.e. fine scale) Manning coefficient, water depth, and edge conveyance.

As it would be expected, upscaling of the bed slope source term depends on water surface and bed elevation at the large cell edges. Updating the solution from time level n to $n + 1$ is achieved via an intermediate time step, where the solution first proceeds for all terms of the governing equations except friction source term whose effect is then incorporated into the solution. Friction slope computations are executed by the present model. Other models similar to the proposed sub-grid model typically assume that friction slope is constant (Sanders and Schubert, 2019; Volp et al., 2013), which can be acceptable for a cross-section of flow, but it might be invalid for flow in space. Herein, it is only assumed that streamlines are parallel within a large cell. This relaxes the need for the assumption of constant friction slope. To avoid numerical instability, the time step must be restricted. In this research, the time step is determined based on the CFL condition. The formulation is applicable to both fully and partially wet cells.

Chapter 4

Results and discussion

In this chapter, six artificial and one real-world test cases are used to assess the performance of the model in terms of accuracy and speedup. The first test corresponds to 1D steady flow over a hump to assess the performance of the model in transcritical problems. In this test case, the ability of the proposed sub-grid model is tested for the case of a hydraulic jump occurring on the downstream face of the hump. In the second test case, the bore wave propagation over dry bed and still water is studied through the modelling of the well-known 1D dam-break problem. In this case, the numerical results are evaluated against the analytical solutions introduced in Chapter 2. Test Case 3 is aimed at analysing how the proposed sub-grid model deals with the wetting and drying conditions. In this test, a dam-break problem is studied over an initially dry bed with three humps, one of which has a sufficiently high peak elevation to be unsubmerged during the simulation. This test represents a 2D unsteady flood inundation problem. Test Case 4 is an artificial problem where the high-resolution topography includes a rectangular compound channel (i.e. main channel and floodplains) that has a 90° bend within a large cell. The aim of this test is to assess the ability of the sub-grid model to capture changes in the flow direction correctly. Test Case 5 involves simulation of 1D steady flow over relatively complex topography. The test evaluates the accuracy of the model predictions for a scenario involving large variations in bed elevation. The analytical solution for this test is obtained by the inverse method reviewed in Chapter 2 (MacDonald, 1996). Test Case 6 simulates flow in a compound channel. In this test, the ability of the method proposed to upscale the source terms is tested for scenarios involving combined variations in topography and roughness within a computational cell. Test Case 7 assesses the model's performance when simulating a real-world flood event. In this test, the numerical results are assessed against a benchmark solution (obtained from a fine resolution simulation) and field data. This test aims to investigate both the computational efficiency and accuracy of the proposed sub-grid model for realistic flood inundation problems.

4.1 Test cases

In this section, the accuracy and computational performance of the sub-grid model presented in the previous chapter are tested through a set of stringent idealised and real-world test cases. This is done by comparing the results of the sub-grid model (herein denoted by SG) against the model hereafter referred to as traditional (T) model. The T model is simply a particular case of the SG model when the resolution of large and small cells are the same, in which case the formulation reverts to a first order Godunov type finite volume method. Both models were run at different resolutions so that the influence of grid size on the improved performance of the SG can be assessed. The results of simulations are labelled by the type of model followed by the grid resolution used. For example, T (50) refers to a simulation performed with the Traditional model and a grid resolution of 50 m, and SG (20/5) denotes the sub-grid model and grid resolutions of 20 m and 5 m for the coarse and fine meshes, respectively.

4.1.1 Test Case 1, One-dimensional steady flow over a hump

This test adopts a 1D steady transcritical channelized flow to examine the performance of the SG model for free surface flow over a bed that is smoothly varying in space. The domain of study contains a 25 m long rectangular channel whose bed elevation is expressed by

$$z(x) = \begin{cases} 0.2 - 0.05(x-10)^2 & \text{if} \quad 8 < x < 12 \\ 0 & \qquad otherwise \end{cases} \tag{4.1}$$

where the x direction is defined along the channel. This topography creates a 4 m long and 0.2 m height hump in the middle of the channel as shown in Figs. 4.1 to 4.5. The channel bed is frictionless. A constant unit width discharge $q_x = 0.18\text{m}^2/\text{s}$ and a fixed water surface level $\eta = 0.33$ m are imposed to the model as the upstream and downstream boundary conditions, respectively. Initially the water surface $\eta = 0.33$ m and the unit width discharge $q_x = 0.18\text{m}^2/\text{s}$ are assumed throughout the domain of study. Under this circumstance, the analytical solution, which was proposed by Goutal and Maurel (1997), includes a hydraulic jump (i.e. transcritical flow) and subcritical flow upstream and downstream of the hump as shown in Figs. 4.1 to 4.5.

Simulations are performed using the SG model at computational mesh resolution 0.0625 m, 0.125 m, 0.25 m, 0.5 m, and 1 m, while in all the fine resolution mesh (small cells) is 0.0625 m. Fig. 4.1 displays the longitudinal bed surface level together with the predicted and analytical water surface profile along the centre line of the channel in the $x-$ direction. Overall, the results display a relatively good agreement with the analytical solution in all computational resolutions, particularly in the downstream part of the channel. In the upstream part of the domain a small difference (of the order of a few milimiters) is observed between the results of the SG model and the analytical solution as the computational mesh is coarsened. As the test case models a 1D steady flow, then the unit width discharge (q_x) must be constant across the domain. Fig. 4.2 illustrates the predicted q_x for different resolutions against the analytical solution. Around the hump, where there is transcritical flow and hydraulic jump, oscillations are observed in the numerical results, the amplitude of which is higher at low-resolution simulations. These oscillations appeared in results of many researchers who studied this test case, such as Wang (2011), Kalita and Sarma (2018), Delis et al. (2018). For such a problem involving numerical instability and non-convergence solution of discharge at the presence of complex geometry and hydraulic jump, Ying and Wang (2008) proposed the analysis of the results obtained from the flux vector (i.e. obtained at the interface between two computational cells), instead of the value of the conserved variables within the computational cells. Using the flux vector quantities, no oscillations in the solution of the SG model for q_x were observed (see Fig. 4.3). In Fig. 4.4, the velocity results obtained by the SG model at different resolutions are compared to the analytical solution. The numerical results are consistent with the analytical solution, except for a small disturbance at the location of the hydraulic jump. The corresponding predicted profiles are available for Froude number in Fig. 4.5. These profiles, which closely agree with the analytical solution, vary from about 0.23 (upstream) to about 2.78 over the bump.

This test case was also modelled by the T model. At the same computational resolutions, the solution of the T and SG models are very close to each other (the differences are at the scale of 1-2 mm). Then for clarity only the SG's results are presented.

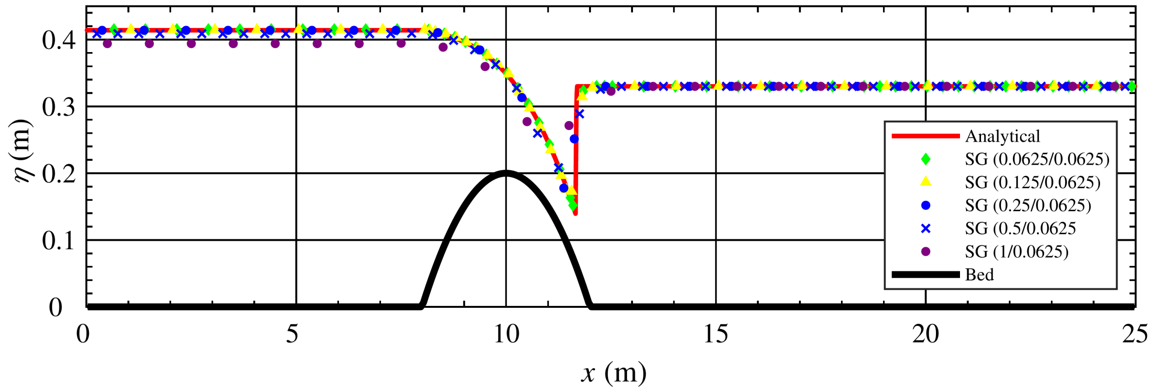


Figure 4.1: Test case 1, steady transcritical flow over a hump: longitudinal bed water surface level profiles.

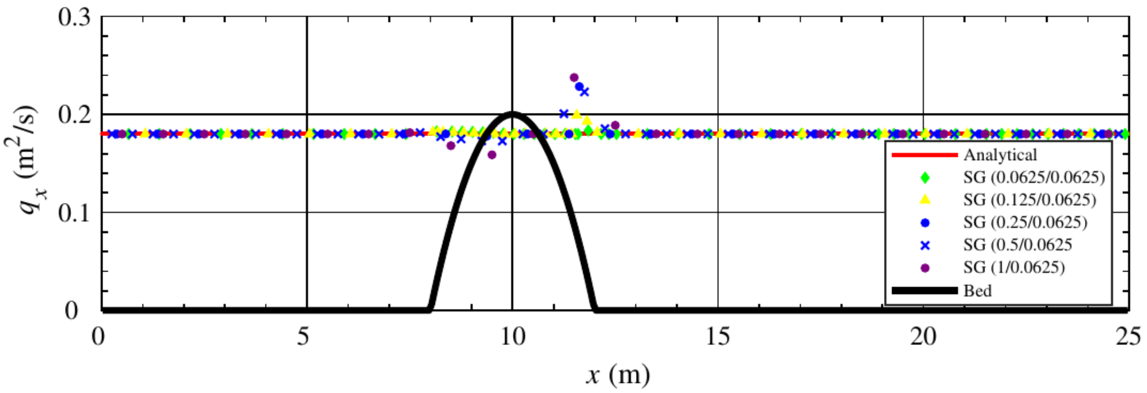


Figure 4.2: Test case 1, steady transcritical flow over a hump: longitudinal unit width discharge profiles.

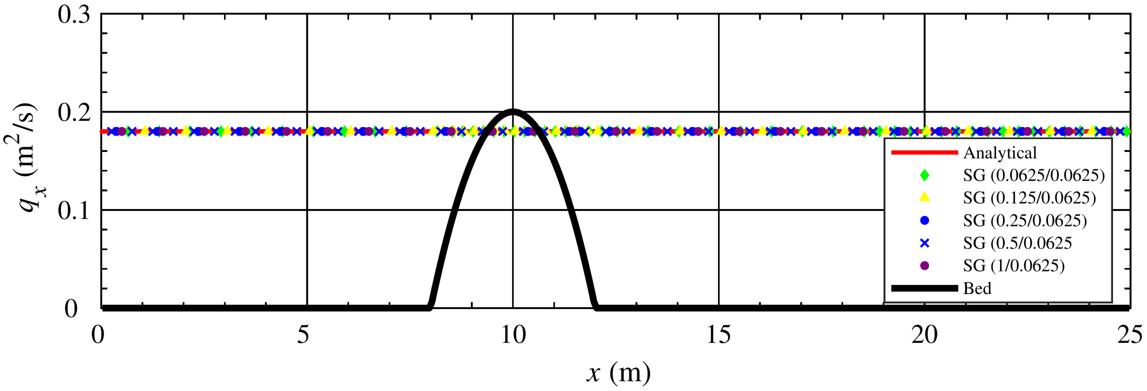


Figure 4.3: Test case 1, steady transcritical flow over a hump: longitudinal unit width discharge profiles, based on the modified solution proposed by Ying and Wang (2008).

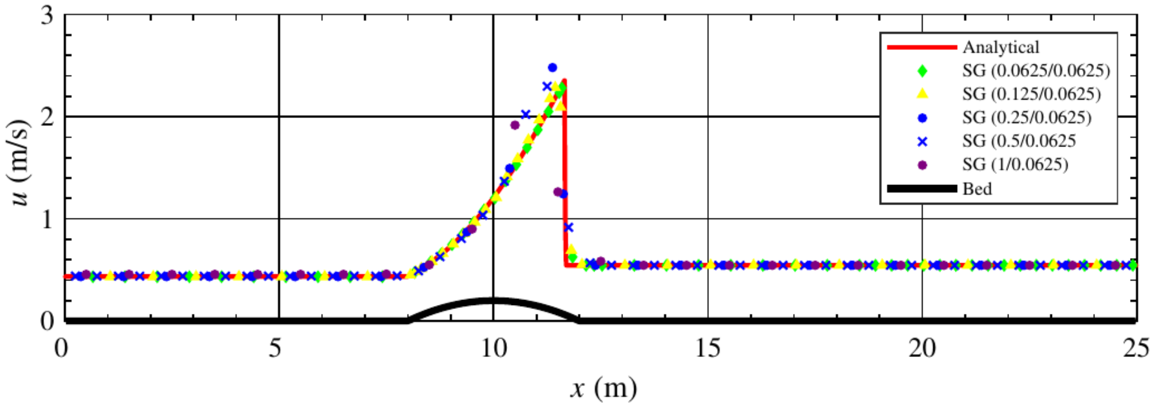


Figure 4.4: Test case 1, steady transcritical flow over a hump: longitudinal velocity profiles.

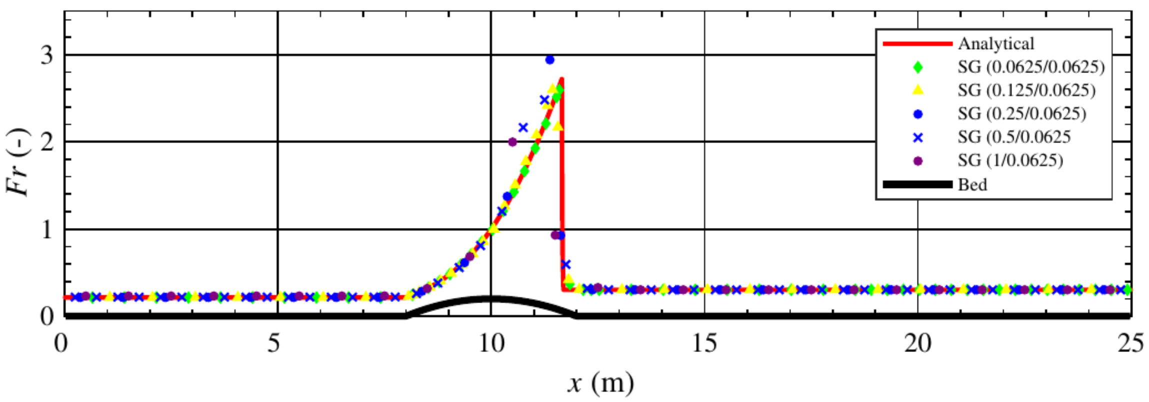


Figure 4.5: Test case 1, steady transcritical flow over a hump: longitudinal Froude number profiles.

In Chapter 2, the GCI method was reviewed. In this test case, the GCI method is applied to the solutions of water surface level. Three points along the channel have been taken, one at the hump (about the crest) and two others 5.5 m upstream and 5 m downstream, respectively from the crest of the hump. At the hump, the point is selected with a sufficient distance from the hydraulic jump at x=9.0 m as the GCI method is only definable for continuous solutions. The adopted computational grid resolutions in the GCI analysis are 0.25 m, 0.5 m, and 1.0 m, representing grid refinement ratio r=2. Table 4.1, presents the computed GCIs for both models and at the selected points. The predicted water surface level in all models/resolutions for the downstream point (p_3) is the same $\eta = 0.33$ m. Thus, the GCI is not computed for this point. The analysis shows that the results are converged to the asymptotic value based on the adopted grids.

Table 4.1: Test Case 1, steady transcritical flow over a hump: GCI analysis based on the predictions of water surface level.

model	sample points		Δx_1 (m)	Δx_2 (m)	Δx_3 (m)	GCI_{21}	GCI_{32}	$\frac{\mathrm{GCI}_{32}}{r^p \mathrm{GCI}_{21}}$
			0.25	0.5	1.0			
	number	x (m)	η_1 (m)	η_2 (m)	η_3 (m)			
T	p_1	4.5	0.427	0.422	0.408	1.004	2.614	1.013
	p_2	9.5	0.391	0.381	0.363	4.223	7.703	1.027
	p_3	15	0.330	0.330	0.330	-	-	-
SG	p_1	4.5	0.414	0.409	0.394	0.755	2.284	1.012
	p_2	9.5	0.393	0.385	0.360	1.447	4.294	1.023
	p_3	15	0.330	0.330	0.330	-	-	-

4.2 Test Case 2, dam-break problem

Chapter 2 reviewed analytical solutions to the dam-break problem based on the 2D shallow water equations. Here, the problem is adopted to evaluate the performance of the SG model in simulating the propagation of bores over wet and dry areas. In this test, a 1000 m long 1D frictionless channel is assumed. The channel is horizontal and the position of the idealised dam, which represents the location of initial discontinuity in hydraulic properties, is just in the middle of the channel. The initial condition is given by the still water level across the domain, and a null value of the velocity. The water depth of the upstream half of the channel is set to 3 m and two scenarios of the water depth is defined for the downstream half of the channel: i) h^R= 1 m (i.e. wet downstream) and ii) h^R= 0 m (i.e. dry downstream). Reflective boundary conditions are set at the two extreme ends of the channel. The simulations are implemented through the computational meshes at resolutions of 1 m, 2 m, 5 m, and 10 m, while in all cases, the small cell dimension is 1 m.

In Figs. 4.6 to 4.9, solutions of the SG model are compared to the analytical solutions. Figs. 4.6 and 4.7, which relates to the first scenario (i.e. wet downstream), show the longitudinal water depth and velocity, respectively. The results show that the position of the waves (shocks and rarefactions) has been captured accurately at all computational mesh resolutions. However, as would be expected, coarse meshes lead to numerical diffusion, in particular around the shock wave. A similar finding (i.e. the larger the cell, the higher the diffusion) is observed in the second scenario (dry downstream) as shown in Figs. 4.8 and 4.9. In Fig. 4.9, the unit width discharge (q_x) instead of velocity (u), which has employed in Fig. 4.7, has been depicted graphically. This is because the SG model provides solutions for the variables η, q_x, and q_y. Then the x-component of velocity is approximated as $u = \frac{q_x}{h}$ (in this case as the bed is flat and horizontal $\eta = h$) at post-processing. For the second scenario (dry downstream) as shown in Figs. 4.8 and 4.9, q_x and h tend to zero, which would lead to extremely high values of u by the SG model. Then for the second scenario, u was replaced by q_x. In this test case, as the channel is frictionless and entirely flat, there are no differences between the SG and T model for the same computational mesh (the results are exactly the same) and only the results of the SG model are then shown.

This test case involves unsteady flow. Then, the GCI analysis is performed for three time instants t=15 s, 30 s, and 45 s. As mentioned before, in this test case, the solutions of the SG and T models are exactly the same when similar grid resolutions are employed. Then the GCI analysis is independent of the model type. In this section, sample points are only taken from the profile of rarefaction wave at the mentioned time instants because the discontinuous water surface level at the shock (in the dam-break over downstream wet bed) invalidates the GCI analysis. In the GCI analysis, it is required that $\frac{\Delta x_2}{\Delta x_1} = \frac{\Delta x_3}{\Delta x_2} = r$ and the computational grids that have been introduced in this test cannot satisfy this expression. Then, along with $\Delta x = 1$ m, and $\Delta x = 2$ m, the solution at the resolution $\Delta x = 4$ m is included for only GCI analysis. Table 4.2 displays the results of the GCI analysis and confirms the convergence of the solutions toward an asymptotic value at each sample point for the used grids.

4.3 Test Case 3, dam-break wave propagating over three humps

In this test, a dam-break wave propagation over a complex topography, which contains three conical humps, is investigated. This test was originally proposed by Kawahara and Umetsu (1986) and later

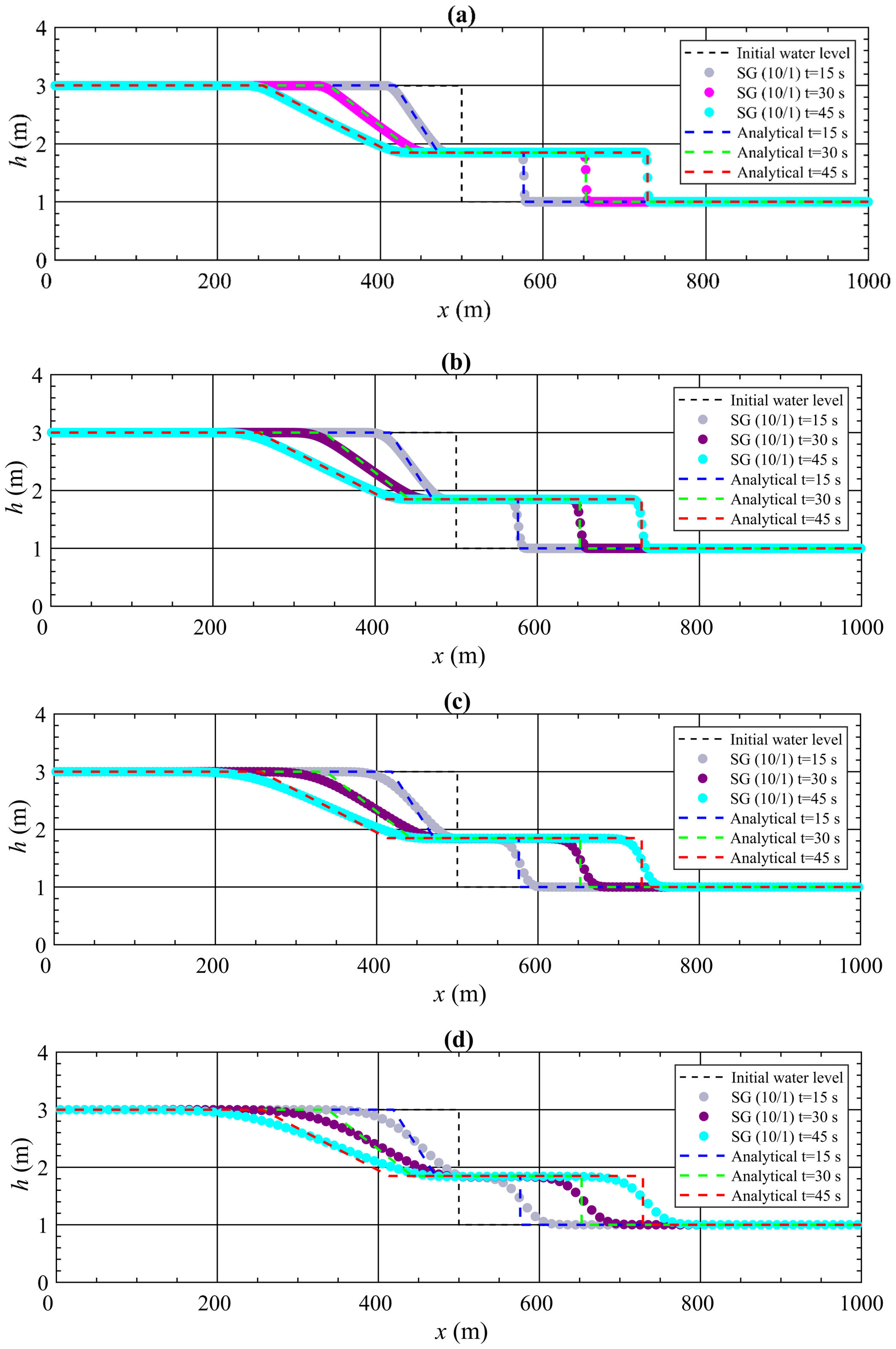


Figure 4.6: Longitudinal water depth profile in the dam-break problem over the wet downstream bed as defined in Test case 2 at the computational mesh resolutions, (a) 1 m, (b) 2 m, (c) 5 m, and (d) 10 m.

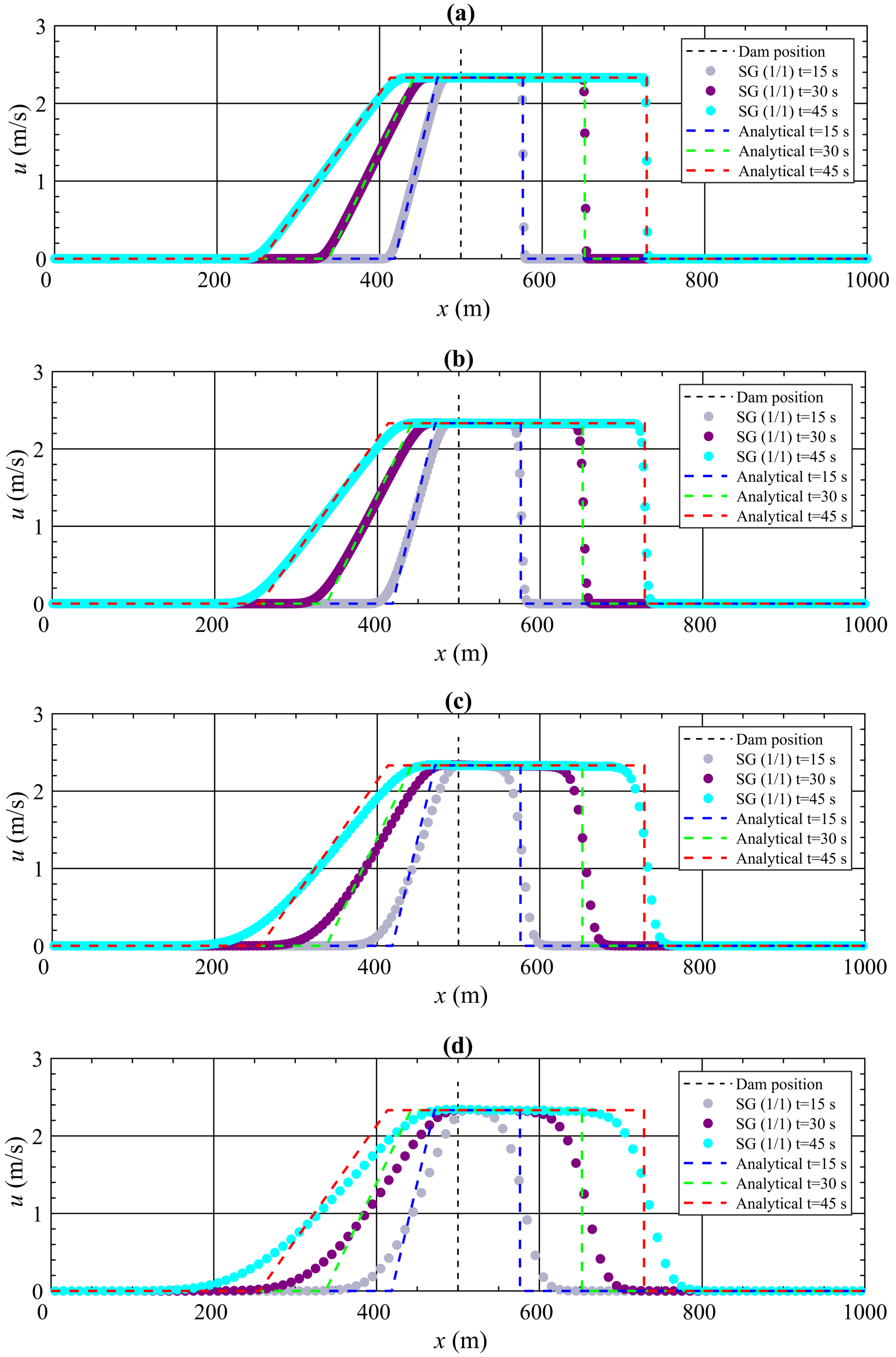


Figure 4.7: Longitudinal velocity profile in the dam-break problem over the wet downstream bed as defined in Test case 2 at the computational mesh resolutions, (a) 1 m, (b) 2 m, (c) 5 m, and (d) 10 m.

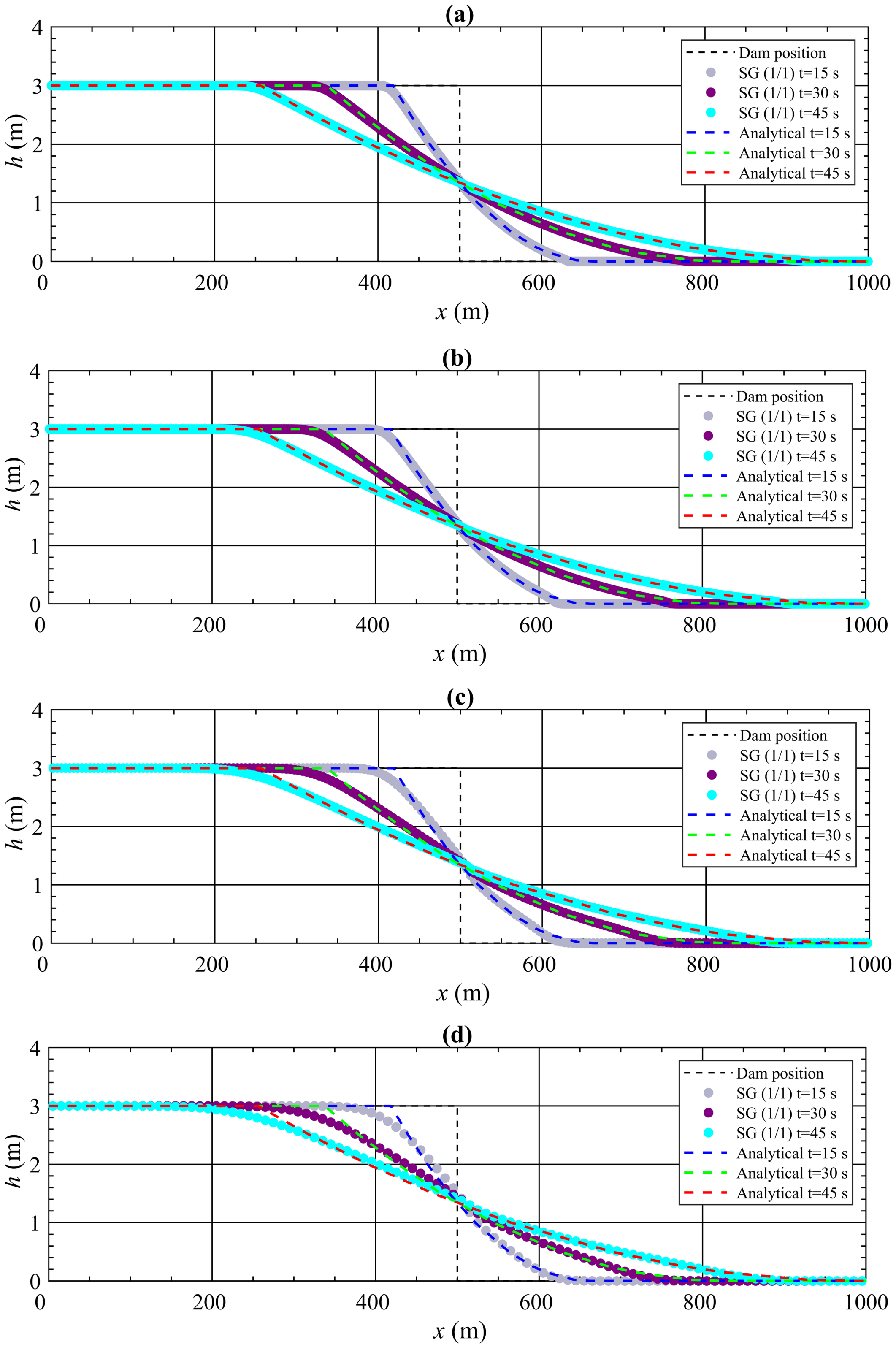


Figure 4.8: Longitudinal water depth profile in the dam-break problem over the dry downstream bed as defined in Test case 2 at the computational mesh resolutions, (a) 1 m, (b) 2 m, (c) 5 m, and (d) 10 m.

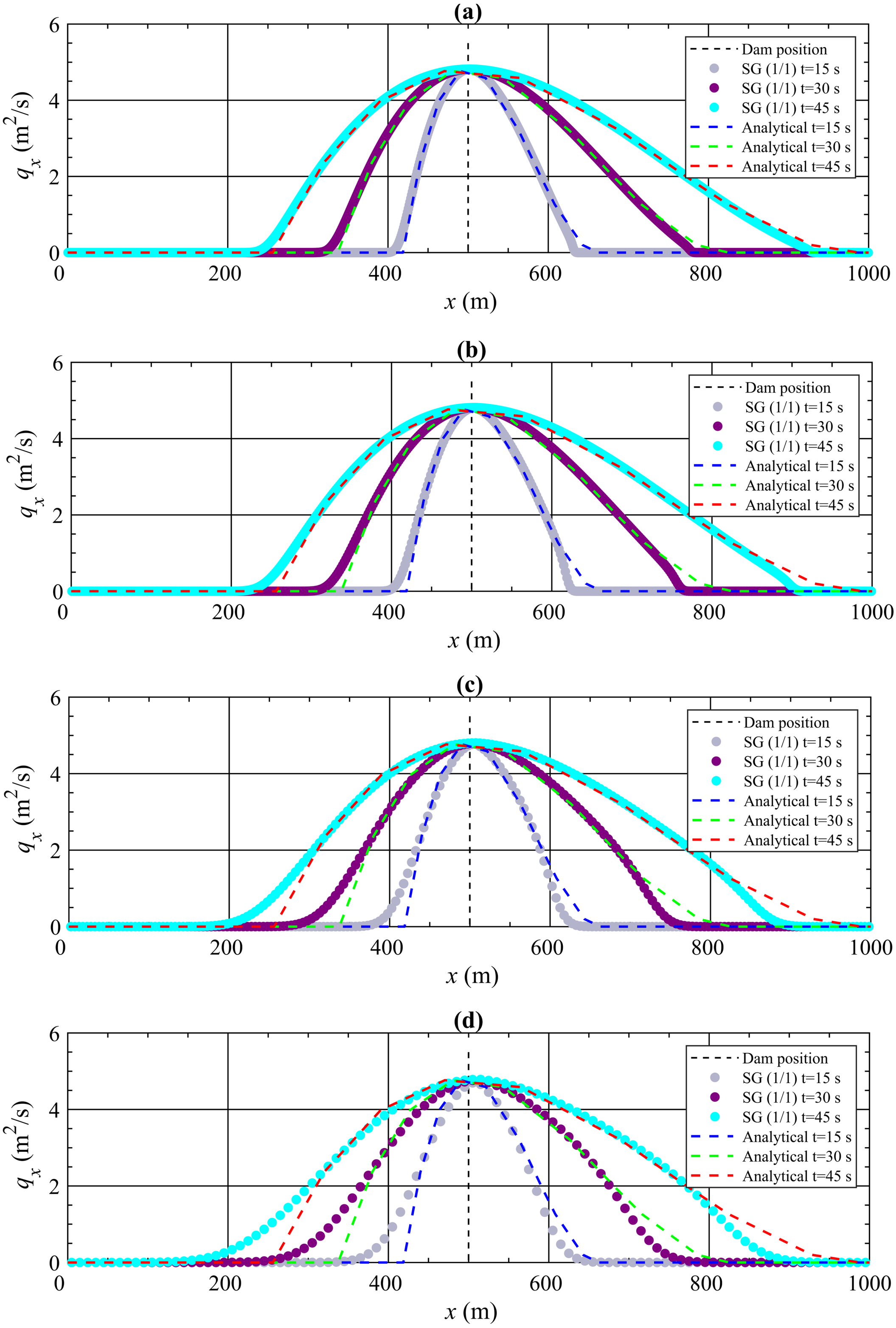


Figure 4.9: Longitudinal unit width discharge profile in the dam-break problem over the dry downstream bed as defined in Test case 2 at the computational mesh resolutions, (a) 1 m, (b) 2 m, (c) 5 m, and (d) 10 m.

Table 4.2: Test case 2, dam-break problem: GCI analysis based on the predictions of water surface level.

		sample points		Δx_1 (m)	Δx_2 (m)	Δx_3 (m)			
state	t (s)			1	2	4	GCI_{21}	GCI_{32}	$\frac{\text{GCI}_{32}}{r^p\text{GCI}_{21}}$
		number	x (m)	η_1 (m)	η_2 (m)	η_3 (m)			
dry	15	P_1	422	2.827	2.812	2.737	0.173	0.852	1.005
	15	P_2	578	0.368	0.374	0.401	0.511	2.414	0.985
	30	P_1	422	2.059	2.063	2.096	0.033	0.278	0.998
	30	P_2	578	0.773	0.774	0.783	0.078	0.361	0.998
	45	P_1	422	1.811	1.826	1.865	0.627	1.633	0.992
	45	P_2	578	0.940	0.939	0.936	0.254	0.437	1.001
wet	15	P_1	442	2.459	2.448	2.413	0.241	0.788	1.004
	30	P_1	442	1.907	1.927	1.979	0.777	2.040	0.990
	45	P_1	442	1.849	1.849	1.857	0.004	0.047	1.000

was adopted by many other researchers such as Liang and Borthwick (2009), Kesserwani and Liang (2010), and Mahdizadeh et al. (2012).

The domain of study is rectangular with the dimensions 75 m × 30 m, in which the topography is defined as follows

$$\begin{split} z(x,y) = \max\Big[0, 1-\tfrac{1}{8}\sqrt{(x-30)^2+(y-6)^2}, 1-\tfrac{1}{8}\sqrt{(x-30)^2+(y-24)^2}, \\ 3-\tfrac{3}{10}\sqrt{(x-47.5)^2+(y-15)^2}\Big]. \end{split} \tag{4.2}$$

The boundaries of the domain are assumed solid walls. The dam is situated between the points (x_d,0) and (x_d,30), where x_d=15 m. In many studies, including the aforementioned studies, x_d =16 m is adopted but herein as the test is simulated at different mesh resolutions, x_d =15 m is preferable to x_d =16 m (in terms of divisibility). Initially a still water state with the upstream (x<15 m) and downstream (x>15 m) depths 1.875 m and 0 m, respectively are set to the domain. The bed roughness is n_M= 0.018 s/m$^{\frac{1}{3}}$ across the domain.

This test is simulated using the T and SG model over computational meshes at the resolutions 0.5 m, 1 m, 1.5 m, and 3 m, while the small cell resolution is 0.1 m for all simulations performed by the SG model. Moreover, a simulation is performed through T (0.1), the solution of which is referred to as a benchmark solution as benefiting from a fine computational mesh.

Figs.4.10 to 4.14 illustrate a sequential plots of 3D water surface and 2D water depth distribution at times $t = 2$ s, $t = 6$ s, $t = 12$ s, $t = 30$ s, and $t = 300$ s, respectively only for the lowest and highest resolutions (for clarity). In all solutions, by t=2 s, the wet-dry front approaches the two small humps and even begins to climb them. At t= 6 s, the two shorter humps are entirely inundated and the wet-dry front reaches the highest mound. At t= 12 s, the small humps are still submerged and the highest hill has been submerged by the flood. The waves are approaching the downstream boundary and bore-like reflection waves, which have been created by the humps, are moving upstream. At t= 30 s, the waves have collided with the boundaries, so that the upstream waves have been damped, however the downstream waves are still active (a reverse movement of the waves toward the larges hump). By t= 300 s, the steady state is achieved so that the domain of study is entirely submerged except the peaks of the three humps.

An important remark must be made with regards to the distribution of depths in Figs.4.10 to 4.14. Although the SG model was not developed with the specific aim of capturing shocks as those appear in the dam-break problems (as this capturing requires high-resolution meshes and high order schemes), in this dam-break problem test, the SG model provides a more accurate water depth distribution compared to the T model at the same resolution because the domain involves a complex topography. This improved accuracy is clearer at low-resolution meshes [e.g. compare the water depth distribution in figures “c” with “e” (of Figs.4.10 to 4.14)].

Figs.4.15 to 4.18 show the distribution of the magnitude of unit width discharge ($\| \boldsymbol{q} \|$) across the domain of study. The distributions are provided for each model/resolution and at all time instants utilized for the water depth distribution, except for t=300 s, since the flood inundation at t=300 s reaches nearly a steady state with the zero (or very near to zero) velocity. The T and SG models do not explicitly provide any value of the cells’ velocity but $\boldsymbol{V} = \frac{\boldsymbol{q}}{\eta - z}$ is computed at post-processing. As this test case involves involves dam-break wave propagation over a dry bed, a wet-dry front propagates over a dry bed during the simulations, in which $\boldsymbol{q} \rightarrow \boldsymbol{0}$ and $\eta \rightarrow z$. Thus, unphysical values may be assigned to the cells’ velocity. To prevent that, herein the distribution of $\| \boldsymbol{q} \|$ instead of $\| \boldsymbol{V} \|$ is considered. According to Figs.4.15 to 4.18, the SG and T models’ solutions are very close to the benchmark solution but by decreasing the mesh resolution, the accuracy is degraded (for both the SG and T model). Although in the SG model (similar to the T model), q_x and q_y are constant across each computational cell, in partially wet cells, the SG model sets the q_x and q_y to zero in dry bed (i.e. in a large cell, $\boldsymbol{q} = \boldsymbol{0}$ in unsubmerged small cells and all other submerged small cells have the same $\boldsymbol{q}$), which leads to more accurate representation of $\boldsymbol{q}$. This advantage is better observed by comparing the figures “c” and “e” of Figs.4.15 to 4.18.

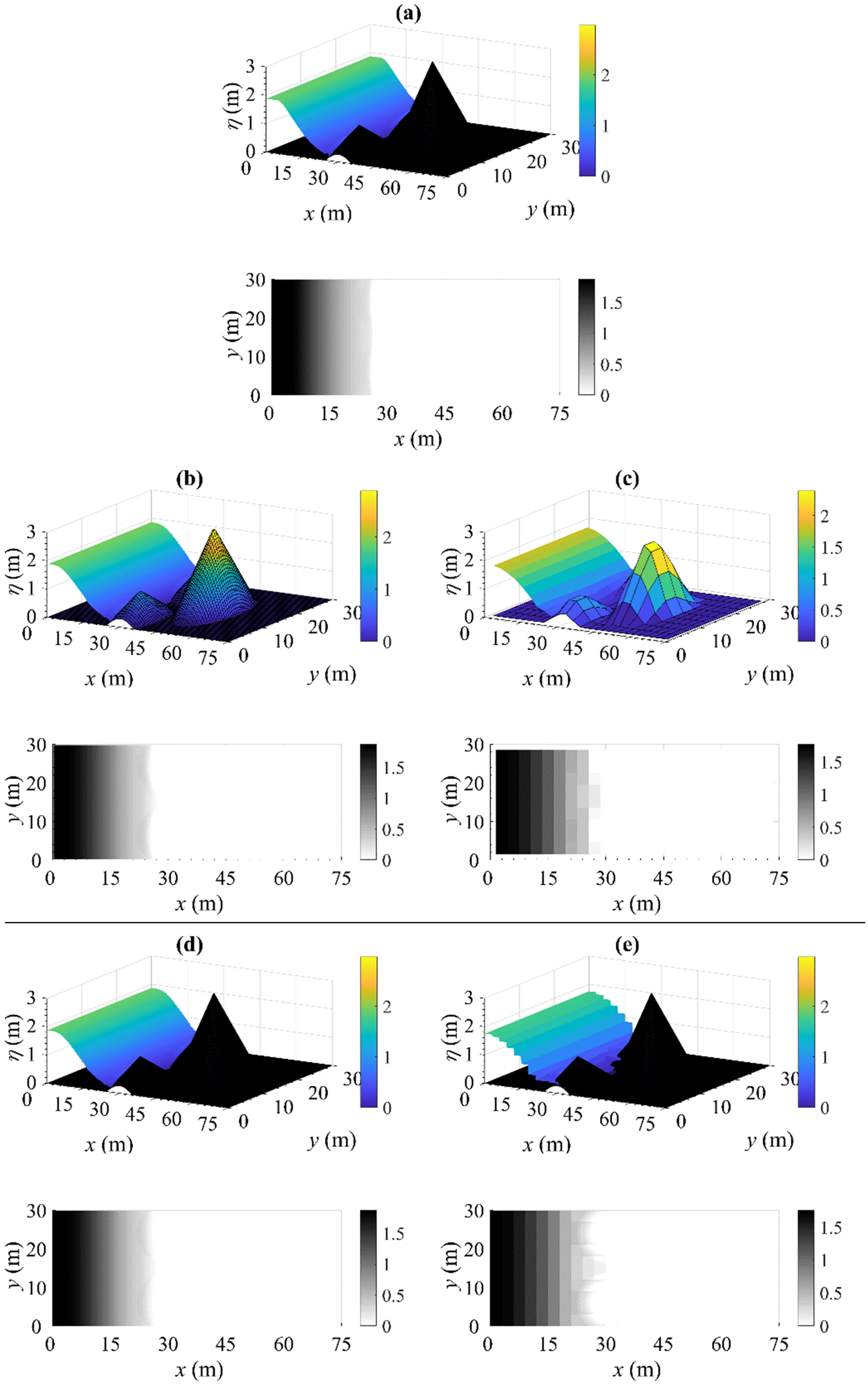


Figure 4.10: Dam-break wave propagation over three humps: 3D view of water surface level and water depth distribution at t=2 s for model/resolution, (a) T (0.1), (b) T (0.5), (c) T (3), (d) SG (0.5/0.1), (e) SG (3/0.1).

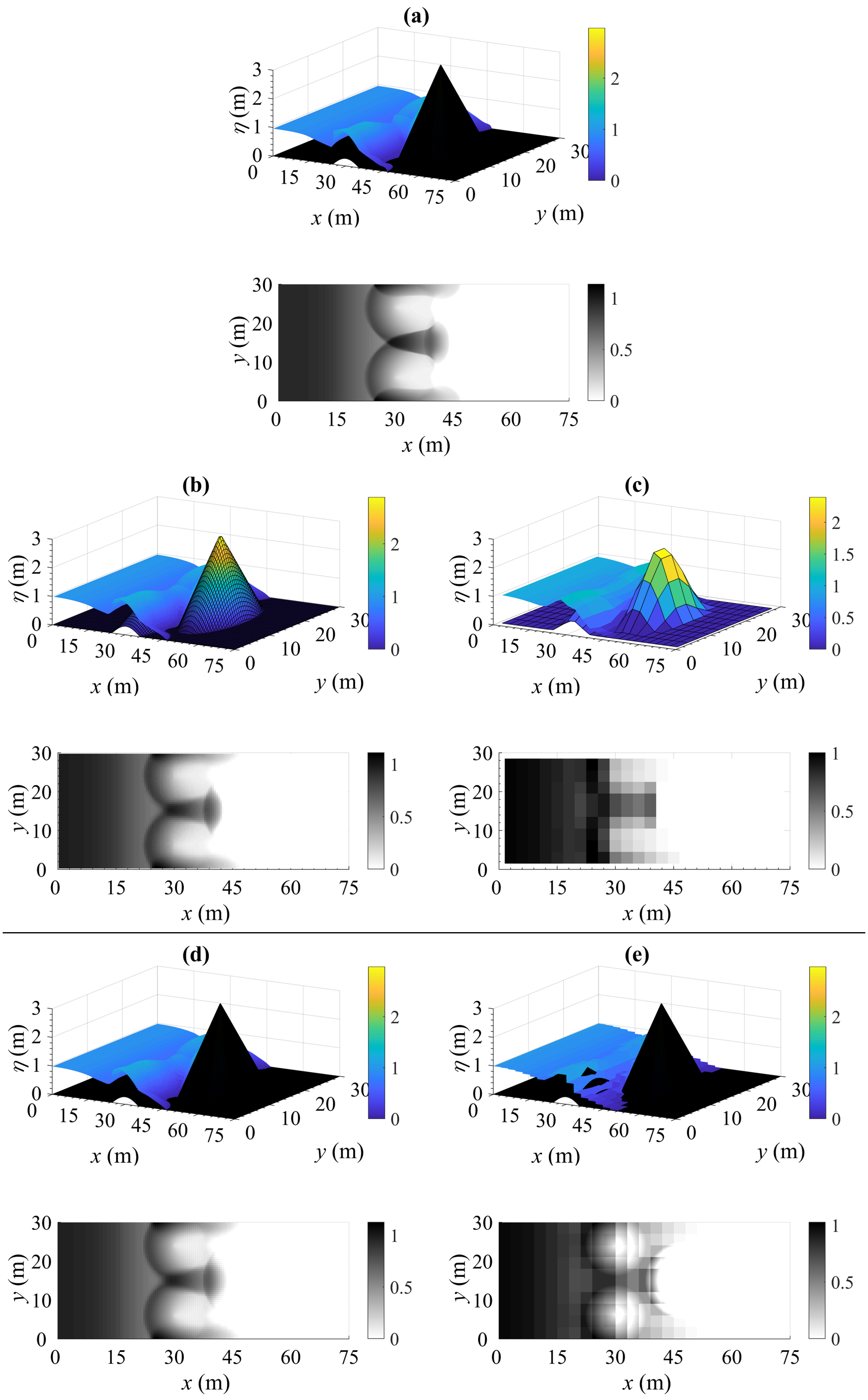


Figure 4.11: Dam-break wave propagation over three humps: 3D view of water surface level and water depth distribution at t=6 s for model/resolution, (a) T (0.1), (b) T (0.5), (c) T (3), (d) SG (0.5/0.1), (e) SG (3/0.1).

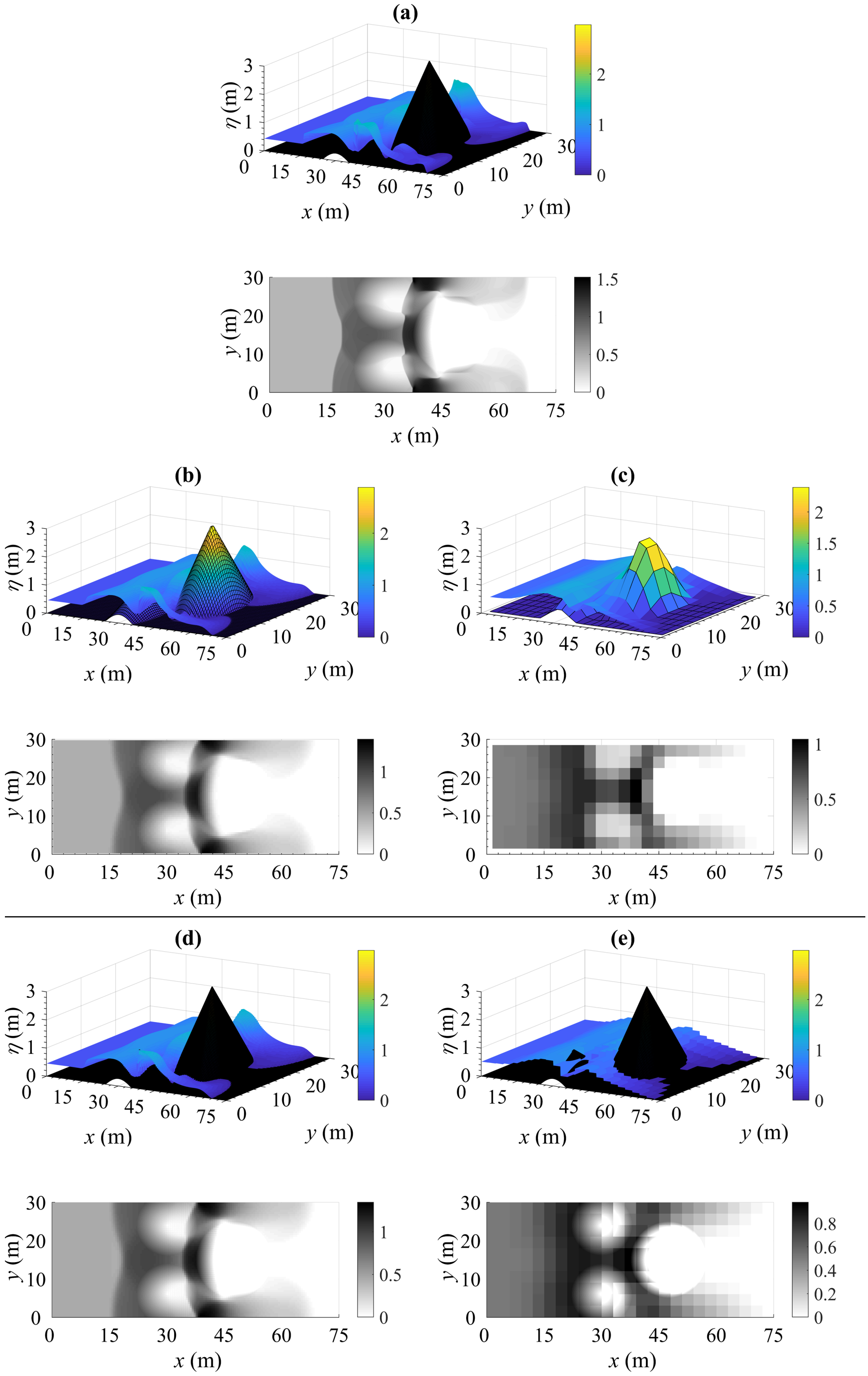


Figure 4.12: Dam-break wave propagation over three humps: 3D view of water surface level and water depth distribution at t=12 s for model/resolution, (a) T (0.1), (b) T (0.5), (c) T (3), (d) SG (0.5/0.1), (e) SG (3/0.1).

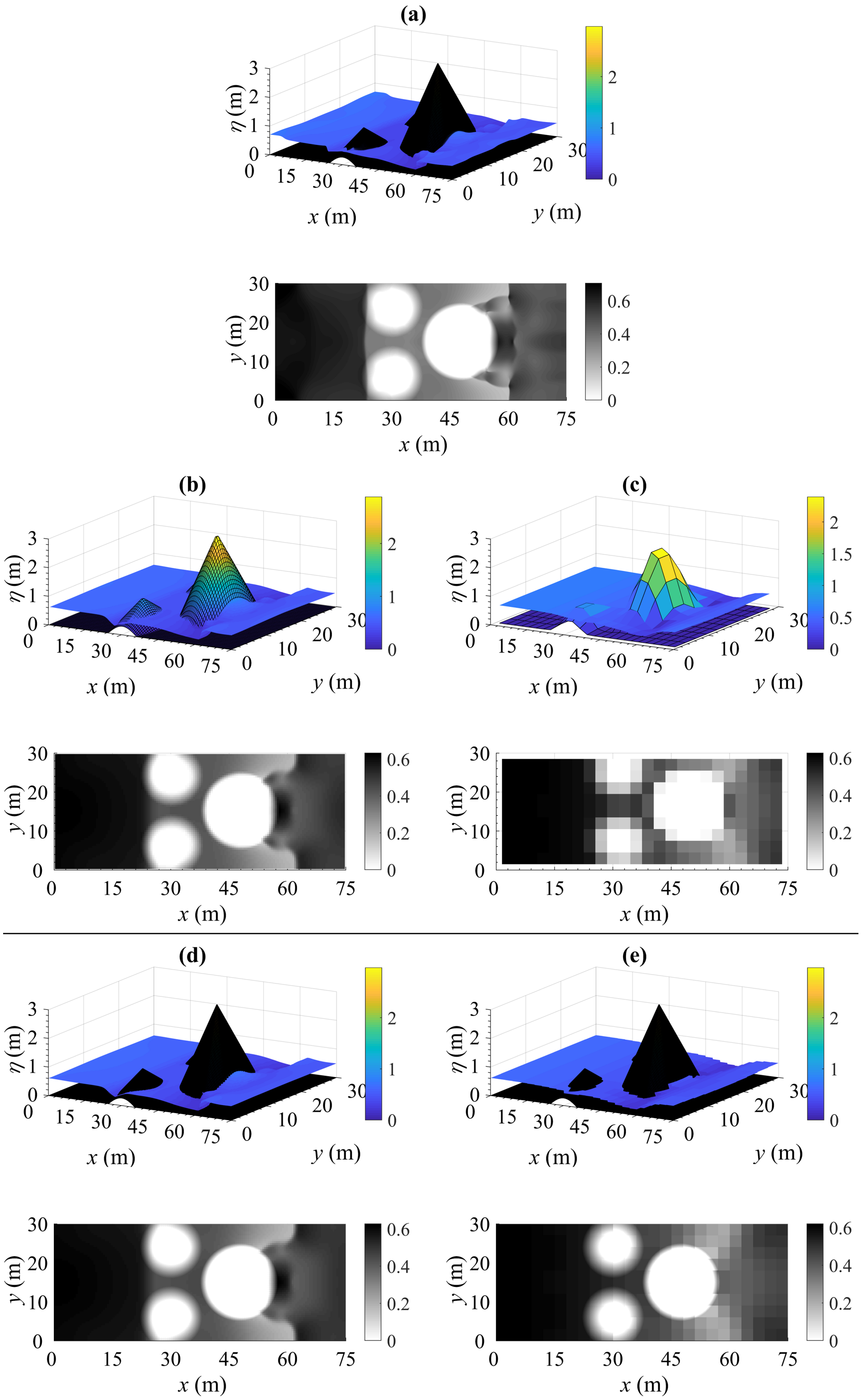


Figure 4.13: Dam-break wave propagation over three humps: 3D view of water surface level and water depth distribution at t=30 s for model/resolution, (a) T (0.1), (b) T (0.5), (c) T (3), (d) SG (0.5/0.1), (e) SG (3/0.1).

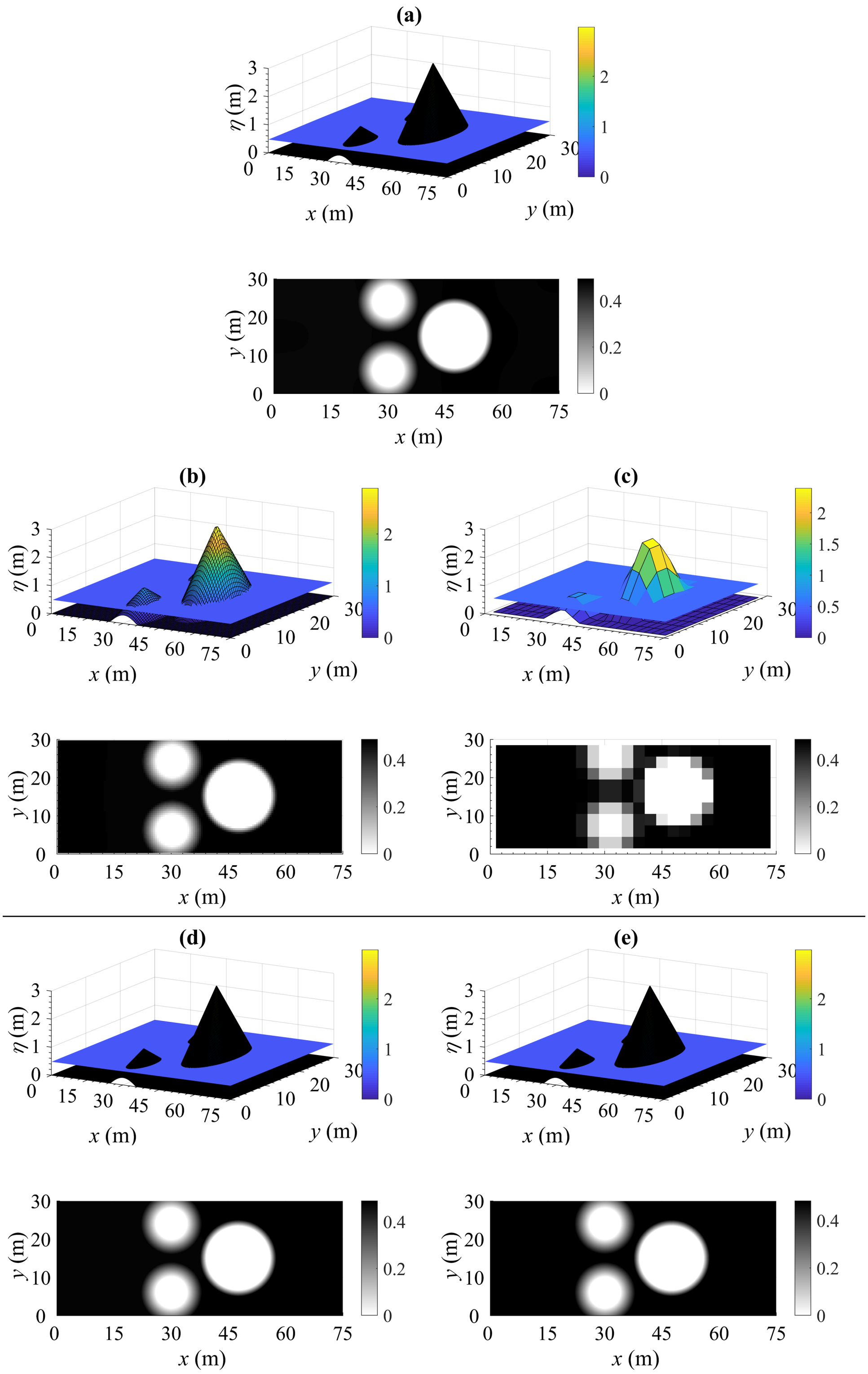


Figure 4.14: Dam-break wave propagation over three humps: 3D view of water surface level and water depth distribution at t=300 s for model/resolution, (a) T (0.1), (b) T (0.5), (c) T (3), (d) SG (0.5/0.1), (e) SG (3/0.1).

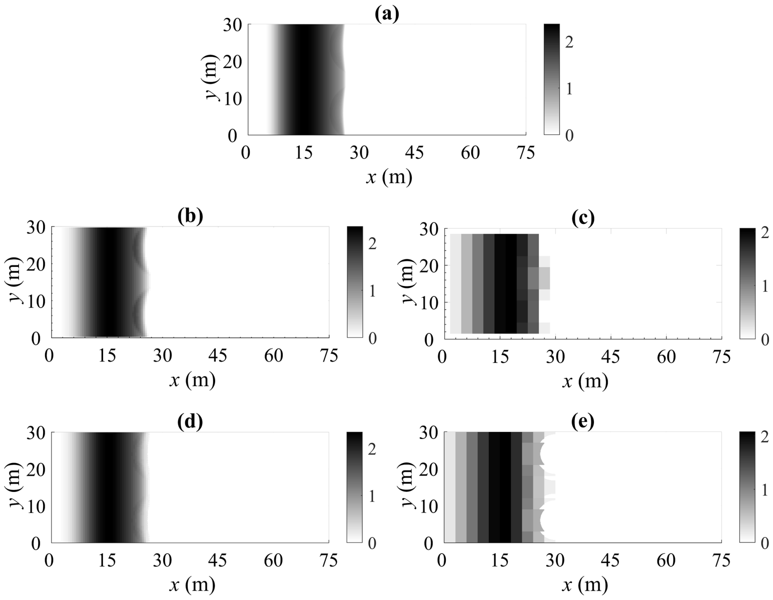


Figure 4.15: Dam-break wave propagation over three humps: distribution of the magnitude of unit width discharge (‖ $\boldsymbol{q}$ ‖) across the domain at t= 2 s for model/resolution, (a) T (0.1), (b) T (0.5), (c) T (3), (d) SG (0.5/0.1), (e) SG (3/0.1).

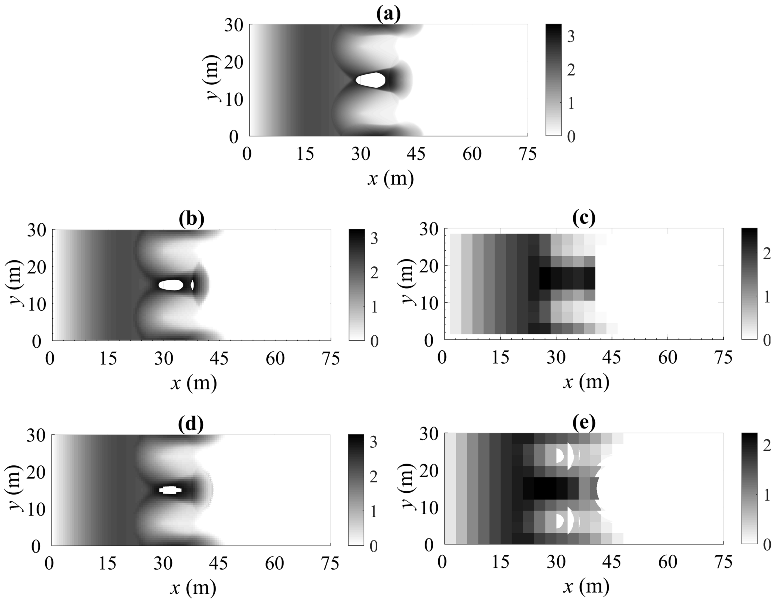


Figure 4.16: Dam-break wave propagation over three humps: distribution of the magnitude of unit width discharge (‖ $\boldsymbol{q}$ ‖) across the domain at t= 6 s for model/resolution, (a) T (0.1), (b) T (0.5), (c) T (3), (d) SG (0.5/0.1), (e) SG (3/0.1).

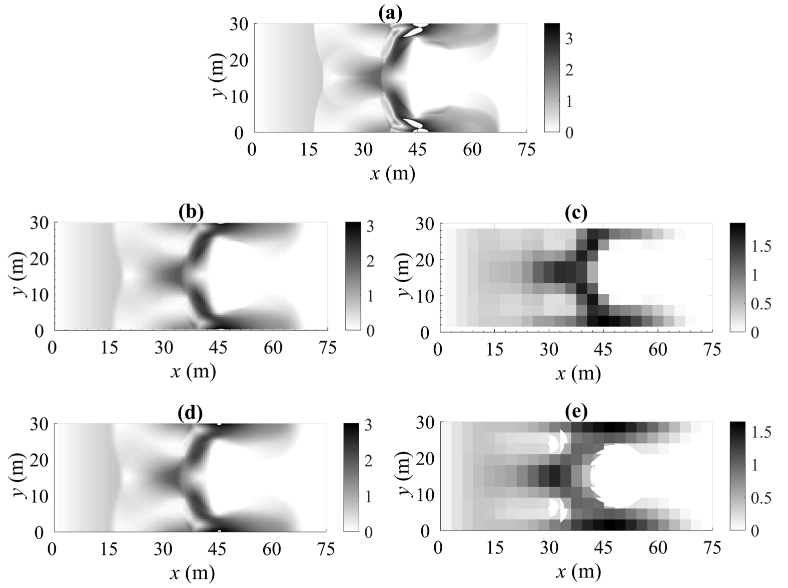


Figure 4.17: Dam-break wave propagation over three humps: distribution of the magnitude of unit width discharge (‖ ***q*** ‖) across the domain at *t*= 12 s for model/resolution, (a) T (0.1), (b) T (0.5), (c) T (3), (d) SG (0.5/0.1), (e) SG (3/0.1).

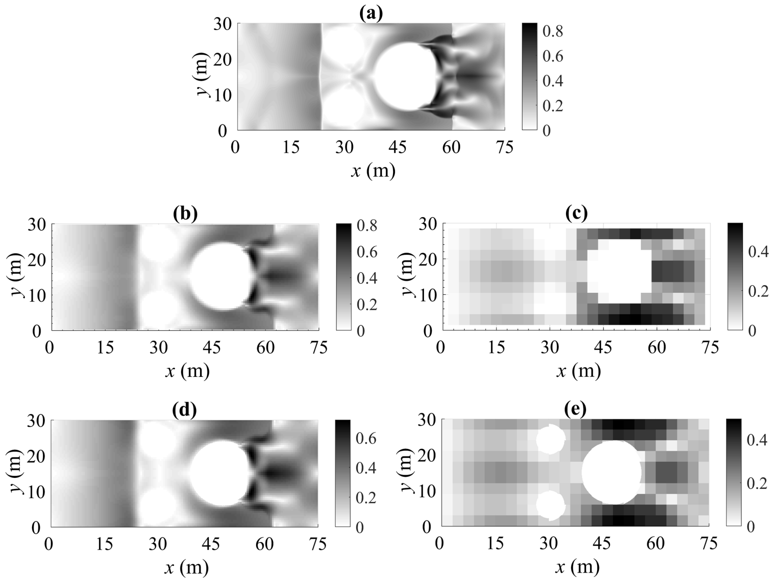


Figure 4.18: Dam-break wave propagation over three humps: distribution of the magnitude of unit width discharge (‖ ***q*** ‖) across the domain at *t*= 30 s for model/resolution, (a) T (0.1), (b) T (0.5), (c) T (3), (d) SG (0.5/0.1), (e) SG (3/0.1).

In Chapter 2, the finite volume method was compared to the certain well-known numerical methods, such as finite difference and finite element. A main advantage of the finite volume is that it guarantees the mass and momentum conservation. In this test case, the performance of the T and SG models in mass conservation is evaluated. Based on the initial conditions, the existing water volume in the domain of study is 15×30×1.875=843.75 m^3. This volume must be constant during the simulation as the boundaries are confined and there is no inflow and outflow to the domain. Table 4.3 tabulates the violation of the mass conservation for all simulations. It is clearly found from Table 4.3 that the T and SG models satisfy the mass conservation law as the maximum violation is less than 0.0045%.

In the first chapter, the C-property was discussed in brief. A model is well-balanced (i.e. satisfies C-property) if it is able to simulate still water without any oscillation. In this test, at t=300 s, the dam-break waves reach a still steady state. The simulations were continued by t=3000 s and no oscillations were observed. This confirms that the T and SG models adopted in this research are well-balanced.

For analysing the GCI of this test case using the solutions of water surface level, three time instants t=6 s, 12 s, and 30 s are considered. The solutions at t= 2 s and t=30 s are excluded from the GCI study. This is because, at t=2 s the extension of dam-break's wave propagation is limited and at t=300 s, the solution involves a stationary state with constant water surface and null velocity across the domain. The GCI analysis requires $\frac{\Delta x_2}{\Delta x_1} = \frac{\Delta x_3}{\Delta x_2} = r$, and this equality does not exist among the computational grids considered. Thus, the solution of T (0.75) [along T (1.5) and T (3), which are already available] is added to the process of the GCI analysis. However, because of divisibility (large cell's size must be a multiple of small cell's size), the SG (0.75/0.1) is impossible to run and the GCI analysis is only performed for the T model. The analysis is provided for three points p$_1$, p$_2$, and p$_3$, which are located at the upstream part of the domain, the downstream part of the domain, and next to the largest hump, respectively. Table 4.4 shows the location of these three points. Table 4.4 also illustrates the information of the GCI analysis. The results verify the convergence of the solutions at the adopted grids.

4.4 Test Case 4, open channel flow around a bend

The aim of this test case is to test the ability of the sub-grid model to capture a 90° change of flow direction within a large cell. Fig. 4.19 shows the computational domain, which represents a simple topography that includes a 4 m wide channel that changes direction by 90°. The grid resolution of the SG model is 16/2, i.e., coarse cells are eight times coarser than the width of the channel. The results

Table 4.3: Violation of mass conservation in the SG and T models.

ΔX (m)	t(s)	T	SG
		error (%)	error (%)
0.1	2	3.6E-07	-
	6	7.9E-06	-
	12	7.0E-02	-
	30	7.0E-02	-
	300	7.0E-02	-
0.5	2	2.7E-06	4.7E-07
	6	3.3E-06	2.6E-06
	12	3.6E-07	2.0E-04
	30	1.2E-07	9.1E-04
	300	3.6E-07	1.3E-03
1	2	7.1E-06	1.1E-05
	6	5.6E-05	9.5E-06
	12	5.6E-05	5.4E-04
	30	5.5E-05	3.0E-03
	300	5.4E-05	3.9E-03
1.5	2	7.9E-06	1.9E-06
	6	2.1E-06	1.2E-05
	12	3.8E-06	7.3E-04
	30	1.1E-06	3.4E-03
	300	4.3E-06	4.5E-03
3	2	1.1E-05	5.6E-06
	6	4.3E-06	4.9E-06
	12	2.1E-06	6.3E-04
	30	4.3E-06	9.2E-04
	300	1.5E-05	1.6E-03

Table 4.4: Test Case 3, dam-break wave propagating over three humps: GCI analysis based on the predictions of water surface level.

	sample points			Δx_1 (m)	Δx_2 (m)	Δx_3 (m)			
t (s)				0.75	1.5	3	GCI_{21}	GCI_{32}	$\frac{\mathrm{GCI}_{32}}{r^p \mathrm{GCI}_{21}}$
	number	x (m)	y (m)	η_1 (m)	η_2 (m)	η_3 (m)			
6	P_1	15	15	0.837	0.841	0.860	0.193	0.832	0.995
	P_2	62	15	0.000	0.000	0.000	-	-	-
	P_3	48	25	0.011	0.013	0.042	9.773	46.211	0.841
12	P_1	15	15	0.453	0.502	0.613	10.906	22.107	0.902
	P_2	62	15	0.000	0.000	0.000	-	-	-
	P_3	48	25	0.260	0.275	0.389	1.210	8.260	0.943
30	P_1	15	15	0.615	0.613	0.610	2.652	3.173	1.004
	P_2	62	15	0.374	0.373	0.347	0.001	0.080	1.001
	P_3	48	25	0.314	0.322	0.365	0.853	4.143	0.974

of this simulation are compared against those obtained with the T model runs with resolutions of 2, 4, 8 and 16 m (Fig. 4.20 shows the meshes). The bed elevation of the white small cells (main channel) is z_c=0 m, and two scenarios are simulated in which the elevation of the green cells (floodplains) is set to z_f=1.0 and z_f=1.5 m. The value of Manning's coefficient $n_M = 0.01\ s/m^{\frac{1}{3}}$ is constant across the domain. Initially, the domain is entirely dry. A constant discharge $Q = 1.6\ m^3/s$ is uniformly introduced along the edges of white cells on the west side of the domain and a constant water depth 0.5 m is set at the edges of the white cells at the south part of the grid. The model is run until a steady flow regime is reached. The value of the CFL number used for all simulations described in this section is set to 0.75.

Fig. 4.21 presents the longitudinal profiles of water surface elevation obtained along the red line shown in Fig. 4.19. These profiles are presented for the different model types, resolutions, and two floodplain elevations (i.e. the green area scenarios in Fig. 4.19) of 1 m (Fig. 4.21.a) and 1.5 m (Fig. 4.21.b). The results of the finest resolution simulation T (2), which are herein used as a benchmark, predict flow only in the low elevation cells (i.e. main channel with z_c=0 m) and never in the higher elevation areas (for both elevations of the floodplain, i.e. z_f=1.0 and z_f=1.5 m). Therefore, solutions are expected to be independent of the elevation set to the higher elevation cells (floodplain). However, the results from the simulations performed by the Traditional model at coarse resolutions with two floodplain elevations differ substantially. Namely, whereas for z_f=1.0 m the elevation of

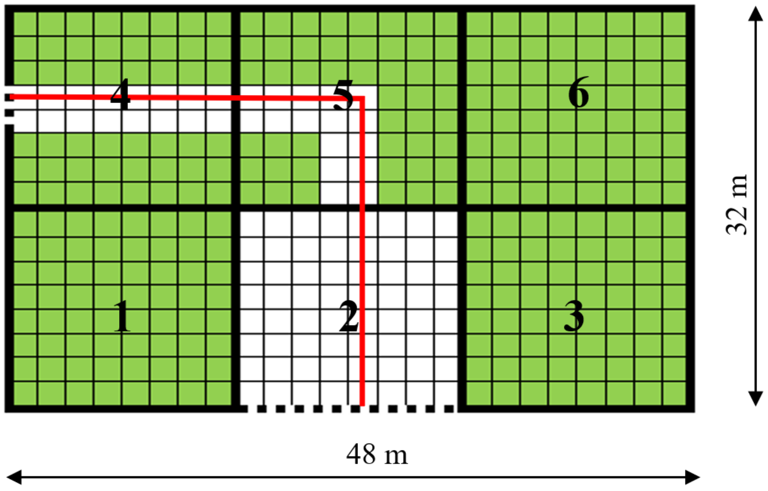


Figure 4.19: Test Case 4, open channel flow around a bend: the used computational domain and SG grid. The results of the model are reported along the red line indicated in the figure.

free surface is less than 1.0 m for all resolutions tested, the increase in floodplain height to z_f=1.5 m results in substantially higher values (∼ 50% higher for the coarser resolution). The reason for these different solutions is related to how the bed elevation was assigned to these cells. The coarse resolution cell elevation was obtained from averaging of the 2 m resolution bed elevation over the area of the coarsened cell. Therefore, in the coarsened grid the elevation of the region representing the channel depends on the values of z_f and z_c. The results in Fig. 4.21 also illustrate how the accuracy of solutions is reduced, as the grid is made coarser for simulations performed with a traditional model. On the other hand, results obtained with the sub-grid model at resolution 16/2 are very close to the results from the finely resolved traditional model [i.e. Root Mean Square Error (RMSE) =0.05 m for SG (16/2) against RMSE=0.21 m and RMSE=0.51 m of T (16) at floodplain scenarios of z_f=1.0 mm and z_f=1.5 m, respectively]. Since all simulations in this test case involve only a small number of cells, the run time was extremely short and is not analysed here.

Fig. 4.22 shows the profile of the magnitude of velocity vector along the red line, shown in Figs. 4.19 and 4.20, for each model/resolution. It is simply found that the most similar solution to the benchmark solution corresponds to SG (16/2), which uses the lowest computational resolution. In the simulations T (4), T (8), and T (16), the highest discrepancy between the numerical predictions and benchmark solution occurs in the upstream part of the domain, where the T model at low-resolution meshes is unable to involve the microtopography in the process of the computations. It is concluded from Figs. 4.21 and 4.22 that in practice to obtain accurate simulations using the traditional model, the maximum computational mesh size may is restricted by the size of the main channel, while this

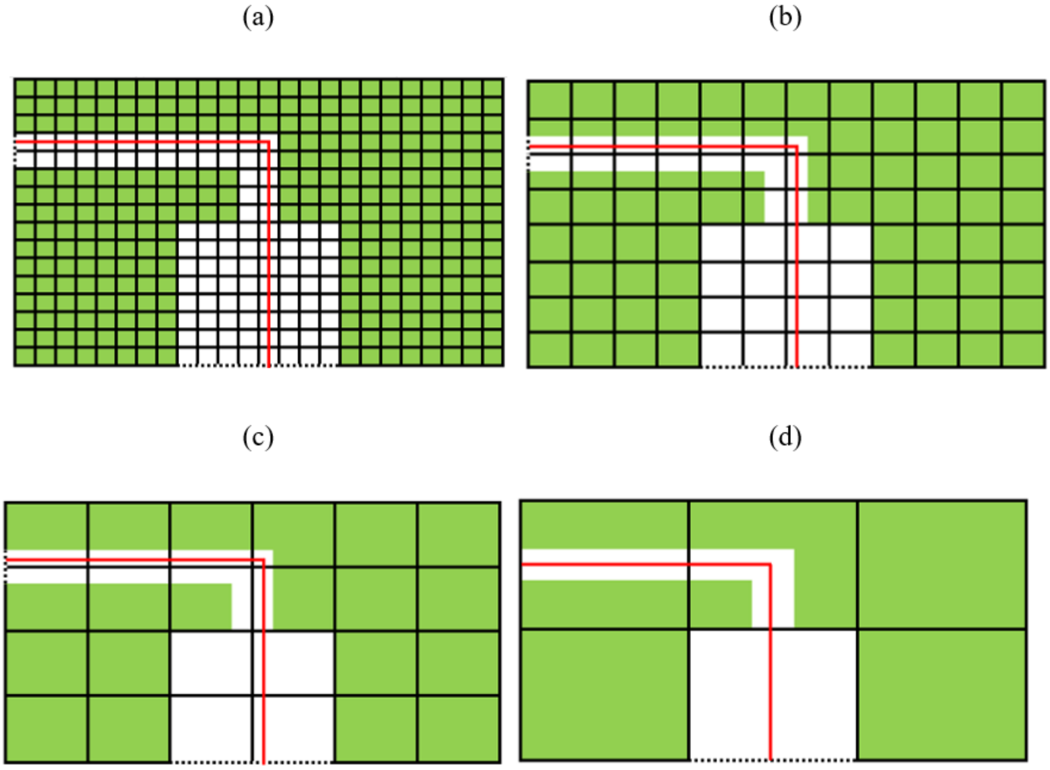


Figure 4.20: Test Case 4, open channel flow around a bend: computational meshes used for the simulations performed with T, (a) T (2), (b) T (4), (c) T (8) and (d) T (16).

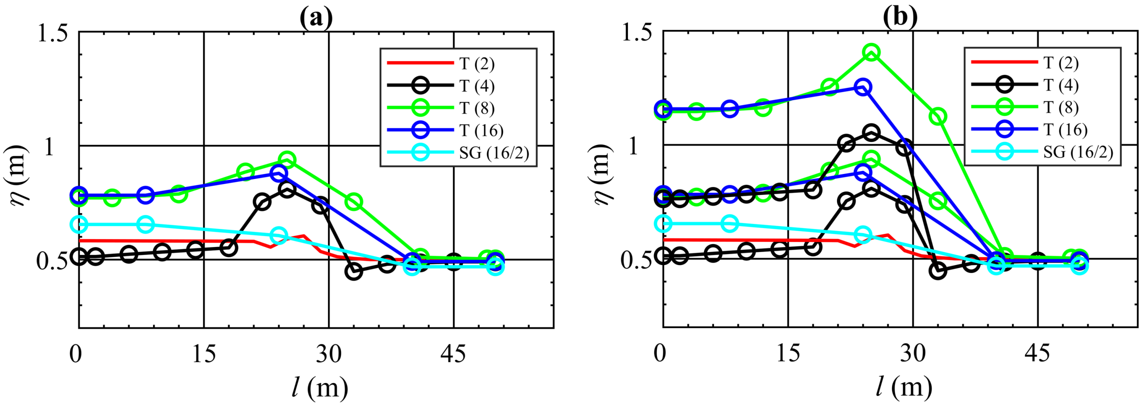


Figure 4.21: Test Case 4, open channel flow around a bend: longitudinal water surface level profiles obtained with the two models at various resolutions. (a) z_f=1.0 m and (b) z_f=1.5 m. Notice that the results of T (2) and SG (16/2) obtained with z_f=1.0 m and z_f=1.5 m are unchanged by choice of z_f.

restriction is invalid for the SG model.

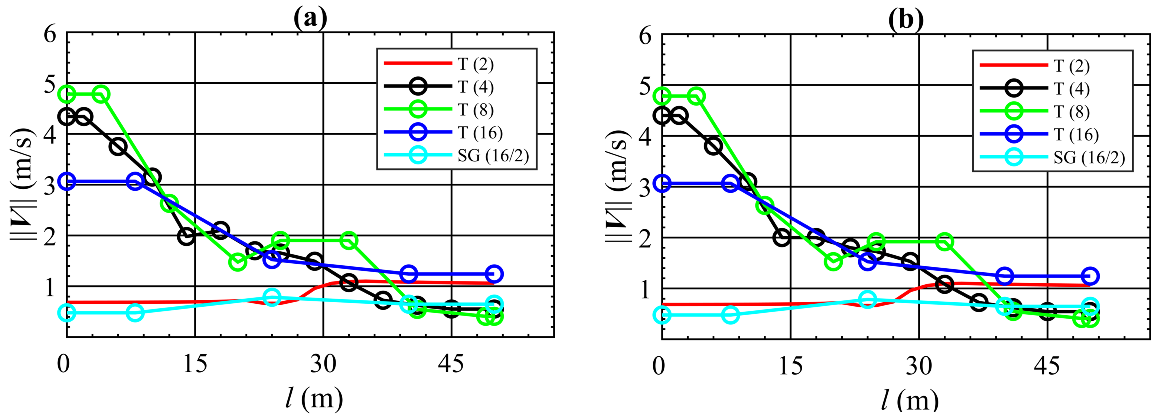


Figure 4.22: Test Case 4, open channel flow around a bend: longitudinal profile of the magnitude of velocity vector obtained with the two models at various resolutions. (a) z_f=1.0 m and (b) z_f=1.5 m. Notice that the results of T (2) and SG (16/2) obtained with z_f=1.0 m and z_f=1.5 m are unchanged by choice of z_f.

In this test case, the GCI analysis is only conducted for SG model predictions because the T model gives inaccurate results in this test (previously discussed). The GCI analysis involves SG (16/2) and two new solutions, performed on SG (8/2) and SG (4/2), representing a grid refinement ratio r=2. At the taken computational grid resolutions, the solutions of the SG model are independent of the scenarios. Then, the GCI analysis, which is only conducted for the SG model is also independent of the scenarios. The analysis is carried out for three points $\mathrm{p}_1, \mathrm{p}_2$, and p_3, which are at the centre of cells 4, 5, and 2 of SG (16/2), respectively (see Fig. 4.19) representing the cell numbers). These three points are all submerged cells in the solution of SG (16/2). Table 4.5 summarises the results of the GCI analysis. This table reveals that at the adopted grids the solutions converge to the asymptotic values.

4.5 Test Case 5, flow over an irregular bed

The fifth test case aims to assess the performance of the developed sub-grid model for the solution of problems involving relatively large variation of topography, including areas of adverse (negative) bed slope. The test case was created using the inverse method described in Chapter 2. The technique uses an inverse solution for the one-dimensional steady non-uniform flow whereby the bed longitudinal profile corresponding to the pre-defined depth profile is obtained. A 10 m wide, 3 km long prismatic

Table 4.5: Test Case 4, open channel flow around a bend: GCI analysis based on the predictions of water surface level.

sample points		Δx_1 (m)	Δx_2 (m)	Δx_3 (m)			
		4	8	16	GCI_{21}	GCI_{32}	$\frac{\text{GCI}_{32}}{r^p\text{GCI}_{21}}$
number	l (m)	η_1 (m)	η_2 (m)	η_3 (m)			
P_1	8	0.609	0.627	0.654	7.226	10.610	0.971
P_2	24	0.616	0.613	0.606	0.408	1.022	1.005
P_3	40	0.494	0.485	0.469	2.452	4.760	1.018

rectangular channel with constant roughness $n_M = 0.05 \text{ s/m}^{\frac{1}{3}}$ and flow discharge $Q = 180 \text{ m}^3/\text{s}$ is adopted in this test case. The water depth varies periodically according to the expression $h = 8 + 1.5 \sin^5\left(\frac{\pi x}{500}\right)$, which produces Froude number values of 0.2 and 0.35 along the channel. The solution to the inverse problem leads to the bed profile shown in Fig. 4.23a. Simulations were performed with grid resolutions of 20, 50, 100 and 200 m with both traditional and sub-grid (coarse) meshes, while 1 m^2 small cells were used for all sub-grid simulations. The value of CFL for all simulations was set to 0.9.

Figs. 4.23(a to c) present the results of selected simulations (for clarity, only solutions on coarse grids are shown) at the centre of the channel, along with the corresponding analytical solution. Fig. 4.23d illustrates the absolute error ($|E|$) between analytical and numerical water surface solutions. Further error analysis is provided by Table 4.6, where the maximum and RMS errors between the two models and the analytical solution are listed. At fine resolution, the accuracy of both models is within the order of cm, with the SG model delivering the best accuracy. For example, at the mesh resolution 20 m, the largest value of the maximum absolute error and RMSE for both models is restricted to 0.06 m and 0.05 m, respectively. The improved accuracy of the SG over the T model becomes apparent as the grid resolution is coarsened. For example, at 100 m resolution, the maximum absolute errors of T and SG models are 0.26 and 0.06 m, respectively. It is important to observe that at the resolution of 200 m, the results of the proposed sub-grid model are more accurate than the results of the Traditional model run at 50 m resolution. This gets clearer when the results of SG (200/1) and T (50) are compared against each other in terms of the maximum absolute error (i.e. 0.13 m vs 0.19 m) and RMSE (i.e. 0.08 m vs 0.1 m). The improved accuracy of the SG model is also shown in Fig. 4.23d.

In Fig. 4.24, the longitudinal profile of the velocity (solely the u component as the test is 1D) of the analytical solution is compared against the numerical solutions, which have been performed on

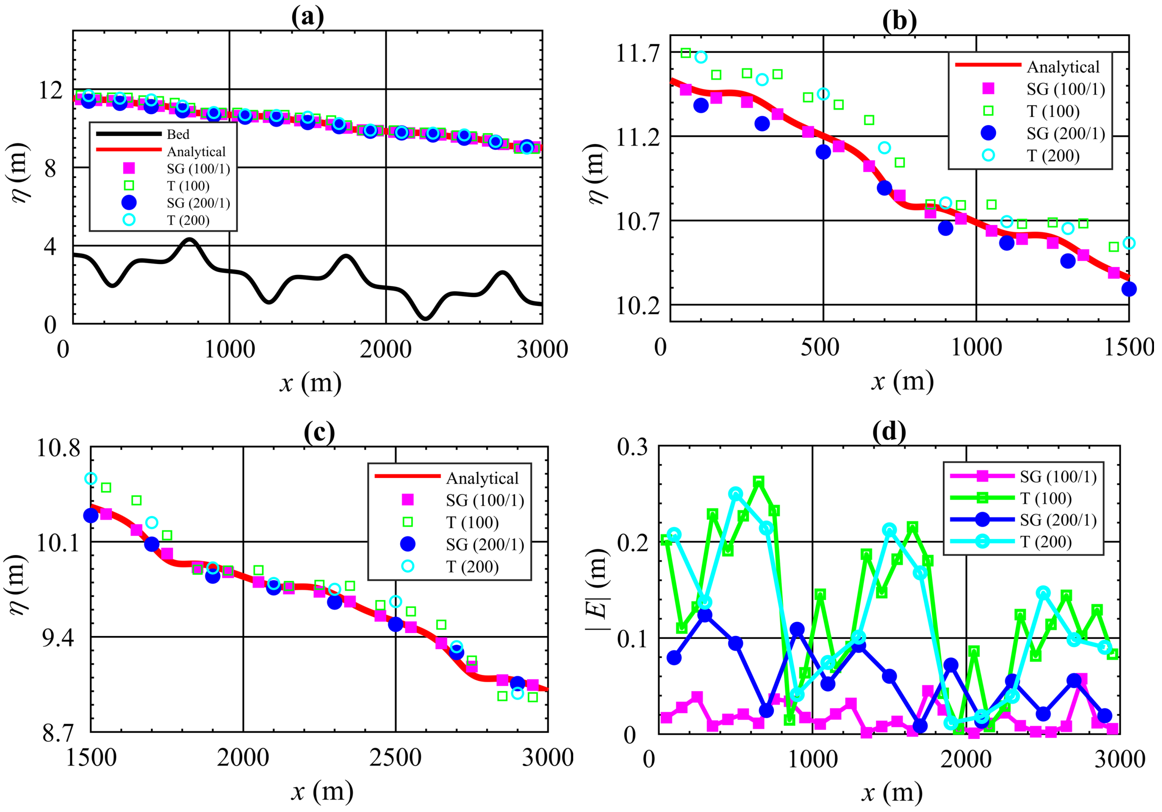


Figure 4.23: Test Case 5, flow over an irregular bed: (a) Longitudinal bed and water surface level for $0 \leq x \leq 3000$; (b) Water surface results at the upstream and (c) downstream halves of the channel; (d) Absolute error ($| E |$) between analytical and numerical solutions along the channel.

Table 4.6: Test Case 5, flow over an irregular bed: Maximum absolute and RMS errors of water level relative to the analytical solution.

Cell size	Max. absolute error (m)		RMSE (m)	
(m)	T	SG	T	SG
20	0.06	0.02	0.05	0.01
50	0.19	0.03	0.10	0.01
100	0.26	0.06	0.15	0.03
200	0.25	0.13	0.14	0.08

low-resolution meshes. In this test case, the velocity changes from 1.89 m/s to 2.77 m/s. At first glance, all numerical solutions provide suitable predictions in this test, however at the same computational resolutions, the solutions of the SG model relative to the T model are more accurate. A good example in this case is obtained from the comparison between SG (100/1) and T (100) at the location where the maximum and minimum values of velocity occur (e.g. x=250 m, and x=750 m).

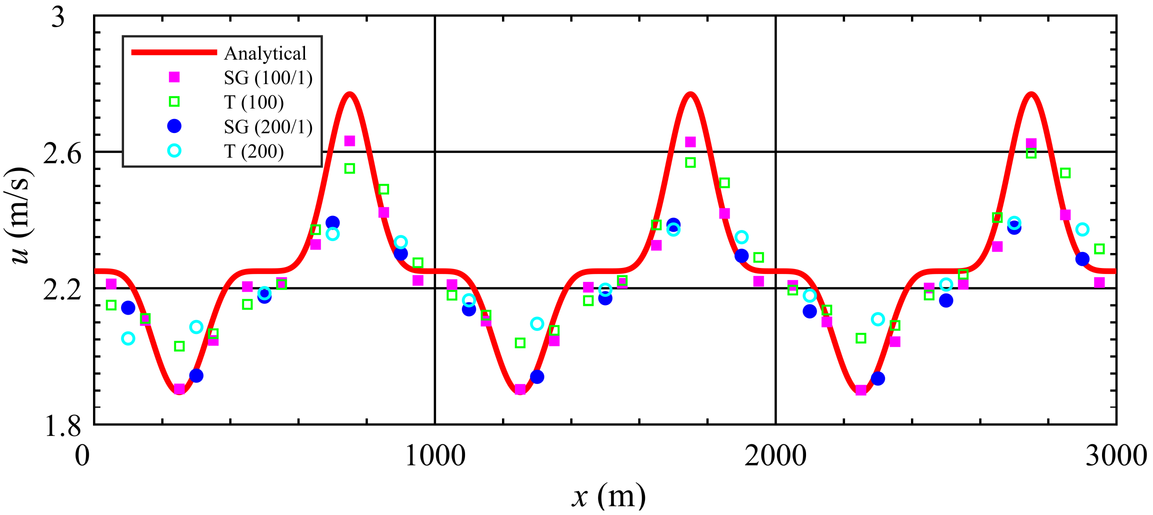


Figure 4.24: Test Case 5, flow over an irregular bed: the longitudinal profile of velocity in the analytical solution and numerical solutions performed on the coarser meshes.

For this test case, the GCI method is utilised for the SG and T model predictions of water surface level at five points. The five points are limited to $750 \leq x \leq 1750$ because outside this range, the topography (and also the solutions of water surface level) are repeated in a similar pattern (see Fig. 4.23). In the GCI analysis, the grids considered are at resolutions of 50 m, 100 m, and 200 m, which provide the grid refinement ratio r=2. Table 4.7 shows the results of the GCI study. This table illustrates that the solutions are substantially within the asymptotic range of convergence.

4.6 Test Case 6, compound channel

Test Case 6 is aimed at testing the capability of the sub-grid model in simulating the problem of flow over a bed with variations in both Manning's coefficient and topography. The domain is a 1200 m long prismatic, compound channel carrying a steady flow rate of $Q = 20$ m^3/s. The main channel and floodplains are rectangular. The base width of the main channel is 4.0 m, and the width of both floodplains is 8 m. The elevation of the main channel is 2 m below the elevation of adjacent floodplains.

Table 4.7: Test Case 5, flow over an irregular bed: GCI analysis based on the predictions of water surface level.

model	sample points		Δx_1 (m)	Δx_2 (m)	Δx_3 (m)	GCI$_{21}$	GCI$_{32}$	$\frac{GCI_{32}}{r^p GCI_{21}}$
			50	100	200			
	number	x (m)	η_1 (m)	η_2 (m)	η_3 (m)			
T	p_1	900	10.801	10.793	10.779	0.111	0.203	1.001
	p_2	1100	10.712	10.737	10.793	0.244	0.537	0.998
	p_3	1300	10.698	10.686	10.658	0.125	0.272	1.001
	p_4	1500	10.425	10.480	10.600	0.547	1.195	0.995
	p_5	1700	10.211	10.217	10.231	0.065	0.143	0.999
SG	p_1	900	10.752	10.729	10.654	0.119	0.388	1.002
	p_2	1100	10.618	10.614	10.596	0.011	0.054	1.000
	p_3	1300	10.555	10.530	10.459	0.165	0.464	1.002
	p_4	1500	10.356	10.346	10.293	0.027	0.147	1.001
	p_5	1700	10.096	10.099	10.105	0.069	0.111	1.000

The constant bed slope is 0.001. Two roughness scenarios are modelled, (1) using Manning coefficient values of 0.04 s/m$^{\frac{1}{3}}$ and 0.08 s/m$^{\frac{1}{3}}$ for main channel and floodplains, respectively; and (2) using the space-averaged value of scenario 1 for the whole domain, i.e. constant n_M=0.072 s/m$^{\frac{1}{3}}$ for the both main channel and floodplain. A constant value of η=2.5 m is imposed as downstream boundary condition at x=1200 m. The model is run from an arbitrary initial condition for a duration sufficiently long for steady flow to be reached. The small cell size is 2 m and the resolutions for the both Traditional and Sub-grid models (coarse mesh) are 2 m, 4 m, and 20 m. The solution of T (2) is taken as the benchmark solution, against which all other solutions are compared.

Fig. 4.25 shows the water surface level at the channel centreline obtained with the two models at different resolutions and both Manning coefficient scenarios. In these figures, no substantial differences can be observed between the free surface profiles predicted by the SG (4/2) and T (4), which also coincide (visually) with the benchmark solution T (2). On the other hand, at the coarsest resolution, T (20) and SG (20/2) results in over- and underestimated solutions, respectively for the both scenarios. The results of SG (20/2) are much closer to the benchmark solution than those of T (20).

Fig. 4.25 shows that differences between the coarse and benchmark solutions are higher in scenario

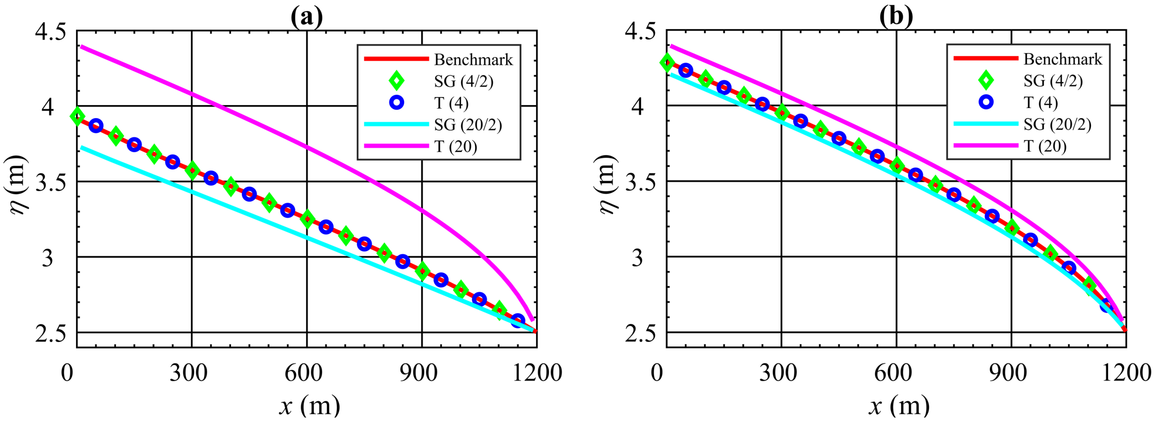


Figure 4.25: Test Case 6, compound channel: longitudinal profile of water surface solutions at the centreline of the channel for (a) Scenario 1 and (b) Scenario 2.

1 (varied roughness) than in scenario 2 (constant roughness). This shows that simple averaging of roughness coefficients across a computational cell may produce results that substantially overestimate the effect of friction. Table 4.8 compares the maximum absolute ($| E |$) and RMS errors (i.e. relative to T (2) simulation) of the two models run at computational cell resolutions of 4 m and 20 m. At 4 m resolution (in which case both the topographic and roughness data of cells in the main channel and floodplain are the same as those in of the finest resolution grid), the results of both models are very close to the benchmark solutions. At the coarsest resolution (i.e. Δx $=$ 20 m), where each computational cell covers parts of both the floodplain and main channel, (i.e. there are large variations in Manning roughness and bed geometry inside a computational cell), errors obtained with the T model are substantially higher than those provided by the SG model. For example, the RMSE of T (20) compared to SG (20/2) is 0.44 m vs 0.13 m in the first and 0.16 m vs 0.06 m in the second scenario, respectively. This shows that SG (20/2) relative to T (20) reduces the RMSE by a factor more than two.

Fig. 4.26 shows the longitudinal profile of velocity along the centreline of the main channel for all numerical solutions. Identical to the results provided by Fig. 4.25, in both the scenarios, the solution on SG (4/2) and T (4) are very close to the benchmark solutions, and by coarsening the computational meshes, the solutions (for the both models) are differentiated from the benchmark solutions. However, the solution of SG (20/2) is more accurate than the solution of T (20). This is particularly the case for the first scenario, which involves the distribution of the roughness coefficient across the domain of study. According to Fig. 4.26, the range of the values of the velocity is more limited in the second scenario, compared to Scenario 1 (except from T (20), which represents the same solution in both

Table 4.8: Test Case 6, compound channel:Maximum absolute and RMS errors of water level relative to the benchmark solution [T (2)] for Scenarios 1 and 2.

Scenario number	Cell size	$\mid E \mid$ (m)		RMSE (m)	
		Traditional	Sub-grid	Traditional	Sub-grid
1	4	0.08	0.007	0.002	0.003
	20	0.51	0.21	0.44	0.13
2	4	0.005	0.005	0.003	0.004
	20	0.17	0.07	0.16	0.06

scenarios because the topography and roughness are unchanged during the two types of simulations). This would be already expected, as Fig. 4.25 presents higher η, which also means higher h, for the second scenario, whereas the discharge is the same in both scenarios.

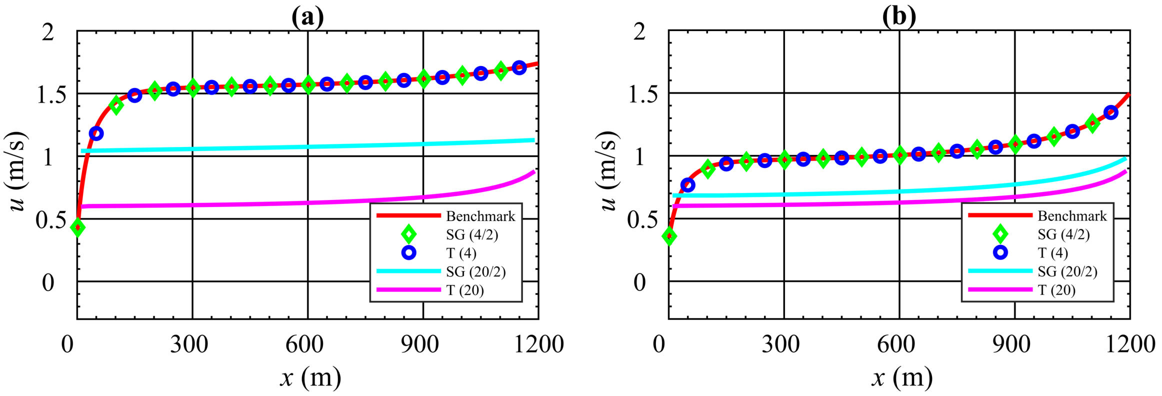


Figure 4.26: Test Case 6, compound channel: the longitudinal profile of velocity along the centreline for (a) Scenario 1 and (b) Scenario 2.

4.7 Test Case 7, the River Tiber

Test Case 7 evaluates the performance of the SG model in simulating a real-world flood inundation problem, where the flow is unsteady and two-dimensional. This test case simulates a 113 hour-long flood event that occurred between 27th November and 1st December 2005 in the River Tiber, Italy, (Morales-Hernández et al., 2016). The River Tiber is the third largest river in Italy flowing between Apennine Mountains and Tyrrhenian Sea. The catchment area and the length of river are approximately 17,000 km^2 and 400 km, respectively. The average flow in the river is 267 m^3/s, while the peak flow of the 200-year return period event is approximately 3,200 m^3/s (Morales-Hernández et al., 2016). The computational domain of study is a 6 km × 2 km area near the city of Rome, which is shown in Fig. 4.27. This simulation adopts the values of Manning's coefficient previously used by Morales-Hernández et al. (2016), where the values for the main channel and floodplains were defined as 0.035 m/s$^{\frac{1}{3}}$ and 0.0446 m/s$^{\frac{1}{3}}$, respectively.

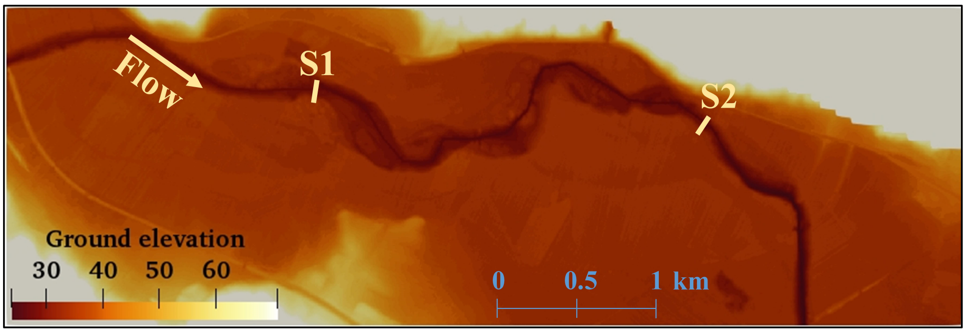


Figure 4.27: Test Case 7, the River Tiber: Plan view of the computational domain.

The initial condition corresponds to the results of a steady flow simulation with discharge of 374 m^3/s. Fig. 4.28 shows the time series of discharge (Q) and water surface (η) used as the upstream and downstream boundary conditions. Water surface levels have been measured during the flood event at cross sections S1 and S2 shown in Fig. 4.27. Gridded topographic data are available at resolutions of 2 m and 5 m. In the proposed sub-grid model, these two resolutions are taken as the small cell resolution, which are combined with different sizes (10 m, 20 m, 40 m, 50 m, and 100 m) of large cells. Cell sizes in the T model are the same as large cell sizes in the SG model except that one more resolution (4 m) is employed. The results of T (4) are adopted here as the benchmark solution.

Fig. 4.29a and 4.29b show time series of free surface elevation at cross sections S1 and S2, respectively, predicted by the two models at different resolutions, along with the field data available. It must be noted that the field measurements are included here for illustrative purposes only, as the aim of the analysis is to compare the results of the SG model against the benchmark (in this case, a high-resolution T model). The performance of the models relative to the field data available depend on several factors related to the uncertainty in the measured data, boundary conditions and model parameters. Such comparison would therefore not serve the purpose of assessing errors introduced by coarsening the grid, and is therefore out of the scope of this work. It was not possible to perform the simulation T (100). This is because coarsening of the cells near the downstream boundary produced cell elevations that were higher than the water surface elevation set at the boundary of the domain.

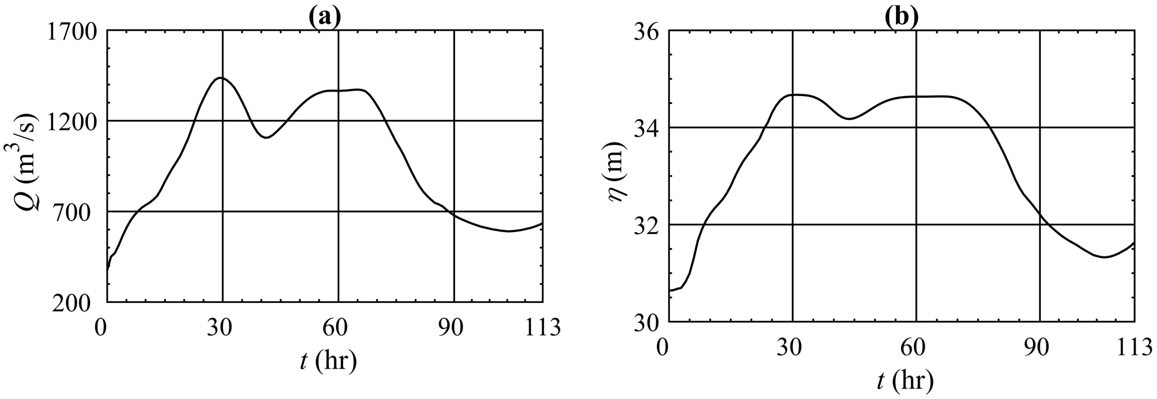


Figure 4.28: Test Case 7, the River Tiber: upstream and downstream boundary conditions. (a) Upstream (western boundary) time series $Q(t)$ and (b) downstream (southern boundary) time series $\eta(t)$.

Since the sub-grid model includes the elevation of small cells near the boundaries, even the coarsest resolution simulation was performed successfully. For clarity, Fig. 4.29 shows results for selected simulations. The predicted elevations are invariably higher than the benchmark solution at both cross-sections, and this difference reaches a maximum near the end of the simulation. The improved accuracy of the sub-grid model becomes clear on the coarse resolution grids, where the maximum error (relative to the benchmark water surface) is reduced from about 1.3 m and 2 m [T (50)] to 0.55 m and 0.65 [SG (50/2)] at S1 and S2, respectively. No significant differences were observed between the results of the SG model at small cell resolutions of 2 and 5 m.

Fig. 4.30 shows the velocity magnitude time history at the deepest points of cross-sections S1 and S2. The numerical models at the adopted mesh resolutions provide underestimated solutions compared to the benchmark solution. The results illustrate that the predictions of model/resolutions are relatively close together. The time series exhibit different patterns at cross-sections S1 and S2. At cross-section S1, similar to curves provided in Fig. 4.28 and 4.29, there are two peaks by t=30 hr and t=60 hr, whereas the pattern at cross-section S2 is closer to uniform. This may be because the main channel cross-section widens between cross-section S1 and S2, as evident in Fig. 4.27 comparing the situation upstream of S1 to that downstream of S2, thus increasing the local storage capacity regulating the flow rate passing S2.

Fig. 4.31 compares the performance of the two models in terms of accuracy and runtime (t_c). The accuracy of the simulations is assessed based on the RMSE of water depth of the solutions relative to the benchmark [T (4)]. Fig. 4.31a represents the solutions at t= 110 hr, whereas Fig. 4.31b

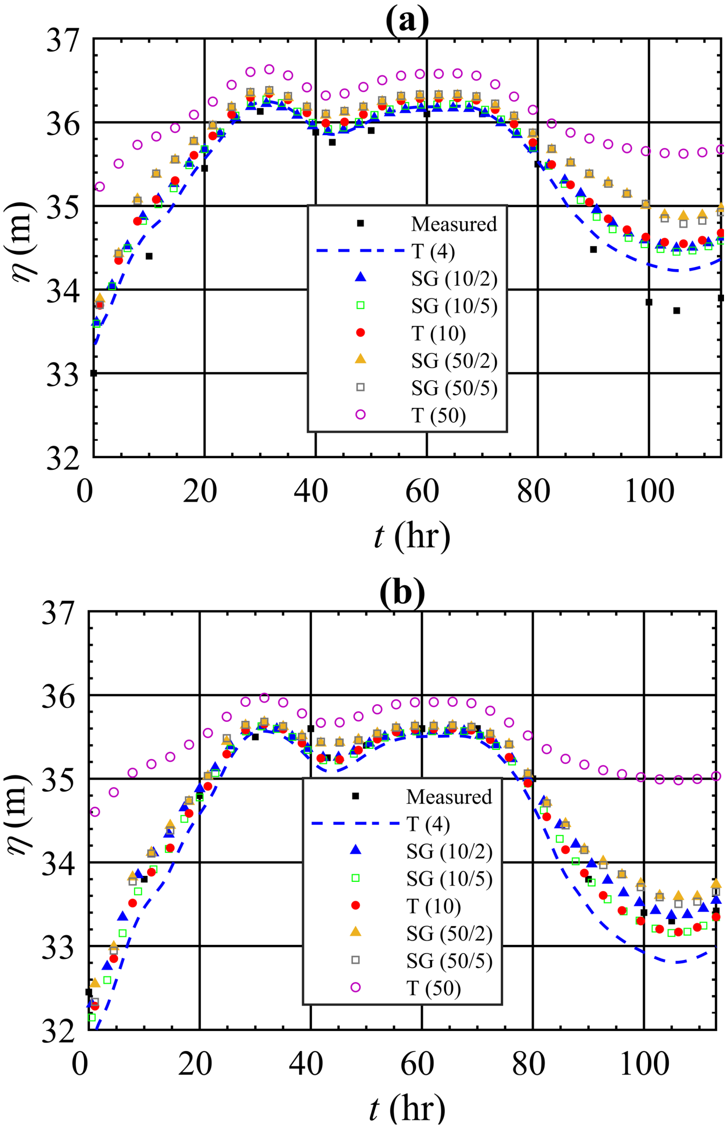


Figure 4.29: Test Case 7, the River Tiber: measured and predicted water surface elevations at cross-sections (a) S1 and (b) S2.

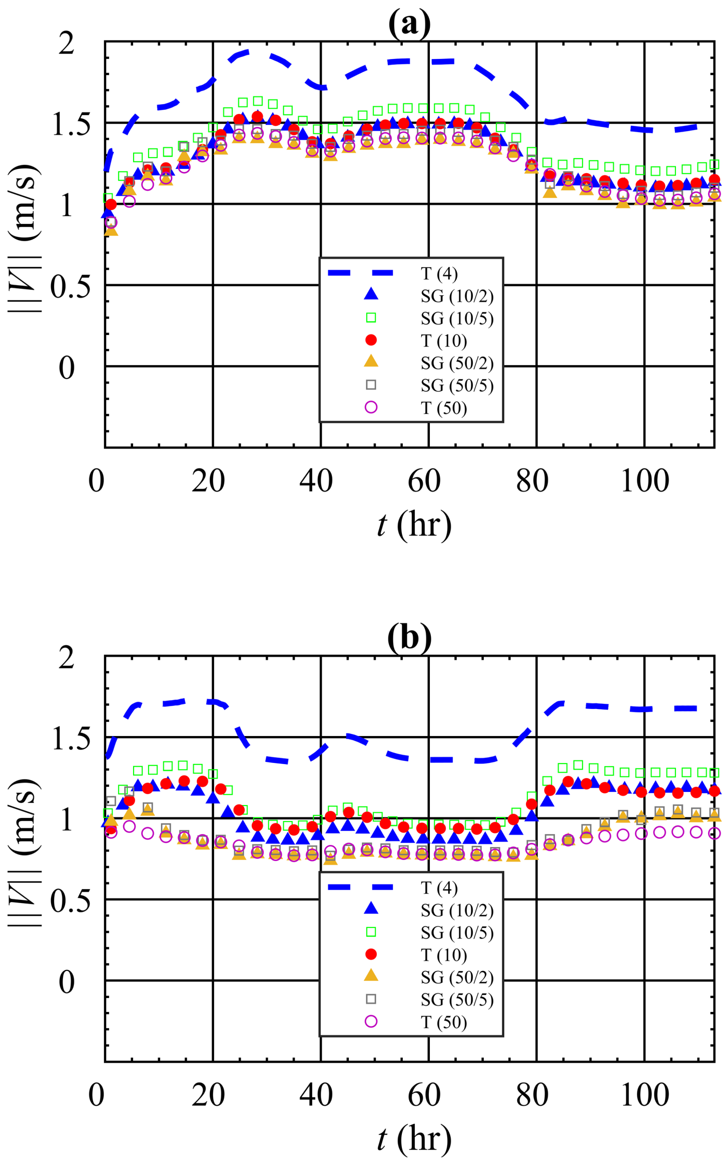


Figure 4.30: Test Case 7, the River Tiber: the predicted velocity magnitude at cross-sections (a) S1 and (b) S2.

corresponds to errors at the maximum predicted water depth at each of the cells during 113 hours of flood event. In Fig. 4.31b, the maximum water surface level computed at each computational cell is first determined for the each model/resolution. The corresponding maximum water depth is then obtained by subtracting the bed elevation at 2 m resolution from the maximum water surface levels. In Fig. 4.31a the process is similar but instead of maximum water surface level, water surface levels at t = 110 hr are adopted. The results in Fig. 4.31a and 4.31b show the improved accuracy provided by the SG model compared to the single mesh approach (i.e. T model) at different resolutions. At resolutions of the computational grid smaller than 20 m, the accuracy of the T and SG models is close and displays a similar trend. However, as the size of the computational cell is coarsened, the curves diverge substantially, with the SG displaying a much milder increase in error than the traditional approach. The analysis shows that the slope of T (. . .) curve in log (RMSE) vs log (Δx) is about 1 (exactly 0.95 and 0.9 for the cases Fig. 4.31a and 4.31b respectively), which would be expected for a first order model. By contrast, the slopes of the curves for SG (. . . /2) and SG (. . . /5) are not constant and always less than 1 (i.e. errors increase at a much slower pace as the grid is coarsened).

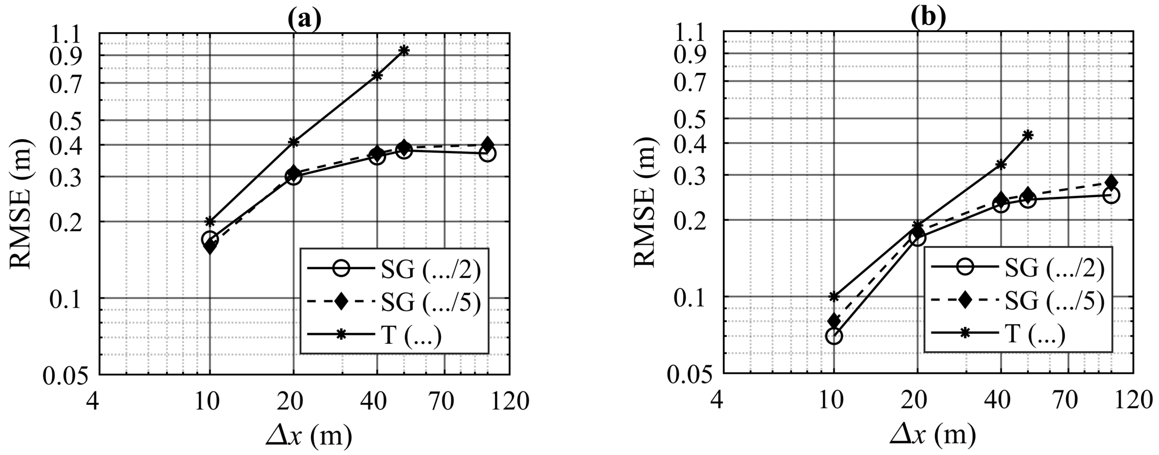


Figure 4.31: Test Case 7, the River Tiber: water depth RMSE of the models/resolutions relative to the benchmark solution against computational cell size (Δx) where, the water depth is: (a) that at t = 110 hr, and (b) the maximum computed during the flood event.

Fig. 4.32a shows the runtime for different simulations performed with the two models and for different computational cell sizes. As previously discussed, the computational cost follows a power law $t_c \sim \Delta x^{-p}$, where p=3 for traditional explicit models. Fig. 4.32a confirms this for the T model (p =3.013) and also shows how p is reduced for the SG model [p=2.216 and 2.294 for SG (. . . /2) and SG (. . . /5), respectively]. Values of p less than 3 for the SG model occur mainly because because the fluxes at the cell edges are computed at small cell resolution. To obtain the fluxes in a square cell, the traditional model solves the Riemann problem, which is an expensive part of the solution, four times, whereas for the SG model, it is executed as many times as there are small cells next to the large cell edges [e.g. 40 times for SG (50/5)]. Thus, the time cost of the SG relative to T model is higher, resulting in lower p (i.e. p<3).

Fig. 4.32b displays RMSE at t = 110 hr, when the differences of η predicted by the benchmark solution and other solutions are relatively large (also previously shown in Fig. 4.29), as a function of model runtime, obtained from the simulations with the two models at different resolutions. The figure shows that as the simulation time is reduced (i.e. grid coarsening), the accuracy gap between the two models gradually increases, with the SG model outperforming the single mesh model. For instance, Fig. 4.32b shows that for runtimes less than t_c= 5 hr, the SG model provides the most accurate results.

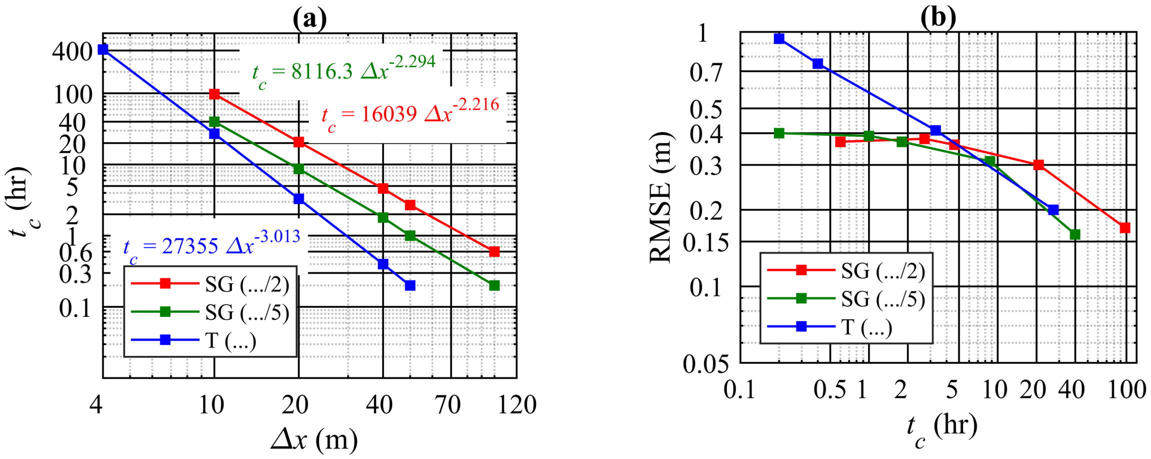
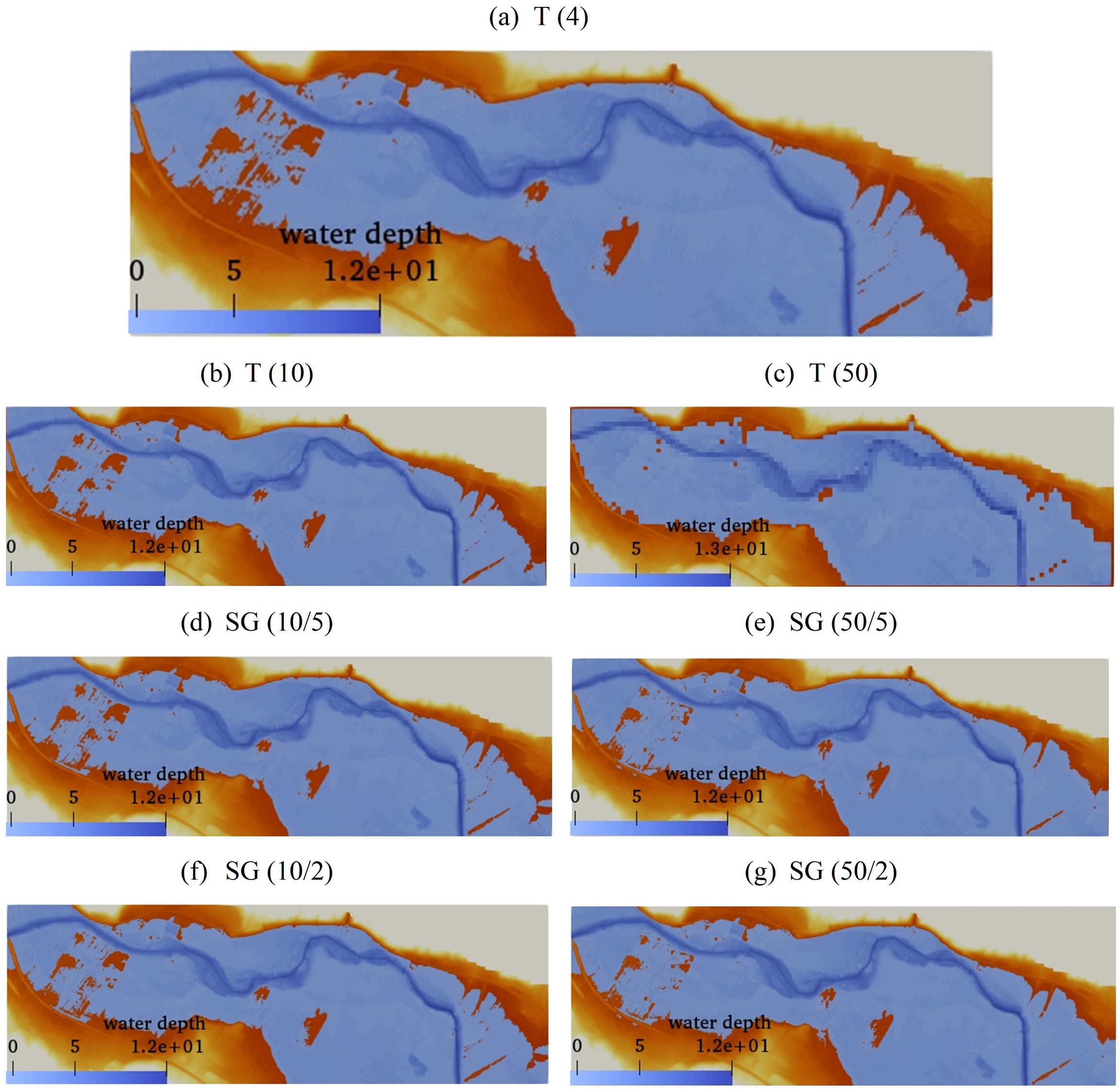


Figure 4.32: Test Case 7, the River Tiber: (a) runtime (t_c) against computational cell size of the models/resolutions. (b) Water depth RMSE (at t = 110 hr) of the models/resolutions relative to the benchmark solution as a function of runtime.

Figs. 4.33 to 4.35 show the flooded areas predicted by the two models at t=33, 80 and 113 hr. For clarity, only the finest (Δx=10 m), one of the lowest (Δx=50 m) and the benchmark (Δx=4 m) resolution results were depicted. As a general observation, the flooded areas predicted by T (10), SG (10/2), SG (10/5), SG (50/2), SG (50/5), are close to the benchmark solution and only T (50) provides results that are clearly (visually) different. This can be analysed through the inspections of Fig. 4.36, where time series of the flooded area are plotted for the T and SG models at the mentioned cell resolutions. All the models/resolutions predicted the flooded area well compared to the benchmark solution in the time range of t =20 hr to t =80 hr, when the peak flow causes the maximum water surface level to occur within the domain (see Figs. 4.28 and 4.30). During the first 20 hours, the results can be classified into three categories. The results produced by T (10), SG (10/2) and SG (10/5), which have the highest resolution, are in good agreement with those by T (4). The results of T (50), SG (50/2), SG (50/5), SG (100/2) and SG (100/5), overestimate the flooded area, although SG (100/2) and SG (100/5) are clearly more accurate when compared to T (50). During the last 30 hours of the flood event, even the results of the SG model at coarse resolution (i.e. 100/5 and 100/2) produce a relatively good estimate, whereas the twice finer T (50) model gives much poorer results.


(a) T (4)

(b) T (10)

(c) T (50)

(d) SG (10/5)

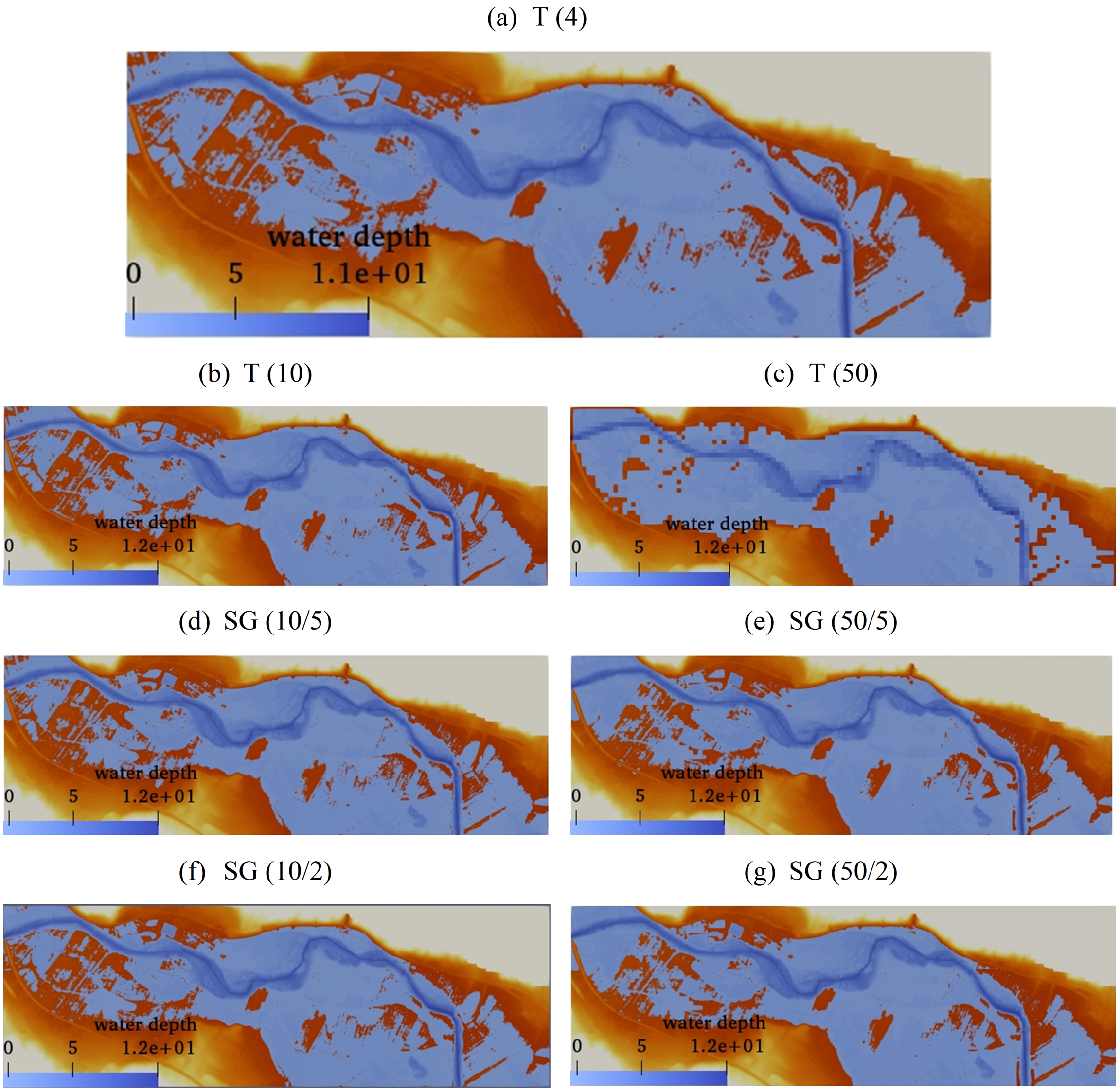
(e) SG (50/5)

(f) SG (10/2)

(g) SG (50/2)

Figure 4.33: Test Case 7, the River Tiber: maps of flooded area at t= 33 hr for the different models/resolutions.

Fig. 4.37 displays the variation in storage as a function of time. On fine resolution meshes, the numerical solutions (in both models) are very close to the benchmark solution [i.e. T (4)] and by



(a) T (4)

(b) T (10)

(c) T (50)

(d) SG (10/5)

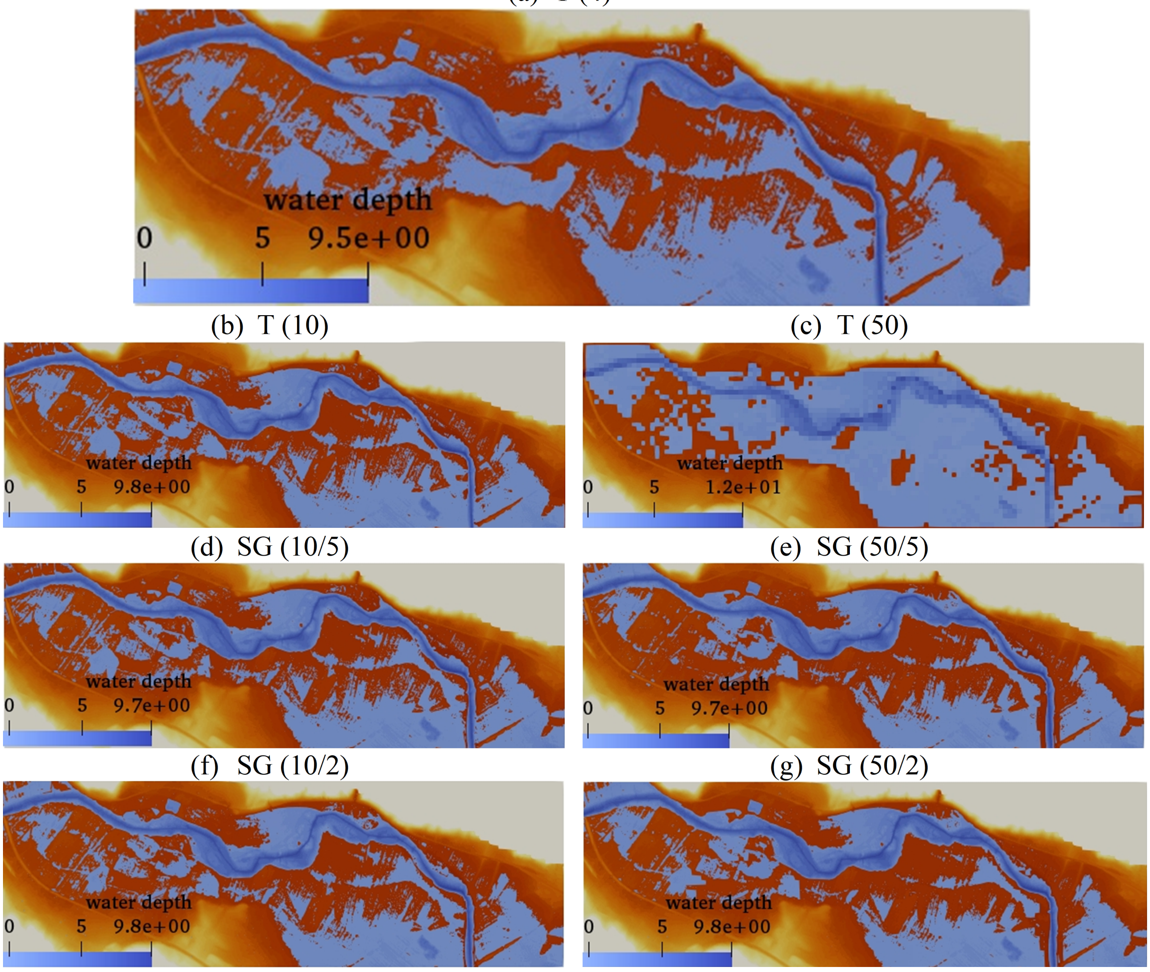
(e) SG (50/5)

(f) SG (10/2)

(g) SG (50/2)

Figure 4.34: Test Case 7, the River Tiber: maps of flooded area at t= 80 hr for the different models/resolutions.

(a) T (4)



(b) T (10)

(c) T (50)

(d) SG (10/5)

(e) SG (50/5)

(f) SG (10/2)

(g) SG (50/2)

Figure 4.35: Test Case 7, the River Tiber: maps of flooded area at t= 133 hr for the different models/resolutions.

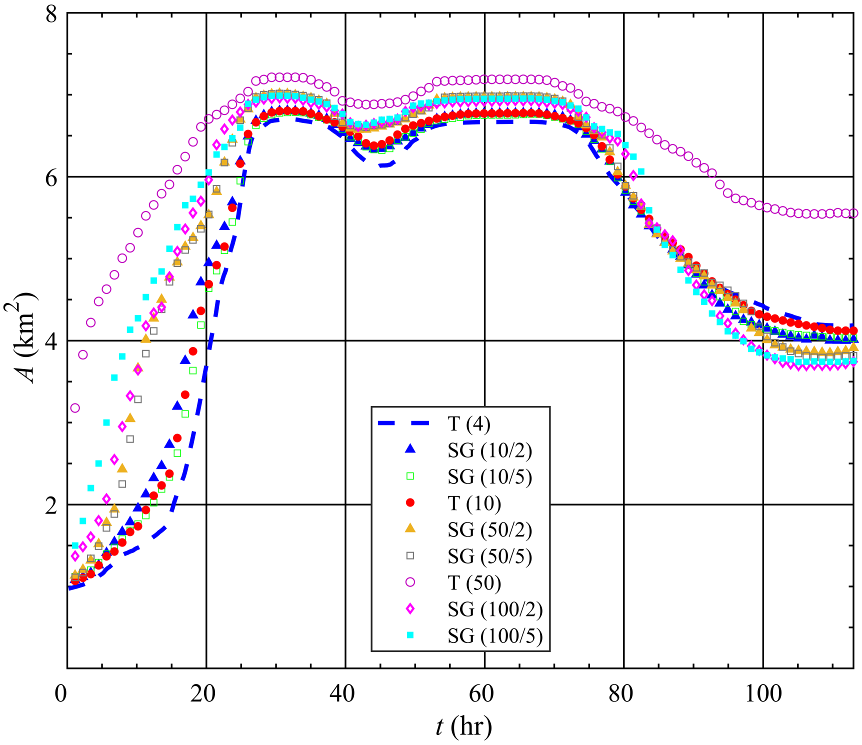


Figure 4.36: Test Case 7, the River Tiber: time series of flooded area (A) obtained by the different models/resolutions.

coarsening the meshes, the difference between the solutions of models and the benchmark solution increases. However, the SG model presents more accurate results on low-resolution meshes. A good example in this case can be obtained from comparing the solutions SG (50/2) and SG (50/5) to T (50), where T (50) gives significant overestimation relative to T (4) (particularly at the beginning and end of the flood event).

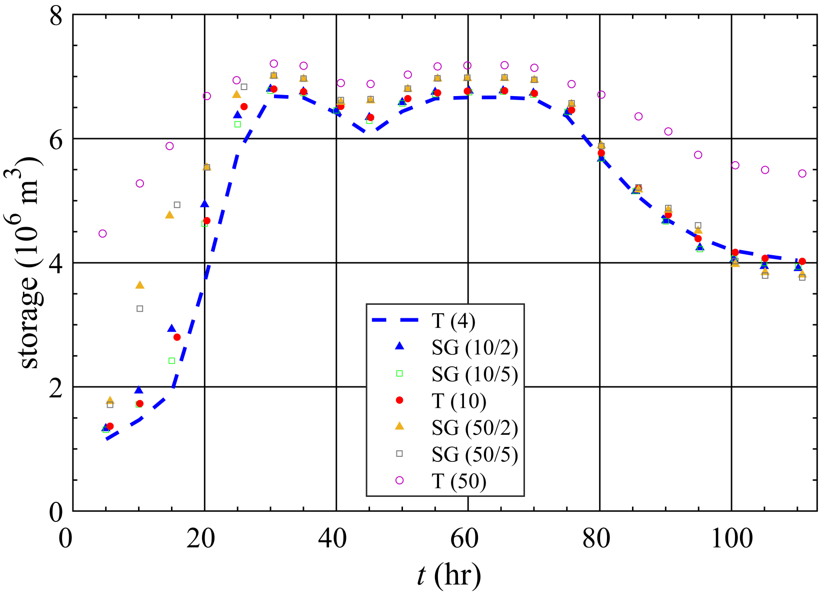


Figure 4.37: Test Case 7, the River Tiber: time series of storage obtained by the different models/resolutions.

For this test case, the GCI analysis is carried out using the water surface elevation solution at five points across the domain of study. Fig. 4.38 shows the position of the points on the map of the domain. The GCI analysis is conducted at three time instants, which relate to before, within and after the peak of the flood event, respectively (see Fig. 4.28). This analysis is proceeded for both the T and SG models. Table 4.9 lists the values of GCI obtained from the aforementioned points, models and times. The results confirm that the calculations converge asymptotically to the correct solutions on the grids adopted in this study.

Table 4.9: Test Case 7, the River Tiber: GCI analysis based on the predictions of water surface level.

			Δx_1 (m)	Δx_2 (m)	Δx_3 (m)			
model	t (hr)	sample points	10	20	40	GCI_{21}	GCI_{32}	$\frac{\mathrm{GCI}_{32}}{r^p \mathrm{GCI}_{21}}$
			η_1 (m)	η_2 (m)	η_3 (m)			
T	15	p_1	35.915	36.146	36.479	1.819	2.607	0.994
		p_2	35.417	35.539	35.749	0.587	1.012	0.997
		p_3	35.002	35.268	35.556	11.272	12.130	0.992
		p_4	34.173	34.319	35.556	0.071	0.601	0.996
		p_5	32.910	33.098	34.893	0.083	0.791	0.994
	60	p_1	37.012	37.106	37.215	2.012	2.323	0.997
		p_2	36.284	36.353	36.439	1.015	1.251	0.998
		p_3	36.023	36.106	36.210	1.122	1.406	0.998
		p_4	35.594	35.603	35.741	0.002	0.035	1.000
		p_5	34.980	34.990	35.042	0.010	0.049	1.000
	90	p_1	35.600	35.881	36.242	3.452	4.403	0.992
		p_2	35.045	35.290	35.605	2.988	3.833	0.993
		p_3	34.744	35.082	35.431	32.990	33.874	0.990
		p_4	33.872	34.097	34.747	0.436	1.255	0.993
		p_5	32.483	32.493	32.624	0.003	0.042	1.000
SG $(\cdots/2)$	15	p_1	36.210	36.391	36.598	4.262	4.862	0.995
		p_2	35.321	35.482	35.654	8.665	9.193	0.995
		p_3	35.083	35.225	35.370	28.738	29.126	0.996
		p_4	34.405	34.500	34.662	0.489	0.832	0.997
		p_5	32.956	33.074	33.213	2.530	2.967	0.996
	60	p_1	37.095	37.178	37.283	1.062	1.339	0.998
		p_2	36.165	36.235	36.320	1.159	1.399	0.998
		p_3	35.939	35.969	36.020	0.146	0.250	0.999
		p_4	35.569	35.601	35.637	1.300	1.413	0.999
		p_5	34.983	34.987	35.010	0.004	0.021	1.000
	90	p_1	35.839	35.913	36.142	0.125	0.383	0.998
		p_2	34.956	35.158	35.420	2.483	3.188	0.994
		p_3	34.765	34.916	35.141	1.092	1.627	0.996
		p_4	34.031	34.177	34.355	2.448	2.971	0.996
		p_5	32.406	32.407	32.536	0.000	0.002	1.000

model	t (hr)	sample points	Δx_1 (m)	Δx_2 (m)	Δx_3 (m)	GCI$_{21}$	GCI$_{32}$	$\frac{\text{GCI}_{32}}{r^p \text{GCI}_{21}}$
			10	20	40			
			η_1 (m)	η_2 (m)	η_3 (m)			
SG $(\cdots/5)$	15	p$_1$	36.046	36.218	36.542	0.668	1.257	0.995
		p$_2$	35.265	35.403	35.622	0.833	1.317	0.996
		p$_3$	34.996	35.106	35.337	0.356	0.746	0.997
		p$_4$	34.227	34.351	34.525	1.138	1.586	0.996
		p$_5$	32.957	32.930	32.889	0.196	0.299	1.001
	60	p$_1$	37.050	37.128	37.296	0.228	0.490	0.998
		p$_2$	36.211	36.263	36.324	1.041	1.218	0.999
		p$_3$	35.971	36.003	36.038	0.782	0.889	0.999
		p$_4$	35.566	35.597	35.648	0.174	0.282	0.999
		p$_5$	34.980	34.987	35.008	0.010	0.033	1.000
	90	p$_1$	35.610	35.828	36.051	35.559	36.104	0.994
		p$_2$	34.906	35.136	35.371	40.306	40.860	0.993
		p$_3$	34.634	34.860	35.103	11.435	12.173	0.993
		p$_4$	33.809	33.908	34.176	0.217	0.583	0.997
		p$_5$	32.399	32.417	32.485	0.027	0.098	0.999

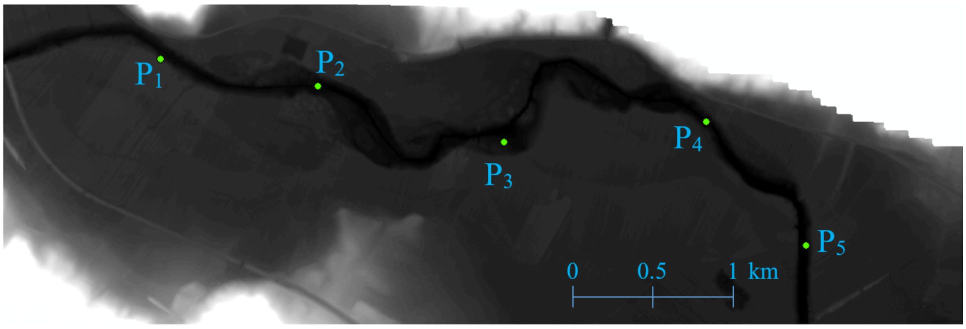


Figure 4.38: Test Case 7, the River Tiber: plan view of the points adopted for the GCI analysis.

4.8 Concluding remarks

In this chapter, seven test cases were studied. In the first test case, a 1D steady flow over a smoothed hump is investigated. The hump provides the transcritical condition and appearance of a hydraulic jump in the channel, which is frictionless. Through the second test case, the 1D dam-break problem is studied for two scenarios; when the downstream part of the channel is dry or wet (still water). This test provides an unsteady flood propagation problem, the analytical solution of which is available. Test Case 3 involves the 2D dam-break problem over a flat rectangular domain containing three conical humps. This widely studied test exams the performance of the proposed model in capturing waves and also wetting and drying conditions. Test Case 4 simulates a steady flow in a sharp 90° channel bend. In this test, the computational cells are much coarser than the width of a rectangular channel that changes direction by 90°. The test was run for two scenarios of bed level in floodplains, and based on the benchmark solution and simple physical considerations (i.e. the flow is fully contained in the main channel), the numerical results should be independent of the floodplain elevation. In Test Case 5, the accuracy of the model is tested for flow in a 1D steady channel that has relatively large and complex variations in the bed elevation. Test Case 6 corresponds to a prismatic compound channel with rectangular floodplain and main channel, where different bed levels and Manning coefficients of floodplain and main channel were adopted to test the proposed upscaling methods. Two scenarios were compared where spatially-distributed and the spatially-averaged values of these parameters are used for each computational cell. The last test case exams the performance of the proposed sub-grid model for a real-world flood inundation event that occurred in the River Tiber, Italy between 27th November and 1st December 2005. The flood was simulated by the proposed and a traditional finite volume model with different mesh resolutions and the results were compared against the benchmark solution and recorded field data.

The analysis of the results of these comparisons allow us to conclude that:

- A SG model has been developed for large scale simulations using low-resolution computational meshes. Although it is not the aim of the present model to capture (standing or moving) discontinuities in the water surface, the SG model is nevertheless able accurately to capture such phenomena on high resolution meshes. Results from the first three test cases (i.e. 1D steady flow over a hump, 1D dam-break problem, and dam-break wave propagating over three humps) indicate that when the topography is complex, the SG model provides more accurate shock capturing compared to the T model at the same mesh resolution.
- Simple grid coarsening (i.e. traditional model) may lead to inaccurate solutions, as evidenced by

Test Case 4 (open channel flow around a bend), where the topography of the dry part of the study domain in a traditional model has an important impact on the numerical solution. In this test, the results obtained with the SG model were considerably more accurate than those provided by the T model. This sheds light onto the limitations of some of the recent modelling techniques previously discussed (e.g. Hénonin et al., 2015), which adopt overly simplified methods to upscale the fine resolution information.

- Overall, at any given resolution, the SG model provides results that are more accurate than those obtained from the single mesh approach (i.e. T model) and this improved accuracy is more obvious on low-resolution meshes.
- Test Case 6 (compound channel) shows that the use of simple averaging of roughness coefficients and bed level for a computational cell (i.e. a traditional model) may affect significantly solution accuracy. This again sheds light onto an issue that has not been reported in previous investigations on the development of sub-grid models that are strongly based on simple averaging (Sanders et al., 2008; Soares-Frazão et al., 2008; Guinot et al., 2017). The inclusion of finely resolved, spatially distributed values of parameters in the SG model proved to be a successful strategy to improve the accuracy of simulations performed at coarse resolution.
- Inclusion of finely resolved data within a coarse resolution grid has enabled the SG to successfully perform simulations when these cannot be obtained by the T model. An example is Test Case 7 (the River Tiber), where T (100) is unable to run the simulation, because of the inaccurate reconstructions of bed level and hydraulic properties, which arise from simple grid coarsening, near the downstream boundary of the domain. On the other hand, the SG (100/2) and SG (100/5) simulations were performed without encountering any problems. It is also expected that the traditional method may be unable to perform certain simulations at coarse resolution using current-generation porosity-based sub-grid models, where use of high-resolution data is limited to porosity coefficients.
- The SG performance was assessed for the real-world Test Case 7 (the River Tiber). The results showed that on high-resolution meshes, it is more efficient to use the traditional model but as the mesh is coarsened, the SG model outperforms the traditional approach.

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

In this work, a new model has been presented for the solution of the two-dimensional shallow water equations based on a Godunov-type finite volume method modified to achieve high computational performance. Central to the model's improved performance is the dual nested mesh approach. In this method, a coarse mesh is used as the computational mesh, which contains large cells defining the sub-domains at which the conservation laws are applied. A fine resolution mesh is used to include finely resolved information (i.e. terrain elevation and roughness) in the solution and therefore improve accuracy. Large cells are allowed to be fully or partially wet, with the wet area and other variables updated dynamically.

New methods are presented to upscale the solution from fine to coarse meshes as follows. To compute the mass and momentum exchanges between two neighbouring large cells, a new method is presented to reconstruct the conservative variables at the edges of the high-resolution mesh, where the Riemann problem is solved. The method was designed so that most of the additional computational time required to reconstruct variables is performed at pre-processing only. The bed slope term is upscaled from the fine to coarse mesh according to the fine scale topographic data and the value of the free surface elevation η along the edges of large cells. The method accounts for both fully and partially wet cells. A new method is presented for upscaling the friction source terms, which assumes a generic distribution of momentum within a large cell under the assumption that the flow is approximately parallel. This assumption leads to a method whereby expensive computations involving fine scale information are also performed at pre-processing only.

A number of test cases were studied in this research, where the results of the sub-grid (SG) model were compared against those from a traditional first order finite-volume method (T). The study showed that at the same grid resolution, the SG model provides improved accuracy over traditional methods. This enhanced accuracy is most obvious at low resolution. The results also showed that the model is capable of delivering higher accuracy through the inclusion of sub-grid scale finely resolved topography and roughness. It has also been shown that, by accounting for sub-grid scale information, the SG model enables simulations to be performed at resolutions that are too coarse for traditional methods to even start the computations. Combined analysis of accuracy and computational time suggests that at coarse resolutions and when similar levels of accuracy are considered, the SG model can be substantially faster than currently available methods. While a more extensive set of real-world tests may be needed to fully understand the performance of the model (i.e. as compared to other existing techniques), these results, show a promising path to developing high-performance inundation models. Finally, the results of the simulations performed in this thesis have shed light onto some of the unreported limitations of other sub-grid techniques that have been proposed in the past years.

5.2 Model limitations

While certain strengths of the SG model were reviewed in the previous sections, some shortcomings and inherent limitations of such modelling strategy must be discussed. First, the SG model presented in this research implements a first order accurate scheme for the solution of the homogenous part of the 2D shallow water equations. This assumes that changes in the values of conserved variables are small and can be approximated as piecewise constant. Such an assumption will induce loss of accuracy under certain scenarios (e.g., steep water surface profiles), specially at coarse resolution. A second order version of the proposed method is therefore likely to further improve the accuracy of coarse resolution simulations under these conditions. It is also important to notice that in the type of upscaling techniques proposed in this work (including the potential future extension to second order), the function describing the water surface profile inside a large computational cell is assumed *a priori*. This means that localised changes in the conserved variables (e.g. a hydraulic jump) that may occur within a large cell cannot be captured by such models.

5.3 Future Work

During the development of the model proposed in this thesis, a number of observations have been made that can be valuable to guide further developments in this field. Further work on the topics described within the next sections have the potential to improve model accuracy and performance, paving the way for many applications that are now hindered by the lack of models of high computational efficiency.

5.3.1 Second order SG model

According to Begnudelli et al. (2008), the combination of a first order accurate scheme for the solution of homogenous shallow water equations and a second order topography (i.e. piecewise linear change in bed elevation) model has higher efficiency, accuracy and robustness, compared to a second order scheme (see Section 3.3.1). In the proposed sub-grid model, topography is presented through small cells (high-resolution mesh), which provides a step forward towards improving the accuracy of the representation of topography at coarse resolution grids. However, whereas the observation by Begnudelli et al. (2008) may correctly represent certain scenarios of high- to medium- resolution, as computational grids are made substantially coarser, it is reasonable to assume that the first-order approximation eventually limits the overall accuracy of the model. Additional simulations performed as part of the present study (but not included in the thesis) indicates that there are cases in which the second order SG model improves the accuracy of the model substantially. This is because the conservative variables are reconstructed more accurately at the edges when solving the Riemann problem. Combining the second order model with the methods proposed in this thesis will therefore lead to a model that is capable of performing simulations at even coarser resolutions and with limited loss of accuracy.

5.3.2 Combination of the model with parallel programming approach

In Chapter 1, several approaches that have been typically adopted to speedup large-scale flooding simulations were reviewed. The structure of the proposed model is such that it can be relatively easily combined with a number of these techniques, including parallel computing. In the SG model, the most time consuming computation is the solution of the Riemann problem at the edges of computational cells, because it is performed at small cells resolution. These computations, which provide the mass and momentum fluxes between the cells, can be performed independently, and therefore parallel computing can speed up the simulation significantly. While great efforts have been made in this thesis to test the model through a sequence of increasingly stringent test cases, this is far from exhaustive. Therefore, further tests would be useful to gain further insights into the limits of the proposed approach, which will also guide new developments in this area.

Appendices

Appendix A

Shallow water equations

The shallow water equations (SWE) are a hyperbolic system of partial differential equations (PDEs) that are employed for the simulation of a large range of free surface flows from oceans, coastal areas and estuaries to rivers. Urban flood inundations by Liang (2010), rainfall-runoff processes by Esteves et al. (2000) and Simons et al. (2014), tidal hydrodynamics and ocean tsunami hazards by Sugawara and Goto (2012), Pelling et al. (2013) and Hou et al. (2015) are examples showing the extensive applicability of shallow water equations in free surface flow studies. Shallow water equations are able to represent complex hydrodynamics including discontinued water surface. These equations appear in engineering studies in 1D, 2D and 3D forms and they are applicable to cases when the vertical scale of flow is much smaller than the horizontal dimension (the reason why they are called 'shallow' water equations).

These governing PDEs are derived from continuity and Navier-Stokes momentum equations, which are based on conservation of mass and momentum. The Navier–Stokes momentum equation is a particular form of the Cauchy momentum equation, (Wikipedia, 2015). In this section, the derivation of depth averaged (two-dimensional) shallow water equations is described. This derivation is completely similar to that mentioned by Toro (2001). To avoid any ambiguity, it should be noted that the process of the derivation starts from the 3D continuity and Navier-Stokes equations, in which z is adopted as the vertical axis of the Cartesian system, while in the rest of the thesis z represents bed level. Therefore, in this appendix, b instead of z is employed for the bed level.

A.1 Derivation of 2D SWE or mass and momentum conservation laws

The general, differential form of the conservation laws of mass and momentum (Cauchy equations) for a compressible fluid is defined as (Toro, 2001),

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{V}) = 0, \tag{A.1}$$

$$\frac{\partial}{\partial t}(\rho \boldsymbol{V}) + \nabla \cdot \left(\rho \boldsymbol{V} \otimes \boldsymbol{V} + \boldsymbol{P}\, | - \prod\right) = \rho \boldsymbol{g}, \tag{A.2}$$

where t is time, x, y and z are distance co-ordinates, ρ is mass density and $\boldsymbol{V} = (u, v, w)$ is velocity vector, which has the components of u, v and w that are defined in the directions of x, y and z, respectively. $\boldsymbol{P}$ denotes the pressure, and $\boldsymbol{g} = (g_1, g_2, g_3)$ is a body force vector and

$$\begin{cases} \boldsymbol{V} \otimes \boldsymbol{V} = \begin{pmatrix} u^2 & uv & uw \\ vu & v^2 & vw \\ wu & wv & w^2 \end{pmatrix}, \\ | = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\ \prod = \begin{pmatrix} \tau^{xx} & \tau^{xy} & \tau^{xz} \\ \tau^{yx} & \tau^{yy} & \tau^{yz} \\ \tau^{zx} & \tau^{zy} & \tau^{zz} \end{pmatrix}, \end{cases} \tag{A.3}$$

The viscous stress tensor $\prod$ is usually modelled using the Newtonian assumption, and here this term can be neglected to decrease the complexity of the equations as is common practice in flood modelling.

Neglecting viscous effects and writing the equations based on primary physical variables such as u, v, w, ρ and $\boldsymbol{P}$, convert the equations into

$$\begin{aligned} &\tfrac{\partial \rho}{\partial t} + u\tfrac{\partial \rho}{\partial x} + v\tfrac{\partial \rho}{\partial y} + w\tfrac{\partial \rho}{\partial z} + \rho\left(\tfrac{\partial u}{\partial x} + \tfrac{\partial v}{\partial y} + \tfrac{\partial w}{\partial z}\right) = 0, \\ &\tfrac{\partial u}{\partial t} + u\tfrac{\partial u}{\partial x} + v\tfrac{\partial u}{\partial y} + w\tfrac{\partial u}{\partial z} + \tfrac{1}{\rho}\tfrac{\partial P}{\partial x} = g_1, \\ &\tfrac{\partial v}{\partial t} + u\tfrac{\partial v}{\partial x} + v\tfrac{\partial v}{\partial y} + w\tfrac{\partial v}{\partial z} + \tfrac{1}{\rho}\tfrac{\partial P}{\partial y} = g_2, \\ &\tfrac{\partial w}{\partial t} + u\tfrac{\partial w}{\partial x} + v\tfrac{\partial w}{\partial y} + w\tfrac{\partial w}{\partial z} + \tfrac{1}{\rho}\tfrac{\partial P}{\partial z} = g_3. \end{aligned} \tag{A.4}$$

Written for an incompressible free surface flow as illustrated schematically in Fig. A.1, whereas x–y is a horizontal plane, and the bed and water surface profiles are defined by the functions $b(x, y)$

and $h(x, y, t)$, respectively, the formulas are simplified, so that $\boldsymbol{g} = (0, 0, -g = -9.806\ \mathrm{m/s^2})$ and we have

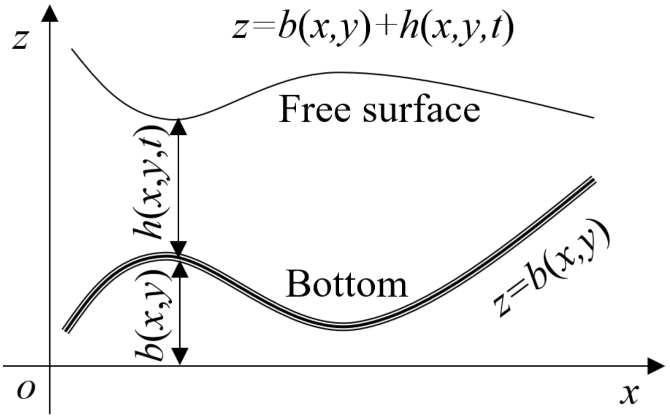


Figure A.1: Longitudinal profile of bed and water surface of an arbitrary free surface flow, (Toro, 2001).

$$\begin{aligned}
&\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0,\\
&\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial x},\\
&\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial y},\\
&\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial P}{\partial z} - g.
\end{aligned} \tag{A.5}$$

One basic assumption in derivation of shallow water equations is associated with omitting the effects of vertical acceleration components, i.e.

$$\frac{dw}{dt} = \frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} + w\frac{\partial w}{\partial z} = 0, \tag{A.6}$$

and substituting the above condition $\left(\text{i.e.}\frac{\mathrm{d}w}{\mathrm{d}t} = 0\right)$ into Eq. A.5d leads

$$\frac{\partial P}{\partial z} = -\rho g. \tag{A.7}$$

In other words, the pressure distribution in the shallow water equations is assumed hydrostatic. This implies that the pressure at the free surface is zero relative to atmospheric pressure, and changes linearly with vertical distance from the free surface. Therefore,

$$P = \rho g(\eta - z), \tag{A.8}$$

where $\eta = \eta(x, y, t) = b(x, y) + h(x, y, t)$ is the water surface elevation. Using Eq. A.8, the partial derivative of pressure with respect to x and y may be written

$$\frac{\partial P}{\partial x} = \rho g\frac{\partial \eta}{\partial x}, \qquad \frac{\partial P}{\partial y} = \rho g\frac{\partial \eta}{\partial y}. \tag{A.9}$$

Hence, $\frac{\partial P}{\partial x}$ and $\frac{\partial P}{\partial y}$ are independent of z. Therefore, the left hand sides of Eq. A.5b and Eq. A.5c, which are $\frac{du}{dt} = \frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + w\frac{\partial u}{\partial z}$ and $\frac{dv}{dt} = \frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + w\frac{\partial v}{\partial z}$ are independent of z. Then, the parameters u and v, which are the $x-$ and $y-$ direction velocity components, are independent of z, and it can be concluded that $\frac{\partial u}{\partial z} = \frac{\partial v}{\partial z} = 0$. Using this last point and substituting Eq. A.9 in Eq. A.5b and Eq. A.5c gives

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -g\frac{\partial \eta}{\partial x}, \tag{A.10}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -g\frac{\partial \eta}{\partial y}. \tag{A.11}$$

The integral of the continuity Eq. A.5a with respect to z, from $z = b(x, y)$ at the bed bottom to the free surface at $z = \eta(x, y)$ must be zero,

$$\int_{z=b}^{z=\eta} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} \right) dz = 0, \tag{A.12}$$

thus,

$$w \,\|_{z=\eta} - w \,\|_{z=b} + \int_b^\eta \frac{\partial u}{\partial x} dz + \int_b^\eta \frac{\partial v}{\partial y} dz = 0. \tag{A.13}$$

Adopting the boundary conditions described below provides the first two expressions in Eq. A.13.

$$\psi_b = \psi_b(x, y, z, t), \tag{A.14}$$

which, for the free surface, is

$$\psi_b\,(x, y, z, t) = z - \eta(x, y, t) = 0, \tag{A.15}$$

and, for the bottom, is

$$\psi_b\,(x, y, z, t) = z - b\,(x, y) = 0. \tag{A.16}$$

The boundary function is time-dependent at the free surface and time-independent at the bed. The first derivative of boundary function ψ with respect to time yields

$$\frac{d}{dt}\psi_b\,(x, y, z, t) = \frac{\partial}{\partial t}\psi_b + u\frac{\partial}{\partial x}\psi_b + v\frac{\partial}{\partial y}\psi_b + w\frac{\partial}{\partial z}\psi_b = 0, \tag{A.17}$$

and it is obvious that at the free surface, the pressure relative to atmospheric is zero,

$$P(x, y, z, t) \,\|_{z=\eta} = 0. \tag{A.18}$$

Applying the boundary condition of Eq. A.17 at the highest and lowest streamlines (i.e. free surface and bottom) results in,

$$\left(\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y} - w \right) \Bigg\|_{z=\eta} = 0, \tag{A.19}$$

$$\left(u\frac{\partial b}{\partial x} + v\frac{\partial b}{\partial y} - w\right)\Bigg\|_{z=b} = 0, \tag{A.20}$$

Therefore, the parameters of $w\ \|_{z=\eta}$ and $w\ \|_{z=b}$ can be defined by the functions of flow surface and bed and the velocity components of u and v,

$$w\ \|_{z=\eta} = \left(\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y}\right)\Bigg\|_{z=\eta}, \tag{A.21}$$

$$w\ \|_{z=b} = \left(u\frac{\partial b}{\partial x} + v\frac{\partial b}{\partial y}\right)\Bigg\|_{z=b}. \tag{A.22}$$

Substituting both Eq. A.21 and Eq. A.22 into Eq. A.13, leads to the following equation,

$$\left(\frac{\partial \eta}{\partial t} + u\frac{\partial \eta}{\partial x} + v\frac{\partial \eta}{\partial y}\right)\Bigg\|_{z=\eta} - \left(u\frac{\partial b}{\partial x} + v\frac{\partial b}{\partial y}\right)\Bigg\|_{z=b} + \int_b^\eta \frac{\partial u}{\partial x}dz + \int_b^\eta \frac{\partial v}{\partial x}dz = 0. \tag{A.23}$$

The Leibnitz formula is used to simplify the above equation

$$\frac{d}{d\alpha}\int_{\xi_1(\alpha)}^{\xi_2(\alpha)} f(\xi,\alpha)d\xi = \int_{\xi_1(\alpha)}^{\xi_2(\alpha)} \frac{\partial f}{\partial \alpha}d\xi + f(\xi_2,\alpha)\frac{\partial \xi_2}{\partial \alpha} - f(\xi_1,\alpha)\frac{\partial \xi_1}{\partial \alpha}, \tag{A.24}$$

thus, the last two integral expressions in Eq. A.23 can be rewritten as

$$\int_b^\eta \frac{\partial u}{\partial x}dz = \frac{\partial}{\partial x}\int_b^\eta u dz - u||_{z=\eta}\cdot\frac{\partial \eta}{\partial x} + u||_{z=b}\cdot\frac{\partial b}{\partial x}, \tag{A.25}$$

$$\int_b^\eta \frac{\partial v}{\partial y}dz = \frac{\partial}{\partial y}\int_b^\eta v dz - v||_{z=\eta}\cdot\frac{\partial \eta}{\partial y} + v|\,|_{z=b}\cdot\frac{\partial b}{\partial y}. \tag{A.26}$$

Integrating Eq. A.25, Eq. A.26, and Eq. A.23 gives

$$\frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}\int_b^\eta u dz + \frac{\partial}{\partial y}\int_b^\eta v dz = 0. \tag{A.27}$$

As previously mentioned, u and v are independent of z, $\eta = b + h$, and the partial derivative of b with respect to time is zero $\left(\text{i.e. } \frac{\partial b}{\partial t} = 0\right)$. This yields the form of the mass conservation (continuity) equation below,

$$\frac{\partial}{\partial t}h + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) = 0. \tag{A.28}$$

The momentum equations may also be expressed in differential conservation form. To do this, Eq. A.28 may be multiplied by u and add to Eq. A.10 that is pre-multiplied by h. The result is as follows, noting that the relation $h\frac{\partial h}{\partial x} = \frac{\partial}{\partial x}\left(\frac{1}{2}h^2\right)$ in the algebraic manipulations.

$$\frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}\left(hu^2 + \frac{1}{2}gh^2\right) + \frac{\partial}{\partial y}(huv) = -gh\frac{\partial b}{\partial x}. \tag{A.29}$$

The above equation is momentum in the x-direction. In the y-direction, an identical procedure is adopted leading to

$$\frac{\partial}{\partial t}(hv) + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}\left(hv^2 + \frac{1}{2}gh^2\right) = -gh\frac{\partial b}{\partial y}. \tag{A.30}$$

The three partial differential equations A.28, A.29, and A.30 may be rewritten as

$$\frac{\partial}{\partial t}\boldsymbol{U} + \frac{\partial}{\partial x}\boldsymbol{F}(\boldsymbol{U}) + \frac{\partial}{\partial y}\boldsymbol{G}(\boldsymbol{U}) = \boldsymbol{S}(\boldsymbol{U}), \tag{A.31}$$

where

$$\boldsymbol{U} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \quad \boldsymbol{F}(\boldsymbol{U}) = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix}, \quad \boldsymbol{G}(\boldsymbol{U}) = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2, \end{pmatrix}, \quad \boldsymbol{S}(\boldsymbol{U}) = \begin{pmatrix} S_1 \\ S_2 \\ S_3, \end{pmatrix}, \tag{A.32}$$

where, $\boldsymbol{U}$ is the vector of conserved variables, $\boldsymbol{F}(\boldsymbol{U})$ and $\boldsymbol{G}(\boldsymbol{U})$ are flux vectors in the $x-$ and $y-$ directions, and $\boldsymbol{S}(\boldsymbol{U})$ is the source term vector. The source term vector may include different terms such as Coriolis forces, wind forces, bottom friction. Setting $\boldsymbol{S}(\boldsymbol{U}) = \boldsymbol{0}$ gives the homogenous version of the shallow water equations.

Appendix B

Gradually varied flows in 1D channels

The 1D-shallow water equations for a non-prismatic, complicated geometry, as shown in Fig. B.1, can be written as (see MacDonald, 1996)

$$\begin{cases} \frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \\ \frac{\partial Q}{\partial t} + \frac{\partial F}{\partial x} = D, \end{cases}, \tag{B.1}$$

in which,

$$\begin{cases} F = \frac{Q^2}{A} + gI_1, & I_1 = \int_0^h (h-\xi)B(x,\xi)d\xi' \\ D = gA\left(s_0 - s_f\right) + gI_2, & I_2 = \int_0^h (h-\xi)\frac{\partial B(x,\xi)}{\partial x}d\xi, \end{cases}, \tag{B.2}$$

where A (m^2) is the local cross-sectional area, Q (m$^3/s$) represents the flow discharge, I_1 (m^3) and

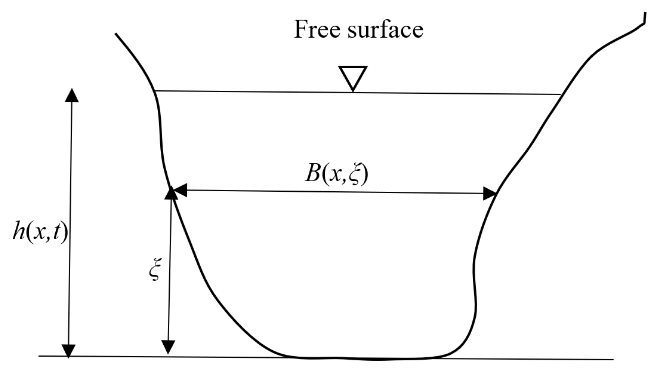


Figure B.1: A typical channel cross-section.

I_2 (m^2) are hydrostatic pressure and side reaction terms for a non-prismatic channel (the effects of channel contraction and expansion on the flow), respectively, B (m) is channel width, h (m) represents the water depth and ξ (m) denotes an elevation parameter, changing from zero to water depth level at each point in agreement with the definitions in Fig. B.1.

Eq. B.1 for steady state flow yields

$$\begin{cases} \frac{\partial Q}{\partial x} = 0, \\ \frac{\partial F}{\partial x} = D. \end{cases} \tag{B.3}$$

The second equation of Eq. B.3 gives

$$\frac{\partial F}{\partial x} = D \Rightarrow \frac{\partial}{\partial x}\left(\frac{Q^2}{A} + gI_1\right) = gA\left(s_0 - s_f\right) + gI_2, \tag{B.4}$$

alternatively,

$$\underbrace{\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right)}_{tm_1} + g\underbrace{\frac{\partial}{\partial x}\left(I_1\right)}_{tm_2} = gA\left(s_0 - s_f\right) + gI_2. \tag{B.5}$$

In Eq. B.5, tm_1 equals

$$\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) = \frac{1}{A^2}\left(A\frac{\partial Q^2}{\partial x} - Q^2\frac{\partial A}{\partial x}\right) = \frac{1}{A^2}\left(A\frac{\partial Q^2}{\partial Q}\frac{\partial Q}{\partial x} - Q^2\frac{\partial A}{\partial x}\right). \tag{B.6}$$

From the first equation of Eq. B.3, $\frac{\partial Q}{\partial x} = 0$ then,

$$\begin{aligned}
&\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) = \frac{-Q^2}{A^2}\frac{\partial A}{\partial x} = \frac{-Q^2}{A^2}\frac{\partial}{\partial x}\int_0^{h(x)} B(x,h)d\xi \\
&= \frac{-Q^2}{A^2}\lim_{\Delta\mathrm{x}\to 0}\frac{1}{\Delta\mathrm{x}}\left\{\int_0^{h(x+\Delta\mathrm{x})} B\left[x+\Delta\mathrm{x}, h\left(x+\Delta\mathrm{x}\right)\right]d\xi - \int_0^{h(x)} B\left[x, h\left(x\right)\right]d\xi\right\} \\
&= \frac{-Q^2}{A^2}\lim_{\Delta\mathrm{x}\to 0}\frac{1}{\Delta\mathrm{x}}\left\{\int_0^{h(x)} B\left[x+\Delta\mathrm{x}, h\left(x+\Delta\mathrm{x}\right)\right]d\xi - \int_0^{h(x)} B\left[x, h\left(x\right)\right]d\xi\right. \\
&\left. + \int_{h(x)}^{h(x+\Delta\mathrm{x})} B\left[x+\Delta\mathrm{x}, h\left(x+\Delta\mathrm{x}\right)\right]d\xi\right\} \\
&= \frac{-Q^2}{A^2}\left\{\int_0^{h(x)}\frac{\partial B}{\partial x}d\xi\right\} + \frac{-Q^2}{A^2}\lim_{\Delta\mathrm{x}\to 0}\frac{1}{\Delta\mathrm{x}}\left\{\int_{h(x)}^{h(x+\Delta\mathrm{x})} B\left[x+\Delta\mathrm{x}, h\left(x+\Delta\mathrm{x}\right)\right]d\xi\right\} \\
&= \frac{-Q^2}{A^2}\left\{\int_0^{h(x)}\frac{\partial B}{\partial x}d\xi + B\left[x, h\left(x\right)\right]\frac{\partial h}{\partial x}\right\}
\end{aligned} \tag{B.7}$$

consequently, tm_1 is,

$$\frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) = \frac{-Q^2}{A^2}\left\{\int_0^{h(x)}\frac{\partial B}{\partial x}d\xi + T\frac{\partial h}{\partial x}\right\}. \tag{B.8}$$

In Eq. B.5, tm_2 is computed as

$$\frac{\partial}{\partial x}\left(I_1\right) = \frac{\partial}{\partial x}\int_0^h (h-\xi)\,Bd\xi$$

$$= \int_0^h \frac{\partial}{\partial x}\left[(h-\xi)\,B\right]d\xi = \int_0^h \left[B\frac{\partial h}{\partial x} + (h-\xi)\,\frac{\partial B}{\partial x}\right]d\xi \quad , \tag{B.9}$$

$$\frac{\partial h}{\partial x}\underbrace{\int_0^h Bd\xi}_{A} + \underbrace{\int_0^h (h-\xi)\,\frac{\partial B}{\partial x}d\xi}_{I_2}$$

then, tm_2 is

$$\frac{\partial}{\partial x}\left(I_1\right) = \frac{\partial h}{\partial x}A + I_2,. \tag{B.10}$$

Eqs. B.5, B.8 (i.e. tm1) and B.10 (i.e. tm_2) yield

$$\frac{-Q^2}{A^2}\left\{\int_0^h \frac{\partial B}{\partial x}d\xi + T\frac{\partial h}{\partial x}\right\} + g\left(\frac{\partial h}{\partial x}A + I_2\right) = gA\left(s_0 - s_f\right) + gI_2, \tag{B.11}$$

$$\frac{\partial h}{\partial x} = \frac{1}{1-\frac{Q^2T}{gA^3}}\left[\left(s_0 - s_f\right) + \frac{Q^2}{gA^3}\int_0^h \frac{\partial B}{\partial x}d\xi\right]. \tag{B.12}$$

Appendix C

Error analysis of the Taylor series approximation of conveyance factor in the proposed *q* distribution

In Section 3.3.2, a new method was developed to distribute $\bar{\boldsymbol{q}}$ of a computational cell at its edges to provide an accurate reconstruction of this conservative variable for the Riemann problem solution. The proposed method was based on a truncated Taylor series expansion about average values of the flow depth and Manning's coefficient at the edges, and under the assumption of a constant friction slope. In the process of deriving the distribution, two functions ψ_e and ψ_a were introduced as conveyance factor such that ψ_a is an approximation of ψ_e. Herein, it is assessed how accurately, ψ_a approximates ψ_e. In this analysis, the effect of water depth is solely considered and Manning roughness is treated as a constant. Thus ψ_e at time level n and the edge Γ_4 (as an example) would appear as

$$\psi^{n}_{e_i}\Big|_{\Gamma_4} = \left[\frac{1}{N_{\Gamma_4} n_M} \sum_{k=1}^{N_{\Gamma_4}} \left(h^{\frac{5}{3}} \right)_{1,k} \right]^{n}_{i} . \tag{C.1}$$

On the other hand,$\psi^{n}_{e_i}$ is

$$\psi^{n}_{a_i}\Big|_{\Gamma_4} = \left\{ \left[(\eta - \tilde{z})^{\frac{5}{3}}\, T_a + \frac{5}{3}\,(\eta - \tilde{z})^{\frac{2}{3}}\, T_b + \frac{5}{9}\,(\eta - \tilde{z})^{\frac{-1}{3}}\, T_c \right]_{\Gamma_4} \right\}^{n}_{i} , \tag{C.2}$$

and as n_{M_i} is constant, T_a to T_c are converted into

$$T^{n}_{a_i}\Big|_{\Gamma_4} = \left\{ \frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left(\frac{1}{n_M} \right)_{1,k} \right\}^{n}_{i} = \left(\frac{1}{n_M} \right)_{i} , \tag{C.3}$$

$$
\begin{aligned}
T^n_{b_i}\Big|_{\Gamma_4} &= \left\{ \frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left[\frac{\left(\tilde{z}|_{\Gamma_4} - z\right)}{n_M} \right]_{1,k} \right\}^n_i = \left(\frac{1}{n_M}\right)_i \left[\frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left(\tilde{z}|_{\Gamma_4} - z|_{1,k}\right) \right]^n_i \\
&= \left(\frac{1}{n_M}\right)_i \left\{ \left[\frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left(\tilde{z}|_{\Gamma_4}\right) \right]^n_i - \left[\frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} (z)_{1,k} \right]^n_i \right\} \\
&= \left(\frac{1}{n_M}\right)_i \left[\left(\tilde{z}|_{\Gamma_4}\right)^n_i - \left(\tilde{z}|_{\Gamma_4}\right)^n_i \right] = 0,
\end{aligned}
\tag{C.4}
$$

$$
\begin{aligned}
T^n_{c_i}\Big|_{\Gamma_4} &= \left\{ \frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left[\frac{\left(\tilde{z}|_{\Gamma_4} - z\right)^2}{n_M} \right]_{1,k} \right\}^n_i = \left(\frac{1}{n_M}\right)_i \left[\frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left(\tilde{z}|_{\Gamma_4} - z|_{1,k}\right)^2 \right]^n_i \\
&= \left(\frac{1}{n_M}\right)_i \left(\sigma_z^2\right)^n_i,
\end{aligned}
\tag{C.5}
$$

in which $\left(\sigma_z^2\right)^n_i = \left[\frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left(\tilde{z}|_{\Gamma_4} - z|_{1,k}\right)^2 \right]$ is the variance of the bed level of the submerged small cells adjacent to the edge Γ_4. Then,

$$
\psi^n_{a_i}\Big|_{\Gamma_4} = \left(\frac{1}{n_M}\right)_i \left\{ \left[(\eta - \tilde{z})^{\frac{5}{3}} + \frac{5}{9} (\eta - \tilde{z})^{\frac{-1}{3}} \sigma_z^2 \right]_{\Gamma_4} \right\}^n_i . \tag{C.6}
$$

The error $\left(E_p\right)$ of $\psi^n_{a_i}\Big|_{\Gamma_4}$ compared to $\psi^n_{e_i}\Big|_{\Gamma_4}$ is given by

$$
E_p = \frac{\left| \left(\frac{1}{n_M}\right)_i \left\{ \left[(\eta - \tilde{z})^{\frac{5}{3}} + \frac{5}{9} (\eta - \tilde{z})^{\frac{-1}{3}} \sigma_z^2 \right]_{\Gamma_4} \right\}^n_i - \left[\frac{1}{N_{\Gamma_4} n_M} \sum_{k=1}^{N_{\Gamma_4}} \left(h^{\frac{5}{3}}\right)_{1,k} \right]^n_i \right|}{\left[\frac{1}{N_{\Gamma_4} n_M} \sum_{k=1}^{N_{\Gamma_4}} \left(h^{\frac{5}{3}}\right)_{1,k} \right]^n_i}, \tag{C.7}
$$

alternatively,

$$
E_p = \frac{\left| \left\{ \left[(\eta - \tilde{z})^{\frac{5}{3}} + \frac{5}{9} (\eta - \tilde{z})^{\frac{-1}{3}} \sigma_z^2 \right]_{\Gamma_4} \right\}^n_i - \left[\frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left(h^{\frac{5}{3}}\right)_{1,k} \right]^n_i \right|}{\left[\frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left(h^{\frac{5}{3}}\right)_{1,k} \right]^n_i}. \tag{C.8}
$$

Herein, 21 challenging cross-sections of large cells are adopted to assess the range of errors. These cross-sections are shown in Fig. C.1. Error analysis of each cross-section is performed for three water levels. The lowest water level (η_low) is defined to have a shallow water depth over the maximum bed elevation (1 mm above maximum bed level). The two other water levels are employed to evaluate the approximation for deeper water depths, namely middle (η_mid) and highest water level (η_high), respectively. Table C.1 illustrates the water levels (η) and their corresponding errors (E_p) where N_{Γ_4} is assumed 30. According to this table, except cross-sections 10 and 13 (for η_low only), the errors for all cross-sections are negligible. However, for all cross-sections, including cross-sections 10 and 13, the errors decline sharply as the water level increases. Cross-sections 10 and 13 represent the flood being on the threshold of overflowing from the main channel to the floodplain, in which case there is a large area of shallow water depth extending over the floodplain.

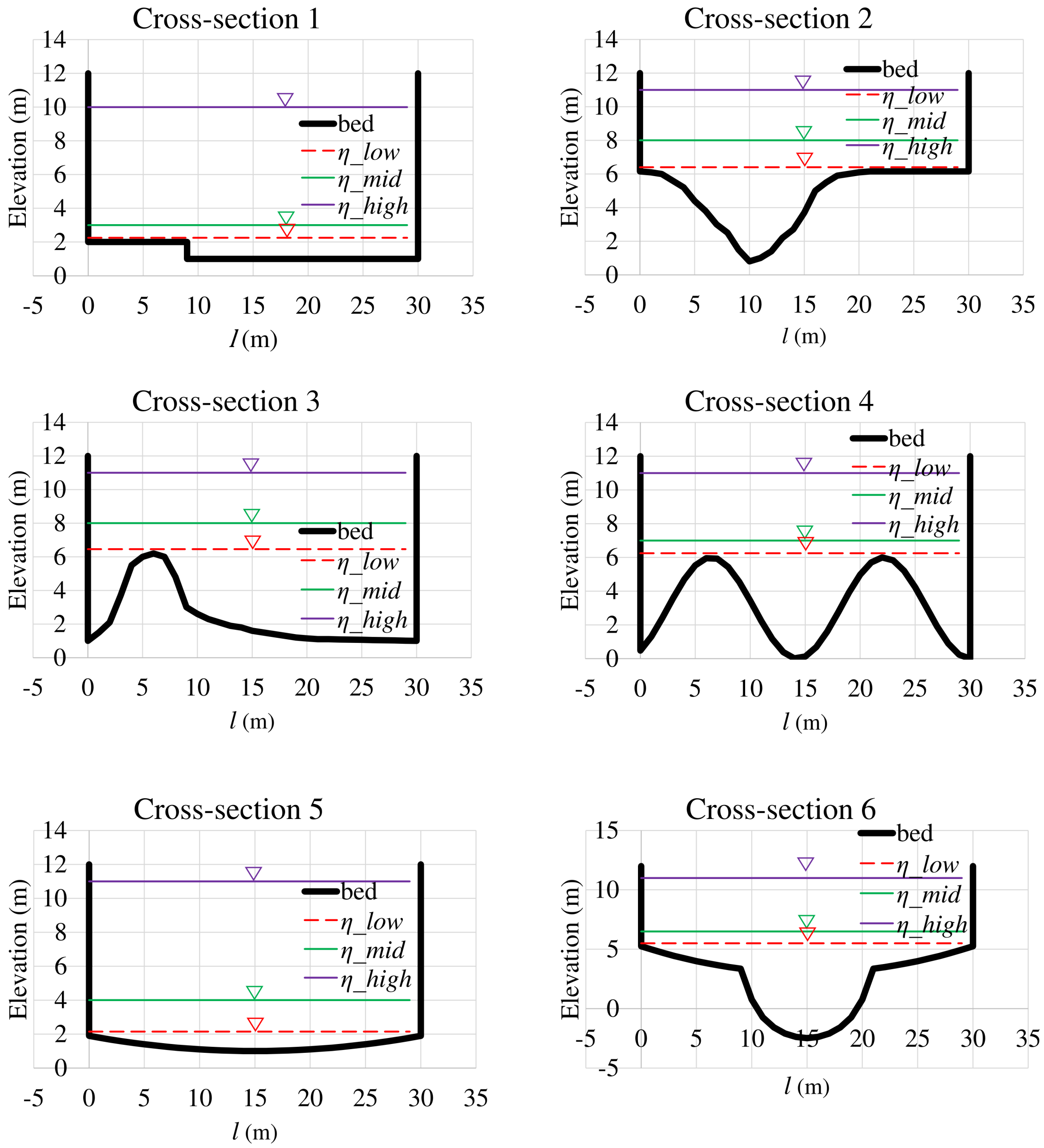


Figure C.1: Adopted cross-sections for error analysis when using ψ_a instead of ψ_e (part 1).

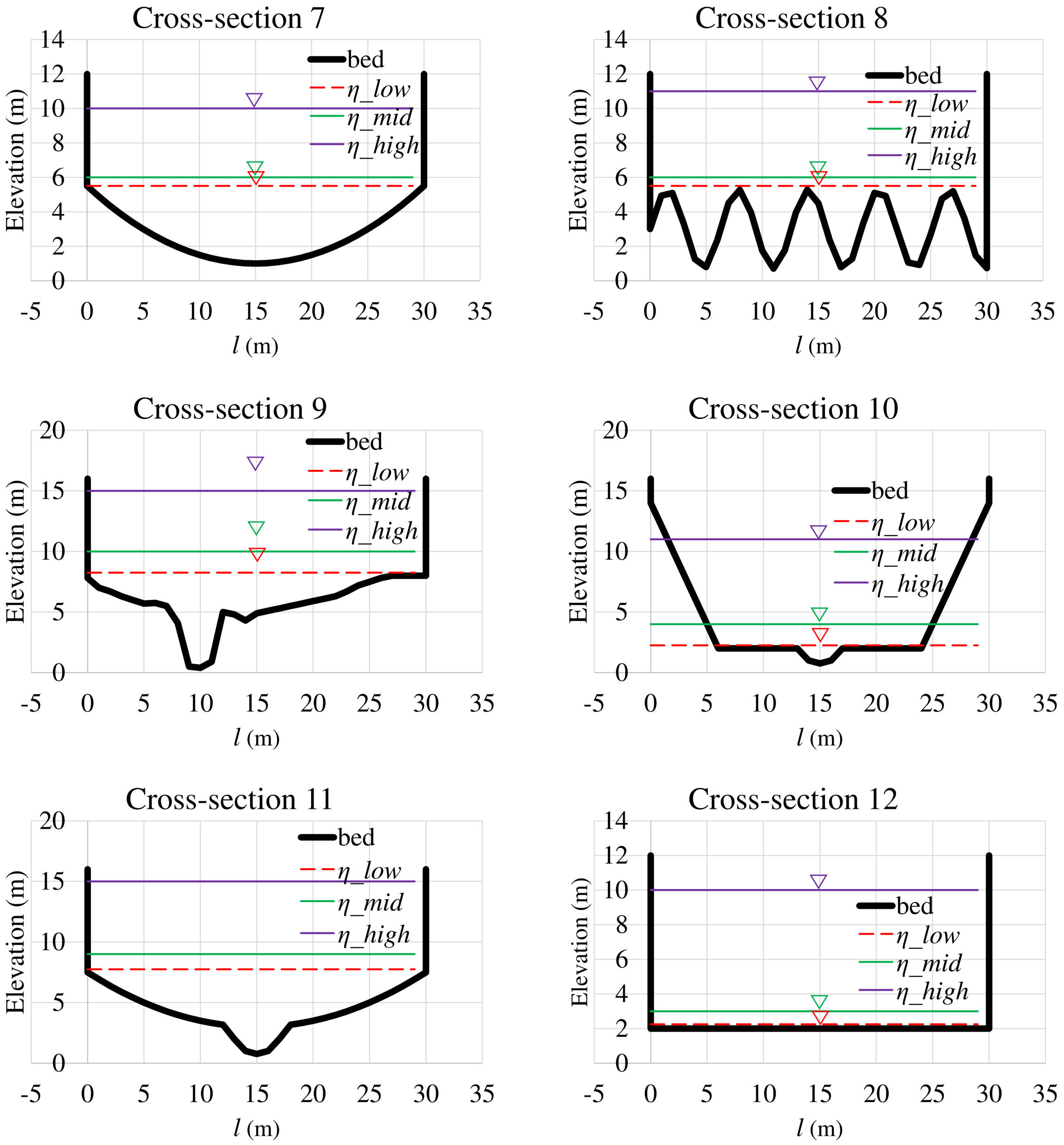


Figure C.1: Adopted cross-sections for error analysis when using ψ_a instead of ψ_e (part 2).

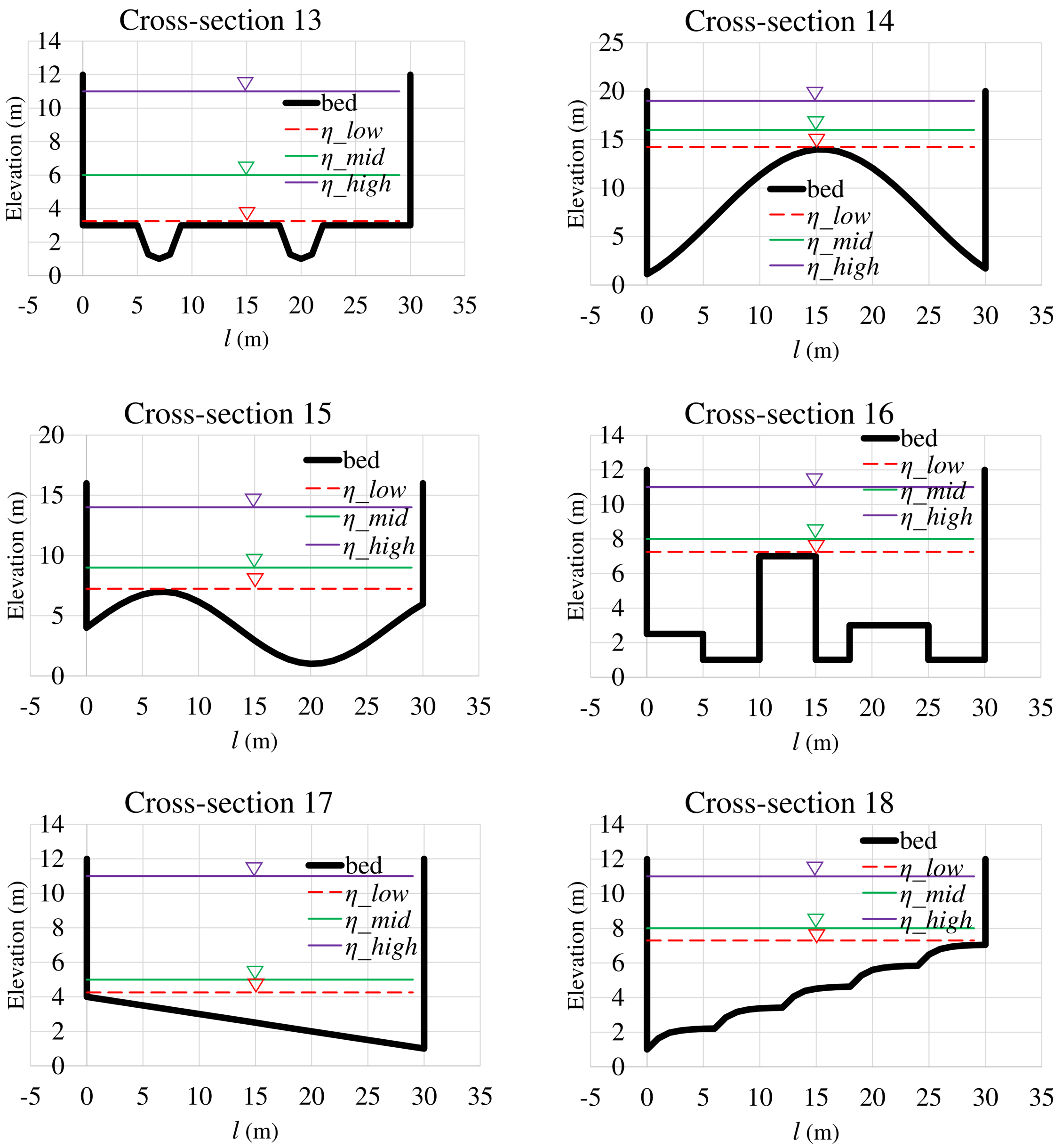


Figure C.1: Adopted cross-sections for error analysis when using ψ_a instead of ψ_e (part 3).

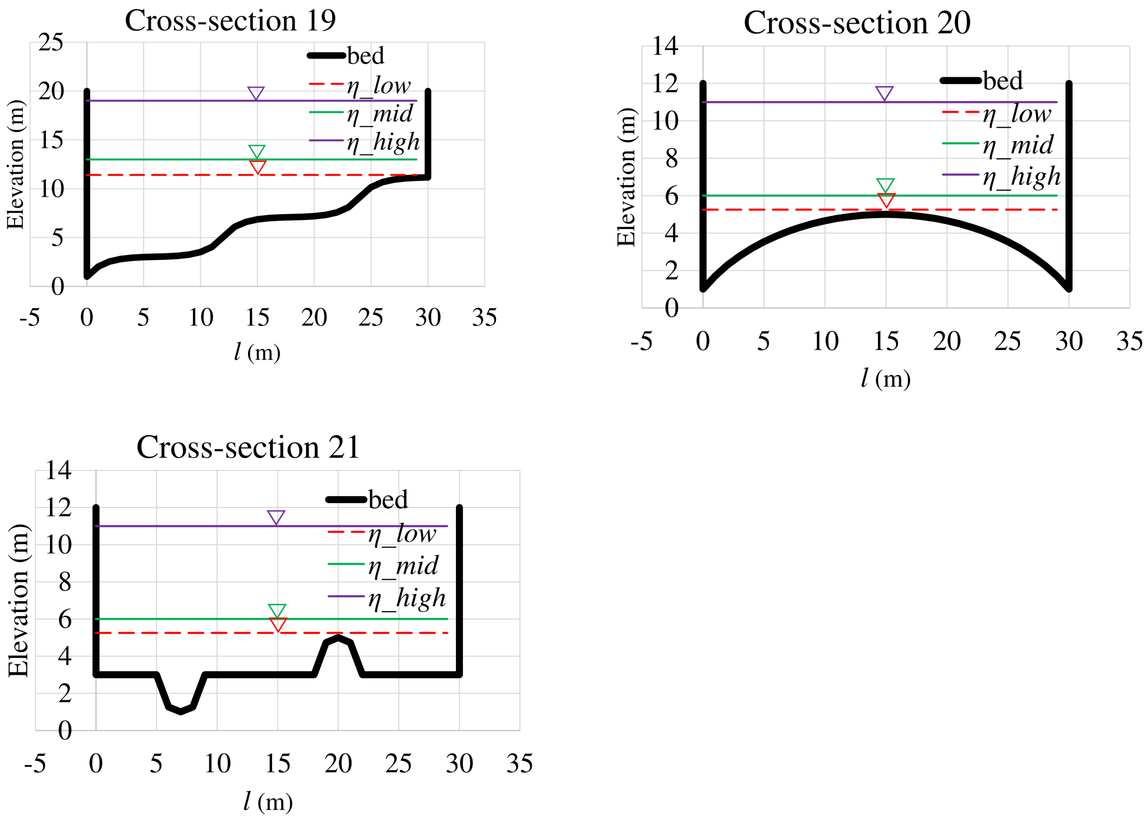


Figure C.1: Adopted cross-sections for error analysis when using ψ_a instead of ψ_e (part 4).

Table C.1: Error analysis when using ψ_a instead of ψ_e for the cross-sections provided in Fig. C.1.

Cross section	η_low (m)	η_mid (m)	η_high (m)	$E_{p_{\eta\_low}}$ (%)	$E_{p_{\eta\_mid}}$ (%)	$E_{p_{\eta\_high}}$ (%)
1	2.001	3	10	2.45	0.12	0.00
2	6.151	8	11	2.04	0.56	0.11
3	6.201	8	11	2.04	0.56	0.11
4	5.997	7	11	0.82	0.23	0.01
5	1.901	4	11	0.82	0.01	0.00
6	5.251	6.5	11	0.75	0.50	0.09
7	5.501	6	10	0.83	0.45	0.03
8	5.279	6	11	0.83	0.25	0.00
9	8.001	10	15	1.38	0.46	0.06
10	2.001	4	11	16.11	0.84	0.54
11	7.501	9	15	0.45	0.07	0.00
12	2.001	3	10	0.00	0.00	0.00
13	3.001	6	11	11.06	0.08	0.01
14	13.983	16	19	0.56	0.08	0.00
15	6.995	9	14	0.55	0.03	0.00
16	7.001	8	11	1.44	0.64	0.14
17	4.001	5	11	0.52	0.06	0.00
18	7.049	8	11	0.74	0.19	0.01
19	11.162	13	19	0.92	0.26	0.03
20	5.001	6	11	1.00	0.43	0.02
21	5.001	6	11	0.36	0.06	0.00

Appendix D

Proof of the bed slope relation presented by Valiani and Begnudelli (2006)

As mentioned in Section 3.3.3 (bed slope source term), Valiani and Begnudelli (2006) showed that $ghs_{0_x} = \frac{\partial}{\partial x}\left(\frac{1}{2}gh^2\right)$ when water level is assumed constant across the element. Herein, the derivation of this relation is provided.

$$ghs_{0_x} = gh\frac{-\partial z}{\partial x} = -g\left(\eta - z\right)\frac{\partial z}{\partial x} = -g\eta\frac{\partial z}{\partial x} + gz\frac{\partial z}{\partial x}, \tag{D.1}$$

when η is constant across the element dx then,

$$ghs_{0_x} = \frac{\partial}{\partial x}\left(-g\eta z\right) + \frac{g}{2}\frac{\partial}{\partial x}\left(z^2\right) = \frac{\partial}{\partial x}\left(-g\eta z + \frac{g}{2}z^2\right), \tag{D.2}$$

as η is constant, then $\frac{\partial}{\partial x}\left(\frac{g}{2}\eta^2\right) = 0$, and

$$ghs_{0_x} = \frac{\partial}{\partial x}\left(-g\eta z + \frac{g}{2}z^2 + \frac{g}{2}\eta^2\right) = \frac{\partial}{\partial x}\left[\frac{g}{2}\left(\eta - z\right)^2\right], \tag{D.3}$$

$$ghs_{0_x} = \frac{\partial}{\partial x}\left[\frac{g}{2}\left(\eta - z\right)^2\right] = \frac{\partial}{\partial x}\left(\frac{g}{2}h^2\right). \tag{D.4}$$

References

Abbott, M. B. (1979). *Computational Hydraulics: Elements of the Theory of Free Surface Flows*. Pitman, London.

Achdou, Y. and Bernardi, C. (2001). Adaptive finite volume or finite element discretization of darcy's equations with variable permeability. *COMPTES RENDUS DE L ACADEMIE DES SCIENCES SERIE I-MATHEMATIQUE*, 333(7):693–698.

Agbaglah, G., Delaux, S., Fuster, D., Hoepffner, J., Josserand, C., Popinet, S., Ray, P., Scardovelli, R., and Zaleski, S. (2011). Parallel simulation of multiphase flows using octree adaptivity and the volume-of-fluid method. *Comptes Rendus Mecanique*, 339(2-3):194–207.

Alcrudo, F. (2002). A state of the art review on mathematical modelling of flood propagation. In *First IMPACT project Workshop, Wallingford, UK, available at: http://www. impact-project. net/cd/papers/print/008_pr_02-05-16_IMPACT_Alcrudo. pdf (last access: December 2013)*.

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.

Alcrudo, F. and García-Navarro, P. (1993). A high-resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations. *International Journal for Numerical Methods in Fluids*, 16(6):489–505.

Almasi, G. S. and Gottlieb, A. (1989). *Highly parallel computing*. Redwood City, Calif.: Benjamin/Cummings.

Almgren, A. S. (2011). Introduction to block-structured adaptive mesh refinement (AMR). Retrieved from http://hipacc.ucsc.edu/Lecture%20Slides/2011ISSAC/AlmgrenHIPACC_July2011.pdf. [Accessed: July 2011].

Alsabbagh, M. (2015). Parallel algorithms advantages and disadvantages. Retrieved from http://www.slideshare.net/lucky43/parallel-computing-advantages-and-disadvantages. [Accessed: July 2015].

Altomare, C., Domínguez, J. M., Crespo, A. J. C., González-Cao, J., Suzuki, T., Gómez-Gesteira, M., and Troch, P. (2017). Long-crested wave generation and absorption for SPH-based DualSPHysics model. *Coastal Engineering*, 127:37–54.

Altomare, C., Viccione, G., Tagliafierro, B., Bovolin, V., Domínguez, J. M., and Crespo, A. J. C. (2018). Free-surface flow simulations with Smoothed Particle Hydrodynamics method using high-performance computing. *Computational fluid dynamics-basic instruments and applications in science. InTech*, pages 73–100.

An, H., Yu, S., Lee, G., and Kim, Y. (2015). Analysis of an open source quadtree grid shallow water flow solver for flood simulation. *Quaternary International*, 384:118–128.

Aricò, C. and Nasello, C. (2018). Comparative analyses between the zero-inertia and fully dynamic models of the shallow water equations for unsteady overland flow propagation. *Water*, 10(1):44.

Aronica, G., Tucciarelli, T., and Nasello, C. (1998). 2D multilevel model for flood wave propagation in flood-affected areas. *Journal of water resources planning and management*, 124(4):210–217.

Audusse, E., Bouchut, F., Bristeau, M.-O., Klein, R., and Perthame, B. t. (2004). A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing*, 25(6):2050–2065.

Badrzadeh, H., Sarukkalige, R., and Jayawardena, A. W. (2013). Impact of multi-resolution analysis of artificial intelligence models inputs on multi-step ahead river flow forecasting. *Journal of Hydrology*, 507:75–85.

Baeza, A. and Mulet, P. (2006). Adaptive mesh refinement techniques for high-order shock capturing schemes for multi-dimensional hydrodynamic simulations. *International Journal for Numerical Methods in Fluids*, 52(4):455–471.

Barreiro, A., Crespo, A. J. C., Domínguez, J. M., and Gómez-Gesteira, M. (2013). Smoothed particle hydrodynamics for coastal engineering problems. *Computers & Structures*, 120:96–106.

Bates, P. D. (2000). Development and testing of a subgrid-scale model for moving-boundary hydrodynamic problems in shallow water. *Hydrological Processes*, 14(11-12):2073–2088.

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.

Bear, J. (1988). *Dynamics of fluids in porous media.* Dover, New York.

Bechteler, W., Hartmaan, S., and Otto, A. J. (1994). Coupling of 2D and 1D models and integration into geographic information systems (GIS). In *2nd International Conference on River Flood Hydraulics*, pages 155–166. Wiley, Chichester, UK.

Begnudelli, L. and Sanders, B. F. (2006). Unstructured grid finite-volume algorithm for shallow-water flow and scalar transport with wetting and drying. *Journal of hydraulic engineering*, 132(4):371–384.

Begnudelli, L. and Sanders, B. F. (2007). Conservative wetting and drying methodology for quadrilateral grid finite-volume models. *Journal of Hydraulic Engineering*, 133(3):312–322.

Begnudelli, L., Sanders, B. F., and Bradford, S. F. (2008). Adaptive Godunov-based model for flood simulation. *Journal of Hydraulic Engineering*, 134(6):714–725.

Benkhaldoun, F. and Seaïd, M. (2010). A simple finite volume method for the shallow water equations. *Journal of computational and applied mathematics*, 234(1):58–72.

Berger, M. J. and Colella, P. (1989). Local adaptive mesh refinement for shock hydrodynamics. *Journal of computational Physics*, 82(1):64–84.

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.

Berger, M. J. and Oliger, J. (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53:484–512.

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

Berz, G. (2000). Flood disasters: lessons from the past—worries for the future. In *Proceedings of the Institution of Civil Engineers-Water and Maritime Engineering*, volume 142, pages 3–8. Thomas Telford Ltd.

Bladé, E., Gómez, M., and Dolz, J. (1994). Quasi-two dimensional modelling of flood routing in rivers and floodplains by means of storage cells. In *Modelling of flood propagation over initially dry areas*, pages 156–170. ASCE.

Borthwick, A., Cruz León, S., and Józsa, J. (2001). The shallow flow equations solved on adaptive quadtree grids. *International Journal for Numerical Methods in Fluids*, 37(6):691–719.

Bradford, R., O'Sullivan, J. J., Van der Craats, I., Krywkow, J., Rotko, P., Aaltonen, J., Bonaiuto, M., De Dominicis, S., Waylen, K., and Schelfaut, K. (2012). Risk perception-issues for flood management in Europe. *Natural Hazards & Earth System Sciences*, 12(7):2299–2309.

Bradford, S. F. and Sanders, B. F. (2002). Finite-volume model for shallow-water flooding of arbitrary topography. *Journal of Hydraulic Engineering*, 128(3):289–298.

Brown, J. D., Spencer, T., and Moeller, I. (2007). Modeling storm surge flooding of an urban area with particular reference to modeling uncertainties: A case study of Canvey Island, United Kingdom. *Water Resources Research*, 43(6).

Brufau, P. and García-Navarro, P. (2000). Two-dimensional dam break flow simulation. *International Journal for numerical methods in fluids*, 33(1):35–57.

Brufau, P., García-Navarro, P., and Vázquez-Cendón, M. (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.

Brufau, P., Vázquez-Cendón, M., and García-Navarro, P. (2002). A numerical model for the flooding and drying of irregular domains. *International Journal for Numerical Methods in Fluids*, 39(3):247–275.

Bruwier, M., Archambeau, P., Erpicum, S., Pirotton, M., and Dewals, B. (2017). Shallow-water models with anisotropic porosity and merging for flood modelling on Cartesian grids. *Journal of Hydrology*, 554:693–709.

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.

Casas, A., Lane, S. N., Yu, D., and Benito, G. (2010). A method for parameterising roughness and topographic sub-grid scale effects in hydraulic modelling from LiDAR data. *Hydrology and Earth System Sciences*, 14:1567–1579.

Castellarin, A., Domeneghetti, A., and Brath, A. (2011). Identifying robust large-scale flood risk mitigation strategies: A quasi-2D hydraulic model as a tool for the Po river. *Physics and Chemistry of the Earth, Parts A/B/C*, 36(7-8):299–308.

Casulli, V. (2009). A high-resolution wetting and drying algorithm for free-surface hydrodynamics. *International Journal for Numerical Methods in Fluids*, 60(4):391–408.

Casulli, V. and Stelling, G. S. (2011). Semi-implicit subgrid modelling of three-dimensional free-surface flows. *International journal for numerical methods in fluids*, 67(4):441–449.

Cea, L. and Bladé, E. (2015). A simple and efficient unstructured finite volume scheme for solving the shallow water equations in overland flow applications. *Water resources research*, 51(7):5464–5486.

Cea, L. and Vázquez-Cendón, M. E. (2010). Unstructured finite volume discretization of two-dimensional depth-averaged shallow water equations with porosity. *International journal for numerical methods in fluids*, 63(8):903–930.

Chen, A. S., Evans, B., Djordjević, S., and Savić, D. A. (2012). A coarse-grid approach to representing building blockage effects in 2D urban flood modelling. *Journal of Hydrology*, 426:1–16.

Chen, W. B., Liu, W. C., and Wu, C. Y. (2013). Coupling of a one-dimensional river routing model and a three-dimensional ocean model to predict overbank flows in a complex river–ocean system. *Applied Mathematical Modelling*, 37(9):6163–6176.

Chow, V. T. (1959). *Open channel hydraulics*. McGraw-Hill, New York.

Cohen, S., Praskievicz, S., and Maidment, D. R. (2018). Featured Collection Introduction: National Water Model. *JAWRA Journal of the American Water Resources Association*, 54(4):767–769.

Cossins, P. J. (2010). *The Gravitational Instability and its Role in the Evolution of Protostellar and Protoplanetary Discs. Chapter 3: Smoothed Particle Hydrodynamics-Or: How I Learned to Stop Worrying and Love the Lagrangian*. PhD thesis, PhD thesis, University of Leicester.

Costabile, P., Costanzo, C., and Macchione, F. (2012). Comparative analysis of overland flow models using finite volume schemes. *Journal of hydroinformatics*, 14(1):122–135.

Costabile, P., Costanzo, C., Macchione, F., et al. (2009). Two-dimensional numerical models for overland flow simulations. *River Basin Manage. V*, pages 137–148.

Cozzolino, L., Cimorelli, L., Della Morte, R., Pugliano, G., Piscopo, V., and Pianese, D. (2019). Flood propagation modeling with the Local Inertia Approximation: Theoretical and numerical analysis of its physical limitations. *Advances in Water Resources*, 133:103422.

Crespo, A. J. C., Altomare, C., Domínguez, J. M., González-Cao, J., and Gómez-Gesteira, M. (2017). Towards simulating floating offshore oscillating water column converters with Smoothed Particle Hydrodynamics. *Coastal Engineering*, 126:11–26.

CUDA (2015). Retrieved from
http://en.wikipedia.org/wiki/CUDA. [Accessed: June 2015].

Cunge, J. A., Holly, F. M., and Verwey, A. (1980). *Practical aspects of computational river hydraulics*. Pitman Publishing Ltd. London.

Dataflair Team, . (2019). Advantages and disadvantages of machine learning language. Retrieved from
http://data-flair.training/blogs/advantages-and-disadvantages-of-machine-learning/. [Accessed: March 2019].

de Almeida, G. A. and Bates, P. (2013). Applicability of the local inertial approximation of the shallow water equations to flood modeling. *Water Resources Research*, 49(8):4833–4844.

de Almeida, G. A., Bates, P., Freer, J. E., and Souvignet, M. (2012). Improving the stability of a simple formulation of the shallow water equations for 2-d flood modeling. *Water Resources Research*, 48(5).

de Almeida, G. A., Bates, P., and Ozdemir, H. (2018). Modelling urban floods at submetre resolution: challenges or opportunities for flood risk management? *Journal of Flood Risk Management*, 11:S855–S865.

de Borst, R. (2017). Fluid flow in fractured and fracturing porous media: A unified view. *Mechanics Research Communications*, 80:47–57.

de la Asunción, M., Castro, M. J., Fernández-Nieto, E. D., Mantas, J. M., Acosta, S. O., and González-Vida, J. M. (2013). Efficient GPU implementation of a two waves TVD-WAF method for the two-dimensional one layer shallow water system on structured meshes. *Computers & Fluids*, 80:441–452.

Defina, A. (2000). Two-dimensional shallow flow equations for partially dry areas. *Water resources research*, 36(11):3251–3264.

Defina, A., D'Alpaos, L., and Matticchio, B. (1994). A new set of equations for very shallow water and partially dry areas suitable to 2D numerical models. In *Modelling of flood propagation over initially dry areas*, pages 72–81. ASCE.

Delis, A. I., Guillard, H., and Tai, Y.-C. (2018). Numerical simulations of hydraulic jumps with the shear shallow water model. *The SMAI journal of computational mathematics*, 4:319–344.

Dhondia, J. and Stelling, G. (2002). Application of one-dimensional-two-dimensional integrated hydraulic model for flood simulation and damage assessment. In *International Conference on Hydroinformatics, UK*, volume 5, pages 265–276.

Di Baldassarre, G., Castellarin, A., Montanari, A., and Brath, A. (2009). Probability-weighted hazard maps for comparing different flood risk management strategies: a case study. *Natural Hazards*, 50:479–496.

Dineva, A., Várkonyi-Kóczy, A. R., and Tar, J. K. (2014). Fuzzy expert system for automatic wavelet shrinkage procedure selection for noise suppression. In *IEEE 18th International Conference on Intelligent Engineering Systems INES 2014*, pages 163–168. IEEE.

Duan, Y., Liu, Z., Chen, Y., and Zhu, D. (2017). Improved 2D shallow water model able to capture the effects of complex bathymetric features through their subgrid modeling. *Journal of Hydraulic Engineering*, 143(2):04016081.

Duran, A. and Marche, F. (2014). Recent advances on the discontinuous Galerkin method for shallow water equations with topography source terms. *Computers & Fluids*, 101:88–104.

Echeverribar, I., Morales-Hernández, M., Brufau, P., and García-Navarro, P. (2018). Numerical simulation of 2D real large scale floods on GPU: the Ebro River. In *E3S Web of Conferences*, volume 40, page 06007. EDP Sciences.

Erduran, K., Kutija, V., and Hewett, C. (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.

Esteves, M., Faucher, X., Galle, S., and Vauclin, M. (2000). Overland flow and infiltration modelling for small plots during unsteady rain: numerical results versus observed values. *Journal of hydrology*, 228(3-4):265–282.

Estrela, T. and Quintas, L. (1994). Use of a GIS in the modelling of flows on floodplains. In *2nd International Conference on River Flood Hydraulics*, volume 177, page 190. Wiley, Chichester, UK.

Faizollahzadeh Ardabili, S., Najafi, B., Alizamir, M., Mosavi, A., Shamshirband, S., and Rabczuk, T. (2018). Using SVM-RSM and ELM-RSM approaches for optimizing the production process of methyl and ethyl esters. *Energies*, 11(11):2889.

Fernández-Pato, J. and García-Navarro, P. (2018). Development of a new simulation tool coupling a 2D finite volume overland flow model and a drainage network model. *Geosciences*, 8(8):288.

Fewtrell, T. J., Neal, J. C., Bates, P. D., and Harrison, P. J. (2011). Geometric and structural river channel complexity and the prediction of urban inundation. *Hydrological processes*, 25(20):3173–3186.

Finaud-Guyot, P., Delenne, C., Guinot, V., and Llovel, C. (2011). 1D-2D coupling for river flow modeling Couplage 1D-2D pour la modélisation des inondations fluviales. 339(4):226–234.

Fleischmann, A., Paiva, R., and Collischonn, W. (2019). Can regional to continental river hydrodynamic models be locally relevant? a cross-scale comparison. *Journal of Hydrology X*, 3:100027.

Foresight (2004). *Foresight: future flooding. Executive summary*. London: Department of Trade and Industry, The Government Office for Science.

Fotovatikhah, F., Herrera, M., Shamshirband, S., Chau, K.-w., Faizollahzadeh Ardabili, S., and Piran, M. d. J. (2018). Survey of computational intelligence as basis to big flood management: Challenges, research directions and future work. *Engineering Applications of Computational Fluid Mechanics*, 12(1):411–437.

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.

García-Navarro, P. and Vázquez-Cendón, M. E. (2000). On numerical treatment of the source terms in the shallow water equations. *Computers & Fluids*, 29(8):951–979.

George, D. (2011). Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: Application to the Malpasset dam-break flood (France, 1959). *International Journal for Numerical Methods in Fluids*, 66(8):1000–1018.

Gettleman, J. (2017). More than 1,000 died in South Asia floods this summer. *New York Times*, page 29.

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(3):375–389.

Gizaw, M. S. and Gan, T. Y. (2016). Regional flood frequency analysis using support vector regression under historical and future climate. *Journal of Hydrology*, 538:387–398.

Godunov, S. K. (1959). A difference scheme for numerical solution of discontinuous solution of hydrodynamic equations. *Math. Sbornik*, 47:271–306.

Gomez-Gesteira, M., Rogers, B. D., Crespo, A. J., Dalrymple, R. A., Narayanaswamy, M., and Dominguez, J. M. (2012). SPHysics–development of a free-surface fluid solver–Part 1: Theory and formulations. *Computers & Geosciences*, 48:289–299.

Gong, K., Shao, S., Liu, H., Wang, B., and Tan, S. K. (2016). Two-phase SPH simulation of fluid–structure interactions. *Journal of Fluids and Structures*, 65:155–179.

Gourgue, O., Comblen, R., Lambrechts, J., Kärnä, T., Legat, V., and Deleersnijder, E. (2009). A flux-limiting wetting–drying method for finite-element shallow-water models, with application to the Scheldt Estuary. *Advances in Water Resources*, 32(12):1726–1739.

Goutal, N. and Maurel, F. (1997). *Proceedings of the 2nd workshop on dam-break wave simulation*. HE 43/97/016/B, Départment Laboratoire National d'Hydraulique, Groupe Hydraulique Fluviale Electricité de France, France.

Guinot, V. (2012). Multiple porosity shallow water models for macroscopic modelling of urban floods. *Advances in Water Resources*, 37:40–72.

Guinot, V. (2017). A critical assessment of flux and source term closures in shallow water models with porosity for urban flood simulations. *Advances in Water Resources*, 109:133–157.

Guinot, V., Delenne, C., Rousseau, A., and Boutron, O. (2018). Flux closures and source term models for shallow water models with depth-dependent integral porosity. *Advances in water resources*, 122:1–26.

Guinot, V., Sanders, B. F., and Schubert, J. E. (2017). Dual integral porosity shallow water model for urban flood modelling. *Advances in Water Resources*, 103:16–31.

Guinot, V. and Soares-Frazão, S. (2006). Flux and source term discretization in two-dimensional shallow water models with porosity on unstructured grids. *International Journal for Numerical Methods in Fluids*, 50(3):309–345.

Hageman, T. and de Borst, R. (2020). Sub-grid models for multiphase fluid flow inside fractures in poroelastic media. *Journal of Computational Physics*, page 109481.

Hallegatte, S., Green, C., Nicholls, R. J., and Corfee-Morlot, J. (2013). Future flood losses in major coastal cities. *Nature climate change*, 3(9):802–806.

Henderson, F. M. (1966). *Open channel flow*. Macmillan, New York.

Heniche, M., Secretan, Y., Boudreau, P., and Leclerc, M. (2000). A two-dimensional finite element drying-wetting shallow water model for rivers and estuaries. *Advances in water resources*, 23(4):359–372.

Hénonin, J., Hongtao, M., Zheng-Yu, Y., Hartnack, J., Havnø, K., Gourbesville, P., and Mark, O. (2015). Citywide multi-grid urban flood modelling: the July 2012 flood in Beijing. *Urban Water Journal*, 12(1):52–66.

Hernández, M. M. (2014). *Efficient explicit finite volume schemes for the shallow water equations with solute transport*. PhD thesis, Universidad de Zaragoza.

Hervouet, J.-M. (2000). A high resolution 2-D dam-break model using parallelization. *Hydrological processes*, 14(13):2211–2230.

Horritt, M. and Bates, P. (2001). Predicting floodplain inundation: raster-based modelling versus the finite-element approach. *Hydrological processes*, 15(5):825–842.

Hosseiny, H., Nazari, F., Smith, V., and Nataraj, C. (2020). A framework for modeling flood depth using a hybrid of hydraulics and Machine Learning. *Scientific Reports*, 10(1):1–14.

Hou, J., Liang, Q., Zhang, H., and Hinkelmann, R. (2015). An efficient unstructured MUSCL scheme for solving the 2D shallow water equations. *Environmental Modelling & Software*, 66:131–152.

Hou, J., Simons, F., Mahgoub, M., and Hinkelmann, R. (2013). A robust well-balanced model on unstructured grids for shallow water flows with wetting and drying over complex topography. *Computer methods in applied mechanics and engineering*, 257:126–149.

Hu, H., Zhang, J., and Li, T. (2018). Dam-break flows: Comparison between flow-3D, MIKE 3 FM, and analytical solutions with experimental data. *Applied Sciences*, 8(12):2456.

Hu, K., Mingham, C. G., and Causon, D. M. (2000). Numerical simulation of wave overtopping of coastal structures using the non-linear shallow water equations. *Coastal engineering*, 41(4):433–465.

Hu, R., Fang, F., Salinas, P., Pain, C., Domingo, N. S., and Mark, O. (2019). Numerical simulation of floods from multiple sources using an adaptive anisotropic unstructured mesh method. *Advances in water resources*, 123:173–188.

Huang, C.-J., Hsu, M.-H., Teng, W.-H., and Wang, Y.-H. (2014). The impact of building coverage in the metropolitan area on the flow calculation. *Water*, 6(8):2449–2466.

Huang, Y., Zhang, N., and Pei, Y. (2013). Well-balanced finite volume scheme for shallow water flooding and drying over arbitrary topography. *Engineering Applications of Computational Fluid Mechanics*, 7(1):40–54.

Hubbard, M. E. and Dodd, N. (2002). A 2D numerical model of wave run-up and overtopping. *Coastal Engineering*, 47(1):1–26.

Huisman, M., Varbanescu, A. L., van Werkhoven, B., and Wijs, A. (2018). An overview of GPU Computing Research in the Netherlands: Results from the NIRICT GPGPU Reconnaissance workshop. In *Fourth workshop in NIRICT GPGPU Reconnaissance workshop series*.

Hunter, N., Bates, P., Horritt, M., and Wilson, M. (2006). Improved simulation of flood flows using storage cell models. In *Proceedings of the Institution of Civil Engineers-Water Management*, volume 159, pages 9–18. Thomas Telford Ltd.

Hunter, N., Bates, P., Neelz, S., Pender, G., Villanueva, I., Wright, N., Liang, D., Falconer, R. A., Lin, B., Waller, S., et al. (2008). Benchmarking 2D hydraulic models for urban flooding. In *Proceedings of the Institution of Civil Engineers-Water Management*, volume 161, pages 13–30. Thomas Telford Ltd.

Hunter, N. M., Bates, P. D., Horritt, M. S., and Wilson, M. D. (2007). Simple spatially-distributed models for predicting flood inundation: a review. *Geomorphology*, 90(3-4):208–225.

Huthoff, F., Augustijn, D. C. M., and Hulscher, S. J. M. H. (2007). Analytical solution of the depth-averaged flow velocity in case of submerged rigid cylindrical vegetation. *Water resources research*, 43(W06413).

Jongman, B., Ward, P. J., and Aerts, J. C. (2012). Global exposure to river and coastal flooding: Long term trends and changes. *Global Environmental Change*, 22(4):823–835.

Juez, C., Lacasta, A., Murillo, J., and García-Navarro, P. (2016). An efficient GPU implementation for a faster simulation of unsteady bed-load transport. *Journal of Hydraulic Research*, 54(3):275–288.

Kalita, H. M. and Sarma, A. K. (2018). An implicit scheme for shallow water flow with wet dry interface. *Water resources*, 45(1):61–68.

Kar, A. K., Lohani, A. K., Goel, N. K., and Roy, G. P. (2010). Development of flood forecasting system using statistical and ANN techniques in the downstream catchment of Mahanadi Basin, India. *Journal of water resource and protection*, 2(10):2965.

Kärnä, T., De Brye, B., Gourgue, O., Lambrechts, J., Comblen, R., Legat, V., and Deleersnijder, E. (2011). A fully implicit wetting–drying method for DG-FEM shallow water models, with an application to the Scheldt Estuary. *Computer Methods in Applied Mechanics and Engineering*, 200(5-8):509–524.

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.

Kay, A., Crooks, S., Pall, P., and Stone, D. (2011). Attribution of Autumn/Winter 2000 flood risk in England to anthropogenic climate change: A catchment-based study. *Journal of Hydrology*, 406(1-2):97–112.

Kesserwani, G. and Liang, Q. (2010). A discontinuous Galerkin algorithm for the two-dimensional shallow water equations. *Computer Methods in Applied Mechanics and Engineering*, 199(49-52):3356–3368.

Kesserwani, G. and Liang, Q. (2012a). Dynamically adaptive grid based discontinuous Galerkin shallow water model. *Advances in water resources*, 37:23–39.

Kesserwani, G. and Liang, Q. (2012b). Locally limited and fully conserved RKDG2 shallow water solutions with wetting and drying. *Journal of scientific computing*, 50(1):120–144.

Khakimzyanov, G., Dutykh, D., Mitsotakis, D., and Shokina, N. Y. (2019). Numerical simulation of conservation laws with moving grid nodes: Application to tsunami wave modelling. *Geosciences*, 9(5):197.

Khoshkonesh, A., Nsom, B., Gohari, S., and Banejad, H. (2019). A comprehensive study on dam-break flow over dry and wet beds. *Ocean Engineering*, 188:106279.

Kim, B., Sanders, B. F., Famiglietti, J. S., and Guinot, V. (2015). Urban flood modeling with porous shallow-water equations: A case study of model errors in the presence of anisotropic porosity. *Journal of Hydrology*, 523:680–692.

Kim, B., Sanders, B. F., Schubert, J. E., and Famiglietti, J. S. (2014). Mesh type tradeoffs in 2D hydrodynamic modeling of flooding with a Godunov-based flow solver. *Advances in Water Resources*, 68:42–61.

Kim, S., Matsumi, Y., Pan, S., and Mase, H. (2016). A real-time forecast model using artificial neural network for after-runner storm surges on the Tottori Coast, Japan. *Ocean Engineering*, 122:44–53.

Kong, C. (2011). Comparison of approximate Riemann solvers. Master's thesis, Department of dvmv,University of Reading, Reading, UK.

Kuiry, S. N., Sen, D., and Bates, P. D. (2010). Coupled 1D–Quasi-2D flood inundation model with unstructured grids. *Journal of Hydraulic Engineering*, 136(8):493–506.

Lamb, R., Crossley, M., and Waller, S. (2009). A fast two-dimensional floodplain inundation model. In *Proceedings of the institution of civil engineers-water management*, volume 162, pages 363–370. Thomas Telford Ltd.

Le, H.-A., Lambrechts, J., Ortleb, S., Gratiot, N., Deleersnijder, E., and Soares-Frazão, S. (2020). An implicit wetting–drying algorithm for the discontinuous Galerkin method: application to the Tonle Sap, Mekong River Basin. *Environmental Fluid Mechanics*, pages 1–29.

Leandro, J., Chen, A., and Schumann, A. (2014). A 2D parallel diffusive wave model for floodplain inundation with variable time step (P-DWave). *Journal of Hydrology*, 517:250–259.

LeVeque, R. J. (2002). *Finite volume methods for hyperbolic problems*. Cambridge University Press, Cambridge, UK.

Li, L., Xu, H., Chen, X., and Simonovic, S. P. (2010). Streamflow forecast and reservoir operation performance assessment under climate change. *Water resources management*, 24:83.

Liang, Q. (2010). Flood simulation using a well-balanced shallow flow model. *Journal of Hydraulic Engineering*, 136(9):669–675.

Liang, Q. (2012). A simplified adaptive Cartesian grid system for solving the 2D shallow water equations. *International Journal for Numerical Methods in Fluids*, 69(2):442–458.

Liang, Q., Borthwick, A., Taylor, P., and Huang, J. (2004). Godunov-type quadtree model of species dispersion in shallow flows. In *Shallow Flows: Research Presented at the International Symposium on Shallow Flows, Delft, Netherlands, 2003*, page 439. Taylor & Francis.

Liang, Q. and Borthwick, A. G. (2009). Adaptive quadtree simulation of shallow flows with wet–dry fronts over complex topography. *Computers & Fluids*, 38(2):221–234.

Liang, Q., Du, G., Hall, J. W., and Borthwick, A. G. (2008). Flood inundation modeling with an adaptive quadtree grid shallow water equation solver. *Journal of Hydraulic Engineering*, 134(11):1603–1610.

Liang, Q., Hou, J., and Xia, X. (2015). Contradiction between the C-property and mass conservation in adaptive grid based shallow flow models: cause and solution. *International Journal for Numerical Methods in Fluids*, 78(1):17–36.

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.

Liang, Q. and Smith, L. S. (2015). A high-performance integrated hydrodynamic modelling system for urban flood simulations. *Journal of Hydroinformatics*, 17(4):518–533.

Lin, B., Wicks, J. M., Falconer, R. A., and Adams, K. (2006). Integrating 1D and 2D hydrodynamic models for flood simulation. In *Proceedings of the Institution of Civil Engineers-Water Management*, volume 159, pages 19–25. Thomas Telford Ltd.

Liu, W., Wang, B., Chen, Y., Wu, C., and Liu, X. (2018). Assessing the analytical solution of one-dimensional gravity wave model equations using dam-break experimental measurements. *Water*, 10(9):1261.

Loukili, Y. and Soulaimani, A. (2007). Numerical tracking of shallow water waves by the unstructured finite volume WAF approximation. *International Journal for Computational Methods in Engineering Science and Mechanics*, 8(2):75–88.

Lucy, L. B. (1977). A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, 82:1013–1024.

MacDonald, I. (1996). *Analysis and computation of steady open channel flow*. PhD thesis, University of Reading, Reading, UK.

Mahdizadeh, H., Stansby, P. K., and Rogers, B. D. (2012). Flood wave modeling based on a two-dimensional modified wave propagation algorithm coupled to a full-pipe network solver. *Journal of Hydraulic Engineering*, 138(3):247–259.

Maidment, D. R., Rajib, A., Lin, P., and Clark, E. P. (2016). National water center innovators program summer institute report 2016. *Research Summary*, 4.

Maliar, L. (2015). Assessing gains from parallel computation on a supercomputer. *Economics Bulletin*, 35(1):159–167. Retrieved from
http://hdl.handle.net/10045/50906 [Accessed: May 2020].

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.

Martins, R., Leandro, J., and Djordjević, S. (2015). A well balanced Roe scheme for the local inertial equations with an unstructured mesh. *Advances in Water Resources*, 83:351–363.

Martins, R., Leandro, J., and Djordjević, S. (2018). Wetting and drying numerical treatments for the Roe Riemann scheme. *Journal of Hydraulic Research*, 56(2):256–267.

McCarthy, J. J., Canziani, O. F., Leary, N. A., Dokken, D. J., White, K. S., et al. (2001). *Climate change 2001: impacts, adaptation, and vulnerability: contribution of Working Group II to the third assessment report of the Intergovernmental Panel on Climate Change*, volume 2. Cambridge University Press.

McMillan, H. K. and Brasington, J. (2007). Reduced complexity strategies for modelling urban floodplain inundation. *Geomorphology*, 90(3-4):226–243.

Medeiros, S. C. and Hagen, S. C. (2013). Review of wetting and drying algorithms for numerical tidal flow models. *International journal for numerical methods in fluids*, 71(4):473–487.

Message Passing Interface (2015). Retrieved from
http://en.wikipedia.org/wiki/Message_Passing_Interface. [Accessed: June 2015].

Miglio, E., Perotto, S., and Saleri, F. (2005). Model coupling techniques for free-surface flow problems: Part I. *Nonlinear Analysis: Theory, Methods & Applications*, 63(5-7):e1885–e1896.

Mignot, E., Paquier, A., and Haider, S. (2006). Modeling floods in a dense urban area using 2D shallow water equations. *Journal of Hydrology*, 327(1-2):186–199.

Mohammadnejad, T. and Khoei, A. R. (2013). Hydro-mechanical modeling of cohesive crack propagation in multiphase porous media using the extended finite element method. *International Journal for Numerical and Analytical Methods in Geomechanics*, 37(10):1247–1279.

Mokos, A., Rogers, B. D., and Stansby, P. K. (2017). A multi-phase particle shifting algorithm for SPH simulations of violent hydrodynamics with a large number of particles. *Journal of Hydraulic Research*, 55(2):143–162.

Monaghan, J. J. (1994). Simulating free surface flows with SPH. *Journal of computational physics*, 110(2):399–406.

Morales-Hernández, M., Echeverribar, I., García-Navarro, P., and Brufau, P. (2018). 1D model vs 2D model for flooding events. *EPiC Series in Engineering*, 3:1449–1456.

Morales-Hernández, M., García-Navarro, P., Burguete, J., and Brufau, P. (2013). A conservative strategy to couple 1D and 2D models for shallow water flow simulation. *Computers & Fluids*, 81:26–44.

Morales-Hernández, M., Petaccia, G., Brufau, P., and García-Navarro, P. (2016). Conservative 1D–2D coupled numerical strategies applied to river flooding: The Tiber (Rome). *Applied Mathematical Modelling*, 40(3):2087–2105.

Mosavi, A. and Edalatifar, M. (2018). A hybrid neuro-fuzzy algorithm for prediction of reference evapotranspiration. In *International Conference on Global Research and Education*, pages 235–243. Springer.

Mosavi, A., Ozturk, P., and Chau, K. w. (2018). Flood prediction using machine learning models: Literature review. *Water*, 10(11):1536.

Munson, B. R., Young, D. F., and Okiishi, T. H. (2006). *Fundamentals of fluid mechanics*. Hoboken, NJ: J. Wiley & Sons.

Neal, J., Fewtrell, T., and Trigg, M. (2009). Parallelisation of storage cell flood models using OpenMP. *Environmental Modelling & Software*, 24(7):872–877.

Neal, J., Villanueva, I., Wright, N., Willis, T., Fewtrell, T., and Bates, P. (2012). How much physical complexity is needed to model flood inundation? *Hydrological Processes*, 26(15):2264–2282.

Neal, J. C., Fewtrell, T. J., Bates, P. D., and Wright, N. G. (2010). A comparison of three parallelisation methods for 2D flood inundation models. *Environmental Modelling & Software*, 25(4):398–411.

Nepf, H. (1999). Drag, turbulence, and diffusion in flow through emergent vegetation. *Water resources research*, 35(2):479–489.

Nirupama, N. and Simonovic, S. P. (2007). Increase of flood risk due to urbanisation: a Canadian example. *Natural hazards*, 40(1):25–41.

NOAA (2018). Billion-dollar weather and climate disasters: Overview. NOAA National Center for Environmental Information, 2018. Coastal Elevation Models. Retrieved from
http://www.ngdc.noaa.gov/mgg/coastal/ [Accessed: 2017-2018].

NVIDIA (2015). Retrieved from
http://www.nvidia.co.uk/object/cuda-parallel-computing-uk.html . [Accessed: June 2015].

Néelz, S. and Pender, G. (2010). Benchmarking of 2D hydraulic modelling packages. *SC080035/R2 Environmental Agency*.

Özgen, I., Liang, D., and Hinkelmann, R. (2016a). Shallow water equations with depth-dependent anisotropic porosity for subgrid-scale topography. *Applied Mathematical Modelling*, 40(17-18):7447–7473.

Özgen, I., Teuber, K., Simons, F., Liang, D., and Hinkelmann, R. (2015). Upscaling the shallow water model with a novel roughness formulation. *Environmental Earth Sciences*, 74(11):7371–7386.

Özgen, I., Zhao, J., Liang, D., and Hinkelmann, R. (2016b). Urban flood modeling using shallow water equations with depth-dependent anisotropic porosity. *Journal of Hydrology*, 541:1165–1184.

Park, S., Kim, B., and Kim, D.-H. (2019). 2D GPU-accelerated high resolution numerical scheme for solving diffusive wave equations. *Water*, 11(7):1447.

Pau, J. C. and Sanders, B. F. (2006). Performance of parallel implementations of an explicit finite-volume shallow-water model. *Journal of computing in civil engineering*, 20(2):99–110.

Pelling, H. E., Green, J. A. M., and Ward, S. L. (2013). Modelling tides and sea-level rise: To flood or not to flood. *Ocean Modelling*, 63:21–29.

Platzek, F., Stelling, G., Jankowski, J., Patzwahl, R., and Pietrzak, J. (2016). An efficient semi-implicit subgrid method for free-surface flows on hierarchical grids. *International Journal for Numerical Methods in Fluids*, 80(12):715–741.

Ponce, V. M. (1990). Generalized diffusion wave equation with inertial effects. *Water Resources Research*, 26(5):1099–1101.

Popinet, S. (2011). Quadtree-adaptive tsunami modelling. *Ocean Dynamics*, 61(9):1261–1285.

Prakash, M., Rothauge, K., and Cleary, P. W. (2014). Modelling the impact of dam failure scenarios on flood inundation using SPH. *Applied Mathematical Modelling*, 38(23):5515–5534.

Ransom, O. T. and Younis, B. A. (2016). Explicit GPU based second-order finite-difference modeling on a high resolution surface, Feather River, California. *Water resources management*, 30(1):261–277.

Rao, P. (2005). A parallel RMA2 model for simulating large-scale free surface flows. *Environmental Modelling & Software*, 20(1):47–53.

Ren, B., Wen, H., Dong, P., and Wang, Y. (2016). Improved SPH simulation of wave motions and turbulent flows through porous media. *Coastal Engineering*, 107:14–27.

Richardson, L. F. (1911). The approximate arithmetical solution by finite differences of physical problems involving differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A*, 210(459-470):307–357.

Ritter, A. (1892). The propagation of water waves. *Ver Deutsch ingenieur zeitschr*, 36(33 part 3):947–954.

Roache, P. J. (1994). Perspective: a method for uniform reporting of grid refinement studies. *Journal of Fluids Engineering*, 116:405–413.

Rogers, B., Fujihara, M., and Borthwick, A. G. (2001). Adaptive Q-tree Godunov-type scheme for shallow water equations. *International Journal for Numerical Methods in Fluids*, 35(3):247–280.

Sætra, M. L. and Brodtkorb, A. R. (2010). Shallow water simulations on multiple GPUs. In *International Workshop on Applied Parallel Computing*, pages 56–66. Springer.

Sanders, B. F. and Schubert, J. E. (2019). Primo: Parallel raster inundation model. *Advances in Water Resources*, 126:79–95.

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.

Schubert, J. E. and Sanders, B. F. (2012). Building treatments for urban flood inundation models and implications for predictive skill and modeling efficiency. *Advances in Water Resources*, 41:49–64.

Sehili, A., Lang, G., and Lippert, C. (2014). High-resolution subgrid models: background, grid generation, and implementation. *Ocean Dynamics*, 64(4):519–535.

Shamseldin, A. Y. (2010). Artificial neural network model for river flow forecasting in a developing country. *Journal of Hydroinformatics*, 12(1):22–35.

Shin, G. (2016). *A subgrid approach for unresolved topography in shallow water hydrodynamic modeling*. PhD thesis, University of Texas at Austin, USA.

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(2):375–391.

Smith, J. and Eli, R. N. (1995). Neural-network models of rainfall-runoff process. *Journal of water resources planning and management*, 121(6):499–508.

Soares-Frazão, S., Lhomme, J., Guinot, V., and Zech, Y. (2008). Two-dimensional shallow-water model with porosity for urban flood modelling. *Journal of Hydraulic Research*, 46(1):45–64.

Soares-Frazão, S. and Zech, Y. (2002). Dam break in channels with 90° bend. *Journal of Hydraulic Engineering*, 128(11):956–968.

Song, L., Zhou, J., Li, Q., Yang, X., and Zhang, Y. (2011). An unstructured finite volume model for dam-break floods with wet/dry fronts over complex topography. *International Journal for Numerical Methods in Fluids*, 67(8):960–980.

Stelling, G. S. (2012). Quadtree flood simulations with sub-grid digital elevation models. In *Proceedings of the Institution of Civil Engineers-Water Management*, volume 165, pages 567–580. Thomas Telford Ltd.

Stout, Q. F. (2015). What is parallel computing ? A not too serious explanation. Retrieved from http://web.eecs.umich.edu/~qstout/parallel.html [Accessed: June 2015].

Sugawara, D. and Goto, K. (2012). Numerical modeling of the 2011 Tohoku-oki tsunami in the offshore and onshore of Sendai Plain, Japan. *Sedimentary Geology*, 282:110–123.

Sulaiman, J. and Wahab, S. H. (2018). Heavy rainfall forecasting model using artificial neural network for flood prone area. In *IT Convergence and Security 2017*, pages 68–76. Springer.

Suykens, J. A. K. and Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural processing letters*, 9:293–300.

Sætra, M. L. (2014). *Shallow Water Simulations on Graphics Hardware*. PhD thesis, University of Oslo, Oslo, Norway.

Taherei Ghazvinei, P., Hassanpour Darvishi, H., Mosavi, A., Yusof, K. b. W., Alizamir, M., Shamshirband, S., and Chau, K.-w. (2018). Sugarcane growth prediction based on meteorological parameters using extreme learning machine and artificial neural network. *Engineering Applications of Computational Fluid Mechanics*, 12(1):738–749.

Taormina, R., Chau, K.-W., and Sethi, R. (2012). Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon. *Engineering Applications of Artificial Intelligence*, 25(8):1670–1676.

Teng, J., Jakeman, A. J., Vaze, J., Croke, B. F. W., Dutta, D., and Kim, S. (2017). Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. *Environmental Modelling & Software*, 90:201–216.

Thirumalaiah, K. and Deo, M. C. (1998). River stage forecasting using artificial neural networks. *Journal of Hydrologic Engineering*, 3(1):26–32.

Toro, E. F. (2001). *Shock-capturing methods for free-surface shallow flows*. John Wiley, Chichester, UK.

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.

Tsai, L. T. and Yang, C.-C. (2012). Improving measurement invariance assessments in survey research with missing data by novel artificial neural networks. *Expert Systems with Applications*, 39(12):10456–10464.

Twigt, D. J., De Goede, E. D., Zijl, F., Schwanenberg, D., and Chiu, A. Y. (2009). Coupled 1D–3D hydrodynamic modelling, with application to the Pearl River Delta. *Ocean Dynamics*, 59(6):1077–1093.

UNISDR (2012). United nations international strategy for disaster reduction. Retrieved from http://www.unisdr.org. [Accessed: March 2012].

Vacondio, R., Dal Palù, A., and Mignosa, P. (2014). GPU-enhanced finite volume shallow water solver for fast flood simulations. *Environmental modelling & software*, 57:60–75.

Vacondio, R., Rogers, B. D., Stansby, P. K., and Mignosa, P. (2013). Shallow water SPH for flooding with dynamic particle coalescing and splitting. *Advances in water resources*, 58:10–23.

Valiani, A. and Begnudelli, L. (2006). Divergence form for bed slope source term in shallow water equations. *Journal of Hydraulic Engineering*, 132(7):652–665.

Verwey, A. (2001). Latest developments in floodplain modelling-1D/2D integration. In *6th Conference on Hydraulics in Civil Engineering: The State of Hydraulics; Proceedings*, page 13. Institution of Engineers, Australia.

Viero, D. P. and Valipour, M. (2017). Modeling anisotropy in free-surface overland and shallow inundation flows. *Advances in Water Resources*, 104:1–14.

Villanueva, I. and Wright, N. (2006a). An efficient multi-processor solver for the 2d shallow water equations. In *Proceedings of the 7th International Conference on Hydroinformatics*. HIC, Nice, France.

Villanueva, I. and Wright, N. (2006b). Linking Riemann and storage cell models for flood prediction. In *Proceedings of the Institution of Civil Engineers-Water Management*, volume 159, pages 27–33. Thomas Telford Ltd.

Violeau, D. and Issa, R. (2007). Numerical modelling of complex turbulent free-surface flows with the SPH method: an overview. *International Journal for Numerical Methods in Fluids*, 53(2):277–304.

Violeau, D. and Rogers, B. D. (2016). Smoothed particle hydrodynamics (SPH) for free-surface flows: past, present and future. *Journal of Hydraulic Research*, 54(1):1–26.

Volp, N., Van Prooijen, B., and Stelling, G. (2013). A finite volume approach for shallow water flow accounting for high-resolution bathymetry and roughness data. *Water Resources Research*, 49(7):4126–4135.

Wang, J.-P. and Liang, Q. (2011). Testing a new adaptive grid-based shallow flow model for different types of flood simulations. *Journal of Flood Risk Management*, 4(2):96–103.

Wang, Y. (2011). *Numerical improvements for large-scale flood simulation*. PhD thesis, Newcastle University, Newcastle, UK.

Wang, Y., Liang, Q., Kesserwani, G., and Hall, J. W. (2011a). A 2D shallow flow model for practical dam-break simulations. *Journal of Hydraulic Research*, 49(3):307–316.

Wang, Y., Liang, Q., Kesserwani, G., and Hall, J. W. (2011b). A positivity-preserving zero-inertia model for flood simulation. *Computers & Fluids*, 46(1):505–511.

White, W. R. (2000). Water in rivers: Flooding, a contribution to the world water vision. *IAHR, Madrid*.

Wikipedia (2015). Navier–stokes equations. Retrieved from
http://en.wikipedia.org/wiki/Navier%E2%80%93Stokes_equations. [Accessed: December 2015].

Wu, C. L. and Chau, K. W. (2010). Data-driven models for monthly streamflow time series prediction. *Engineering Applications of Artificial Intelligence*, 23(8):1350–1367.

Wu, G.-f., He, Z.-g., Zhao, L., and Liu, G.-h. (2018). A well-balanced positivity preserving two-dimensional shallow flow model with wetting and drying fronts over irregular topography. *Journal of Hydrodynamics*, 30(4):618–631.

Xia, R. (1994). Impact of coefficients in momentum equation on selection of inertial models. *Journal of hydraulic research*, 32(4):615–621.

Xia, X., Liang, Q., and Ming, X. (2019). A full-scale fluvial flood modelling framework based on a high-performance integrated hydrodynamic modelling system (HiPIMS). *Advances in Water Resources*, 132:103392.

Yeh, G.-T., Shih, D.-S., and Cheng, J.-R. C. (2011). An integrated media, integrated processes watershed model. *Computers & Fluids*, 45(1):2–13.

Ying, X. and Wang, S. S. Y. (2008). Improved implementation of the HLL approximate riemann solver for one-dimensional open channel flows. *Journal of Hydraulic Research*, 46(1):21–34.

Yu, D. and Lane, S. (2007). Coupled modelling of flood inundation over a topographically complex urban floodplain. In *Geophysical Research Abstracts*, volume 9.

Yu, D. and Lane, S. N. (2006). Urban fluvial flood modelling using a two-dimensional diffusion-wave treatment, part 2: development of a sub-grid-scale treatment. *Hydrological Processes*, 20(7):1567–1583.

Yu, D. and Lane, S. N. (2011). Interactions between subgrid-scale resolution, feature representation and grid-scale resolution in flood inundation modelling. *Hydrological Processes*, 25(1):36–53.

Zhang, C., Liang, Q., and Yin, J. (2013). A first-order adaptive solution to rapidly spreading flood waves. *Progress in Computational Fluid Dynamics, An International Journal*, 13(1):1–10.

Zhang, H., Zhou, J., Bi, S., Li, Q., and Fan, Y. (2014). A well-balanced numerical scheme for shallow water simulation on adaptive grids. In *Journal of Physics: Conference Series*, volume 495, page 012049. IOP Publishing.

Zhang, L., Liang, Q., Wang, Y., and Yin, J. (2015). A robust coupled model for solute transport driven by severe flow conditions. *Journal of Hydro-Environment Research*, 9(1):49–60.

Zhang, Y. J. (2020). Assessment of subgrid method in a finite-volume model. *Computers & Mathematics with Applications*.

Zhao, D. H., Shen, H. W., Lai, J. S., and Tabios III, G. Q. (1996). Approximate Riemann solvers in FVM for 2D hydraulic shock wave modeling. *Journal of Hydraulic Engineering*, 122(12):692–702.

Zhou, J. G., Causon, D. M., Mingham, C. G., and Ingram, D. M. (2004). Numerical prediction of dam-break flows in general geometries with complex bed topography. *Journal of Hydraulic Engineering*, 130(4):332–340.

Ziegler, U. (2012). Block-structured adaptive mesh refinement on curvilinear-orthogonal grids. *SIAM Journal on Scientific Computing*, 34(3):C102–C121.

Zienkiewicz, O. C. and Taylor, R. L. (2000). *The Finite Element Method: Volume 3- Fluid Dynamics*. Butterworth Heinemann, London, UK.

Zoppou, C. and Roberts, S. (2000). Numerical solution of the two-dimensional unsteady dam break. *Applied Mathematical Modelling*, 24(7):457–475.