



Upscaling the shallow water equations for fast flood modelling

Journal:	<i>Journal of Hydraulic Research</i>
Manuscript ID	TJHR-2019-0273.R2
Manuscript Type:	Research paper
Date Submitted by the Author:	31-Jul-2020
Complete List of Authors:	SHAMKHALCHIAN, ALIREZA; University of Southampton, Faculty of Engineering and Physical Sciences DE ALMEIDA, GUSTAVO A. M. ; University of Southampton, Faculty of Engineering and Physical Sciences
Keywords:	2D Shallow water equations, finite volume, flooding, nested meshes, solution upscaling, sub-grid
JHR Keywords:	Shallow flows < Environmental Fluid Mechanics, Flood modeling < Applied fluid mechanics and hydraulic engineering, Two- dimensional models < Computational methods in hydro-environment research and fluid dynamics

SCHOLARONE™
Manuscripts

Upscaling the shallow water equations for fast flood modelling

ALIREZA SHAMKHALCHIAN, PhD Student, *Faculty of Engineering and Physical Sciences,
University of Southampton, Southampton, United Kingdom*

Email: as1v14@soton.ac.uk (author for correspondence)

GUSTAVO A. M. DE ALMEIDA, Lecturer, *Faculty of Engineering and Physical Sciences,
University of Southampton, Southampton, United Kingdom*

Email: g.dealmeida@soton.ac.uk

Running Head: Upscaling the shallow water equations.

Upscaling the shallow water equations for fast flood modelling

ABSTRACT

This paper presents a new sub-grid flood inundation model aimed at high computational performance. The model solves the two-dimensional shallow water equations (SWE) by a Godunov-type finite volume (FV) method that uses two nested meshes. Runtime computations are performed at a coarse computational mesh, while a fine mesh is used to incorporate fine resolution information into the solution at pre-processing level. New upscaling methods 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. The accuracy and performance of the model are tested through artificial and real-world test problems. Results showed that *i*) for the same computational (coarse mesh) resolution, the inclusion of sub-grid information delivers more accurate results than a single-mesh FV model and *ii*) for the same accuracy and at low resolution, the proposed methods improve computational performance.

Keywords: 2D shallow water equations; finite volume; flooding; nested meshes; solution upscaling; sub-grid.

1 Introduction

Computational models based on the two-dimensional (2D) 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. Liang & Borthwick, 2009; Liang & Marche, 2009; Hou et al., 2015). 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. Namely, while they capture complex flow features accurately when a fine resolution mesh is used, this accuracy degrades significantly at coarse resolutions. Currently one of the main challenges to simulating large-scale flood inundation is the computational time 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, most models currently available –and in particular those that do not rely on parallelization– 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).

1
2
3 Substantial research efforts have been devoted to reduce the computational limitations of
4 inundation models through the development of modelling techniques that offer different trade-offs
5 between computational speed and accuracy. These techniques involve, for example, the use of parallel
6 computing (Neal et al., 2009; Park et al., 2019), adaptive meshes (e.g. Liang, 2012; Hoch et al., 2018),
7 coupled 1D-2D models (e.g. Morales-Hernández et al., 2016) and simplified SWE models (e.g. Horrit
8 & Bates, 2001; Wang et al., 2011). One such approach that is particularly interesting because of its
9 generality (i.e. it can be used *in tandem* with other techniques) is the use of sub-grid parameterizations
10 of the governing equations to approximate part of the physics operating within a large computational
11 cell, so that the accuracy of simulations performed at a coarse resolution is improved. Because the
12 computational cost t_c of explicit numerical schemes for the solution of the 2D SWE is very sensitive
13 to the grid size [i.e., typically, $t_c \sim \Delta x^{-3}$] grid coarsening can deliver outstanding model speedups. For
14 example, an increase in cell size by a factor of 10 would typically translates into a 1000x model
15 speedup. However, simple grid coarsening also degrades the accuracy of numerical results, as a result
16 of the growth of truncation errors. Of particular importance to the accuracy of such models is the
17 misrepresentation of topographical details that are of primary importance to the propagation of the
18 flood wave (Yu & Lane, 2006; Begnudelli et al., 2008).

19
20
21 A few grid-coarsening techniques have been developed in the last decades that incorporate
22 important information at sub-grid scale –in particular fine-scale topographical data– into the model’s
23 solution procedure (e.g. Bates, 2000; McMillan & Brasington, 2007; Sanders et al., 2008) via the
24 concept of porosity (Defina et al., 1994). Namely, volumetric and area porosity parameters have been
25 incorporated into the governing equations to account for the fact that *i*) parts of computational cells
26 may be blocked and not contribute to storage of mass and momentum, and *ii*) that only a fraction of
27 cell edges may contribute to fluxes (Sanders et al., 2008; Kim et al., 2015). Since first proposed by
28 Defina et al. (1994), the use of porosity parameters to represent partially-wet cells has received
29 increased attention within the literature (Defina, 2000; Guinot & Soares-Frazão, 2006; Soares-Frazão
30 et al., 2008; Sanders et al., 2008; Cea & Vázquez-Cendón, 2010; Yu & Lane, 2011; Guinot, 2012;
31 Schubert & Sanders, 2012; Huang et al., 2014; Kim et al., 2015; Özgen et al., 2016a, b; Guinot, 2017;
32 Bruwier et al., 2017; Guinot et al., 2017). Porosity models have mainly focused on the representation
33 of buildings and other types of flow blockages that are typically found in urban areas (e.g. Soares-
34 Frazão et al., 2008; Guinot, 2012). More sophisticated versions of this concept have also been
35 developed to include anisotropy of the porosity parameters, which stems from the arrangement of
36 obstacles within the computational cells (Sanders et al., 2008; Guinot, 2012).

37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
Despite their main advantages over simple (i.e. non-porosity) models, porosity models have a
number of shortcomings. 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 (i.e. non-prismatic features) requires the definition of depth-
dependent porosity parameters, (Özgen et al., 2016a, b; Guinot et al., 2018). Second, while porosity

1
2
3 may correctly model part of the physics (in particular, storage capacity), other processes that are
4 influenced by the irregular topography such as friction are not explicitly represented. This is because
5 the conserved variables are averaged over the cell to become the arguments of a nonlinear flow
6 resistance equation. Coarse computational cells may also display large variations in roughness, which
7 may not be accurately modelled through simple averaging. A similar issue occurs when fluxes are
8 estimated at the edges of two neighbouring computational cells. When large variations of topography
9 occur at these edges, fluxes estimated by Riemann solvers from averaged quantities may be highly
10 inaccurate. Also, the effects of bed slope computed based on micro-topography or an averaged bed
11 level (or the level of the terrain at the centroid of a cell) may differ substantially.
12
13
14
15

16 In this paper a new sub-grid model is proposed, which addresses some of the main
17 shortcomings of other existing sub-grid methods –in particular porosity models, which constitute the
18 main class of the sub-grid approach. The model is based on the integral form of 2D shallow water
19 equations and uses two nested meshes. The governing equations are solved at the coarse mesh,
20 leading to significant model speedups. The solution is implemented through the Godunov finite
21 volume method. A high-resolution mesh nested within the coarse grid is used to improve the accuracy
22 of the coarse resolution model. Namely, this is achieved by defining parameters that represent, either
23 exactly or approximately, the effects of finely resolved sub-grid topography on storage, momentum
24 and friction. The proposed model has a number of advantages over current generation porosity-based
25 models, as follows. First, although porosity parameters are not explicitly defined in the model, the
26 formulation resolves porosity effects (e.g. the effect of topography on storage) and anisotropy in a
27 more generalised way that includes the intrinsic dependency on the free surface elevation. That is, the
28 effects of porosity and anisotropy are not constant in time during the propagation of a flood wave.
29 Second, the proposed model implements a new technique to upscale the effects of fine-scale
30 distribution of bed elevations and roughness on friction and fluxes across edges between cells.
31 Finally, the model improves the solution of the governing equations at partially-wet cells using sub-
32 grid, fine resolution data. Most of the above improvements require additional computational efforts
33 only at pre-processing level.
34
35
36
37
38
39
40
41
42

43 The model presented in this paper bears close similarities to other sub-grid models recently
44 proposed in the literature. For example, Volp et al. (2013), Hénonin et al. (2015) and Sanders and
45 Schubert (2019) also used dual-resolution nested computational meshes, although their proposed
46 methods to upscale each of the terms in the SWE are different to those presented in this paper.
47 Examples of upscaling methods used previously include simple averaging of depths (Hénonin et al.,
48 2015), modelling the distribution of velocity according to a resistance law under the assumption of
49 constant energy slope for the friction source term (Sanders & Schubert, 2019), or a combination of
50 both (Volp et al., 2013). In this paper, we define upscaling methods for each of the terms in the SWE
51 based on integration of the governing equations over the coarse computational cell. The effects of
52
53
54
55
56
57
58
59
60

sub-grid modelling choices adopted here are discussed in light of and compared against methods previously proposed.

The remaining of the paper is structured as follows. Sections 2 and 3 briefly describe the structure of the nested meshes and the governing equations used by the proposed sub-grid model, respectively. Section 4 describes the methods used to upscale each of the terms in the SWE to coarse resolution. In section 5, the proposed model is evaluated through a number of artificial and real-world test cases and the performance of the sub-grid model is discussed both in terms of accuracy and speedup. The main conclusions of the work are synthesized in the last section.

2 The nested meshes sub-grid model approach

The governing equations are solved using two nested meshes, coarse and fine, as shown in Fig. 1. Cells defined at 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. At each small cell, information such as the bed elevation and roughness coefficient are defined as constant. Therefore, within a large cell, both the topography and the roughness coefficient are defined as piecewise constant functions. In this paper, 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 & Schubert, 2019).

Figure 1 illustrates the nested mesh and some key 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 size of the rectangular computational domain. ΔX , ΔY , δx and δy are the dimensions of the large and small cells, respectively and z is the elevation of the bed (hereafter 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 , while the position of a small cell inside this large cell is represented by subscripts j and k (column and row, respectively), (see Fig. 1). As an example $z_i|_{j,k}$ represents the bed level at j^{th} column and k^{th} row of small cells inside the i^{th} large cell.

3 Governing equations and time integration

The proposed model solves the integral form of the two-dimensional shallow water equations

$$\frac{\partial}{\partial t} \int_{\Omega} \mathbf{U} d\Omega + \oint_{\Gamma} [\mathbf{E}(\mathbf{U}) \cdot \mathbf{e}] d\Gamma = \int_{\Omega} \mathbf{S}(\mathbf{U}) d\Omega, \quad (1)$$

where

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

$$\mathbf{S}(\mathbf{U}) = \mathbf{S}_b(\mathbf{U}) + \mathbf{S}_f(\mathbf{U}), \quad \mathbf{S}_b(\mathbf{U}) = \begin{bmatrix} 0 \\ ghs_{0x} \\ ghs_{0y} \end{bmatrix}, \quad \mathbf{S}_f(\mathbf{U}) = \begin{bmatrix} 0 \\ -ghs_{fx} \\ -ghs_{fy} \end{bmatrix}, \quad (3)$$

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, $\mathbf{e} = [e_x, e_y]$ is the unit vector normal to Γ , s_{fx} , s_{fy} , s_{0x} , s_{0y} , 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 $s_{fx} = \frac{u\|\mathbf{V}\|n_M^2}{h^{4/3}}$, $s_{fy} = \frac{v\|\mathbf{V}\|n_M^2}{h^{4/3}}$, where u and v are x and y velocity components and $\|\mathbf{V}\| = \sqrt{u^2 + v^2}$ is the magnitude of the velocity vector. It should be noted that the conserved variable h in the vector $\mathbf{U}(x,y,t)$ is replaced by 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, i.e.

$$\bar{\mathbf{U}} = \frac{1}{\Omega} \int_{\Omega} \mathbf{U} d\Omega. \quad (4)$$

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

$$\bar{\mathbf{U}}_i^* = \bar{\mathbf{U}}_i^n - \frac{\Delta t}{\Omega_i} \left\{ \sum_{m=1}^4 \left[\phi_{\Gamma_m} \mathbf{E}(\mathbf{U}_i^n) \cdot \mathbf{e}_m d\Gamma_m \right] + \int_{\Omega_i} \mathbf{S}_b(\mathbf{U}_i^n) d\Omega_i \right\}, \quad (5)$$

where, $\bar{\mathbf{U}}_i^*$ 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:

$$\frac{d}{dt} \int_{\Omega_i} \mathbf{U}_i^* d\Omega = \int_{\Omega_i} \mathbf{S}_f(\mathbf{U}_i^*, n+1) d\Omega, \quad (6)$$

which in differential form reads as

$$\frac{d}{dt} \mathbf{U}_i^* = \mathbf{S}_f(\mathbf{U}_i^*, n+1). \quad (7)$$

Godunov type finite volume techniques for the solution of Eq. (5) estimate the first integral using Riemann solvers with values of \bar{U}_i^n (i.e., averaged values) reconstructed on each side of the edges of the cell, while the second integral is typically solved as $\mathcal{S}_b(\bar{U}_i^n)d\Omega_i$, (e.g. Valiani & Begnudelli, 2006; Liang & Marche, 2009; Liang, 2010). At fine grid resolutions, this approach typically results in good accuracy, but since the objective of the current paper is to solve Eqs 5 and 6 at coarse resolutions, an improved method needs to be adopted. The following sections describe the methods proposed in this paper to approximate each of the integrals in Eqs 5 and 6.

To prevent numerical instabilities, the time step used in the solution of Eq. (5) must be restricted by the Courant–Friedrichs–Lewy (CFL) condition (see e.g. Toro, 2001; Leveque, 2002),

$$\Delta t = \min \left(\frac{\Delta_X \text{CFL}}{\frac{|\bar{q}_{x_i}^n|}{\bar{h}_i^n} + \sqrt{g\bar{h}_i^n}}, \frac{\Delta_Y \text{CFL}}{\frac{|\bar{q}_{y_i}^n|}{\bar{h}_i^n} + \sqrt{g\bar{h}_i^n}} \right), \quad \text{CFL} < 1, \quad N = \text{number of large cells} \quad (8)$$

4 Upscaling the shallow water equations

4.1 Solving the homogeneous part of the governing equations

The proposed model uses a Godunov-type finite volume numerical 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 Riemann problem at each interface between small cells located at the edges of large cells, while 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). Figure 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 $h = \max(\eta - z, 0)$, where η is assumed constant across 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, against (ii) a first order accurate scheme adopting 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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

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 (q_x, q_y) conserved variables at the interface between two large cells.

4.2 Flow reconstruction at the edges of two neighbouring large cells

Equation (5) computes the spatially averaged values of \bar{U}^* from the values of \bar{U}^n . In order to approximate the fluxes across large cells accurately, values of \bar{U}^n need to be downscaled to define the values of η and $\mathbf{q} = [q_x, q_y]$ at small cells interface. Sanders and Schubert (2019) approached this problem by adopting constant values of η and \mathbf{q} on both sides of the edges of a large cell. Given that $V = \mathbf{q}/h$, such a distribution of variables implies that flow velocities are high at shallow parts of the edge (indeed, as $\eta \rightarrow z$, $V \rightarrow \infty$) and low at deep areas; which is contrary to what is commonly observed for shallow water flows. In this paper we propose an alternative heuristic method to downscale spatially averaged conserved variables \bar{q}_x^n and \bar{q}_y^n to the fine resolution grid at the interface between two large cells.

Figure 3 shows an example of a cross-section along the edge Γ_4 of i^{th} large cell. The values of water depth and Manning's roughness coefficient of the small cells vary along the cross-section. The model distributes the values of $\bar{q}_{x_i}^n$ and $\bar{q}_{y_i}^n$ along the edge Γ_4 only for the wet small cells, the number of which (N_{Γ_4}) is a function of η_i^n and the high-resolution bed elevations. 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., 2007). Using Manning's relation for the x component of the unit discharge,

$$q_{x_i}^n|_{1,k} = \left[\left(\frac{1}{n_M} h^{5/3} \right)_{1,k} (\sqrt{s_f} \varepsilon_x)_{\Gamma_4} \right]_i^n \quad 1 \leq k \leq N_{\Gamma_4}, \quad (9)$$

where, $\varepsilon_{x_i}^n$ is the x -component of the unit vector parallel to \mathbf{q} at i^{th} large cell and time level n , and $s_{f_i}^n|_{\Gamma_4}$ 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 $\tilde{q}_{x_i}^n|_{\Gamma_4}$) is by definition given by

$$\tilde{q}_{x_i}^n|_{\Gamma_4} = \left[\frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} (q_{x_i}^n|_{1,k}) \right]_i^n \quad (10)$$

or alternatively,

$$\tilde{q}_{x_i}^n|_{\Gamma_4} = \left[(\psi_e \sqrt{s_f} \varepsilon_x)_{\Gamma_4} \right]_i^n, \quad (11)$$

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

$$\psi_{ei}^n|_{\Gamma_4} = \left[\frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \left(\frac{h^{5/3}}{nm} \right)_{1,k} \right]_i^n \quad (12)$$

Downscaling the values of q through Eq. (10) requires the computation of ψ_e through Eq. (12), which can be computationally inefficient at fine grid resolutions since N_{Γ_4} terms need to be computed in Eq. (12). Much of this cost can be reduced by the following method, which transfers most of this computational burden to the pre-processing. The method computes the values of ψ_e for different values of η_i at pre-processing level and recovers them from tables at run time. Such method requires knowledge of the maximum value of η_i^n attained during simulations, which is unknown *a priori*. Setting a maximum η_i^n 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 Taylor series expansion centred about a point $\tilde{h}_i^n|_{\Gamma_4}$ (the averaged 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 $(h_i^n)^{5/3}$ is written as

$$(h_i^n)^{5/3} = \left[(\tilde{h})_{\Gamma_4}^{5/3} + \frac{5}{3}(\tilde{h})_{\Gamma_4}^{2/3}(\Delta h)_{1,k} + \frac{5}{9}(\tilde{h})_{\Gamma_4}^{-1/3}(\Delta h)_{1,k}^2 + O(\Delta h)_{1,k}^3 \right]_i^n, \quad (13)$$

where, $(\Delta h_i)_{1,k}^n = (h|_{1,k} - \tilde{h}|_{\Gamma_4})_i^n$ and $\tilde{h}_i|_{\Gamma_4}$ is 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

$$\psi_{ai}^n|_{\Gamma_4} = \left\{ \frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \frac{1}{nm|_{1,k}} \left[(\tilde{h})_{\Gamma_4}^{5/3} + \frac{5}{3}(\tilde{h})_{\Gamma_4}^{2/3}(\Delta h)_{1,k} + \frac{5}{9}(\tilde{h})_{\Gamma_4}^{-1/3}(\Delta h)_{1,k}^2 \right] \right\}_i^n, \quad (14)$$

since, $\eta_i^n = h_i|_{j,k}^n + z_i|_{j,k}$ is assumed constant, then $(\Delta h_i)_{1,k}^n = (h|_{1,k} - \tilde{h}|_{\Gamma_4})_i^n = (\tilde{z}^n|_{\Gamma_4} - z|_{1,k})_i = (\Delta z_i)_{1,k}^n$

where $\tilde{z}_i^n|_{\Gamma_4}$ is the average of wet small cells bed levels at the cross-section and time level n . Then,

$$\psi_{ai}^n|_{\Gamma_4} = \left\{ \frac{1}{N_{\Gamma_4}} \sum_{k=1}^{N_{\Gamma_4}} \frac{1}{nm|_{1,k}} \left[(\eta - \tilde{z})_{\Gamma_4}^{5/3} + \frac{5}{3}(\eta - \tilde{z})_{\Gamma_4}^{2/3}(\Delta z)_{1,k} + \frac{5}{9}(\eta - \tilde{z})_{\Gamma_4}^{-1/3}(\Delta z)_{1,k}^2 \right] \right\}_i^n, \quad (15)$$

$$\psi_{ai}^n|_{\Gamma_4} = \frac{1}{N_{\Gamma_4}} \left\{ (\eta - \tilde{z})_{\Gamma_4}^{5/3} \sum_{k=1}^{N_{\Gamma_4}} \left(\frac{1}{nm} \right)_{1,k} + \frac{5}{3}(\eta - \tilde{z})_{\Gamma_4}^{2/3} \sum_{k=1}^{N_{\Gamma_4}} \left(\frac{\Delta z}{nm} \right)_{1,k} + \frac{5}{9}(\eta - \tilde{z})_{\Gamma_4}^{-1/3} \sum_{k=1}^{N_{\Gamma_4}} \left[\frac{(\Delta z)^2}{nm} \right]_{1,k} \right\}_i^n, \quad (16)$$

or

$$\psi_{ai}^n|_{\Gamma_4} = \left\{ \left[(\eta - \tilde{z})_{\Gamma_4}^{5/3} T_a + \frac{5}{3}(\eta - \tilde{z})_{\Gamma_4}^{2/3} T_b + \frac{5}{9}(\eta - \tilde{z})_{\Gamma_4}^{-1/3} T_c \right]_{\Gamma_4} \right\}_i^n, \quad (17)$$

where, $T_{ai}^n|_{\Gamma_4}$, $T_{bi}^n|_{\Gamma_4}$ and $T_{ci}^n|_{\Gamma_4}$ are defined as

$$T_{ai}^n|_{\Gamma_4} = \left\{ \frac{1}{N_{\Gamma_4}} \sum_{k=1}^{k=N_{\Gamma_4}} \left(\frac{1}{n_M} \right)_{1,k} \right\}_i^n, \quad (18)$$

$$T_{bi}^n|_{\Gamma_4} = \left\{ \frac{1}{N_{\Gamma_4}} \sum_{k=1}^{k=N_{\Gamma_4}} \left[\frac{(\tilde{z}|_{\Gamma_4} - z)}{n_M} \right]_{1,k} \right\}_i^n, \quad (19)$$

$$T_{ci}^n|_{\Gamma_4} = \left\{ \frac{1}{N_{\Gamma_4}} \sum_{k=1}^{k=N_{\Gamma_4}} \left[\frac{(\tilde{z}|_{\Gamma_4} - z)^2}{n_M} \right]_{1,k} \right\}_i^n, \quad (20)$$

Equations (11) and (12) and the assumptions that $\tilde{q}_{xi}^n|_{\Gamma_4} = \bar{q}_{xi}^n$ and $\psi_{ai}^n|_{\Gamma_4} \cong \psi_{ei}^n|_{\Gamma_4}$ lead to the following approximate expression for the friction slope:

$$[(\sqrt{S_f} \varepsilon_x)_{\Gamma_4}]_i^n = \left(\frac{\bar{q}_x}{\psi_a|_{\Gamma_4}} \right)_i^n. \quad (21)$$

The downscaled value of the unit discharge is then obtained by

$$q_{xi}^n|_{1,k} = \left[\left(\frac{1}{n_M} h^{5/3} \right)_{1,k} \frac{\bar{q}_x}{(\psi_a)_{\Gamma_4}} \right]_i^n \quad 1 \leq k \leq N_{\Gamma_4}^n, \quad (22)$$

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

A few important remarks must be made about Eq. (22). First, the parameters T_a to T_c are computed based on the values of bed elevations and roughness of wet small cells only. Therefore, these values are not constant during a simulation, since the number of wet cells depend 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, since 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).

4.3 Source terms

Bed slope source term

The model approximation of the bed slope source term in Eq. (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

$$ghs_{0x} = \frac{\partial}{\partial x} \left(\frac{1}{2} gh^2 \right). \quad (23)$$

Substituting Eq. (23) and the expression $h(x,y) = \eta - z(x,y)$ into the last term of Eq. (5) yields:

$$-\frac{\Delta t}{\Omega_i} \left(\int_{\Omega_i} ghs_{0x} d\Omega \right)_i^n = \frac{-g \Delta t}{2\Omega_i} \left\{ \int_{y|_{r_1}}^{y|_{r_3}} [(\eta - z|_{r_2})^2 - (\eta - z|_{r_4})^2] dy \right\}_i^n. \quad (24)$$

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 condition can be modelled by considering only the wet sub-region of the cell. This is achieved in the model by substituting the variable $\hat{z} = \min(z, \eta)$ for z in Eq. (24). Figure 4 illustrates how this change of variable translates into integration over the wet sub-domain of a large cell only. For instance, in Fig. 4a, the cell is fully wet and $\hat{z} = z$. On the other hand, the left edge of the cell shown in Fig. 4b is unsubmerged, and the change to $\hat{z} = \eta$ effectively moves the boundary of integration to the wet/dry front. Using \hat{z} in Eq. (24), expanding the quadratic terms and substituting $K = \Delta Y / \delta y$ yields:

$$-\frac{\Delta t}{\Omega_i} \left(\int_{\Omega_i} ghs_{0x} d\Omega \right)_i^n = \frac{-g \Delta t}{2\Delta XK} \sum_{k=1}^{k=K} \left[2\Delta \hat{z}_{x|k} (\eta - \hat{z}|_k) \right]_i^n. \quad (25)$$

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

Equation (25) shows that the bed slope source term is a function of water surface (η), cell sizes and topographic data on 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 for the bed slope source term in the y direction.

Friction source term

Equation (7) is discretised based on a widely tested semi-implicit scheme (e.g. Liang & Marche, 2009; Kesserwani & Liang, 2012; Cea & Bladé, 2015; de Almeida et al., 2018). While the new scheme by Xia and Liang (2018) offers a fully implicit method, the earlier scheme (Liang & Marche, 2009) is adopted here because given its simpler structure, it results in a more compact form when adapted to the nested-mesh used in this paper. 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. (7), and ignoring the first (zero-valued) component of the friction source term vector results in:

$$\frac{d}{dt}(\mathbf{q}_{j,k}^*)_i = \left\{ -g \left[n_M^2 \mathbf{q}^{n+1} \|\mathbf{q}^*\| / (h^*)^{7/3} \right]_{j,k} \right\}_i, \quad (26)$$

where j,k represents the location of a small cell inside the i^{th} large cell (see Fig. 1) and $\|\mathbf{q}_{j,k}^*\|$ is the magnitude of $\mathbf{q}_{j,k}^*$. It is assumed here that while \mathbf{q} varies over a large cell, its direction is constant. Therefore, $\mathbf{q}_{j,k}(x,y)|_i = [\Phi_{j,k}(x,y)\bar{\mathbf{q}}]_i$, where, $\Phi(x,y)$ is the coefficient of unit width discharge distribution inside the large cell. Substituting this relation into Eq. (26) yields:

$$\frac{d}{dt}(\Phi^* |_{j,k} \bar{\mathbf{q}}^*)_i = \left\{ -g \bar{\mathbf{q}}^{n+1} \|\bar{\mathbf{q}}^*\| [\Phi^* \Phi^{n+1} n_M^2 / (h^*)^{7/3}]_{j,k} \right\}_i. \quad (27)$$

Further, it is assumed that $\frac{d}{dt}(\Phi^* |_{j,k})_i$ is negligible. This means $\left[\bar{\mathbf{q}}^* \frac{d(\Phi^*)_{j,k}}{dt} \right]_i \ll \left[(\Phi^*)_{j,k} \frac{d\bar{\mathbf{q}}^*}{dt} \right]_i$ and $(\Phi^* |_{j,k})_i \approx (\Phi^{n+1} |_{j,k})_i$, which yields:

$$\left[\frac{(h^*)^{7/3}}{n_M^2} \right]_{j,k} \frac{d}{dt}(\bar{\mathbf{q}}^*)_i = \left[-g \bar{\mathbf{q}}^{n+1} \|\bar{\mathbf{q}}^*\| (\Phi^{n+1})_{j,k} \right]_i, \quad (28)$$

Integrating Eq. (28) over the wet fraction of the large cell:

$$\int_{\Omega_{w_i}} \left[\frac{(h^*)^{7/3}}{n_M^2} \right]_{j,k} \frac{d}{dt}(\bar{\mathbf{q}}^*) d\Omega_w = \int_{\Omega_{w_i}} -g \bar{\mathbf{q}}^{n+1} \|\bar{\mathbf{q}}^*\| (\Phi^{n+1})_{j,k} d\Omega_w, \quad (29)$$

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

$$\frac{d}{dt}(\bar{\mathbf{q}}^*)_i \int_{\Omega_{w_i}} \left[\frac{(h^*)^{7/3}}{n_M^2} \right]_{j,k} d\Omega_w = -g \bar{\mathbf{q}}^{n+1} \|\bar{\mathbf{q}}^*\| \int_{\Omega_{w_i}} (\Phi^{n+1})_{j,k} d\Omega_w. \quad (30)$$

Since $\bar{\mathbf{q}}_i^{n+1} = \frac{\int_{\Omega_{w_i}} \mathbf{q}_{j,k}^{n+1} d\Omega_w}{\Omega_{w_i}} = \frac{\int_{\Omega_{w_i}} (\Phi^{n+1})_{j,k} \bar{\mathbf{q}}^{n+1} d\Omega_w}{\Omega_{w_i}}$, it follows that $\int_{\Omega_{w_i}} (\Phi^{n+1})_{j,k} d\Omega_w = \Omega_{w_i}$, and

$$\frac{d}{dt}(\bar{\mathbf{q}}^*)_i = \left[-g \bar{\mathbf{q}}^{n+1} \|\bar{\mathbf{q}}^*\| / \left(\frac{(h^*)^{7/3}}{n_M^2} \right)_i \right]^*. \quad (31)$$

where

$$\left(\frac{(h^*)^{7/3}}{n_M^2} \right)_i^* = \frac{1}{\Omega_{w_i}} \int_{\Omega_{w_i}} \left[\frac{(h^*)^{7/3}}{n_M^2} \right]_{j,k} d\Omega_w = \left[\frac{1}{N_w} \sum_{j=1}^J \sum_{k=1}^K \left(\frac{(h^*)^{7/3}}{n_M^2} \right)_{j,k} \right]_i^*, \quad (32)$$

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

Discretising the derivative in Eq. (31) using the first order forward-time finite-difference yields:

$$\bar{q}_i^{n+1} = \left[\frac{\bar{q}}{1 + g \|\bar{q}\| \Delta t / \left(\frac{h^{7/3}}{n_M^2} \right)_i} \right]^* \quad (33)$$

Equation (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. (32) is derived here which follows an approach similar to that described in Section 4.2 for the reconstruction of q at the edges of large cells. First, $(h_i^*)^{7/3}$ is approximated using Taylor series centered about \bar{h}_i^* (the mean water depth over the submerged area of the large cell):

$$(h_i^*)^{7/3} = \left[(\bar{h})^{\frac{7}{3}} + \frac{7}{3}(\bar{h})^{\frac{4}{3}}(h - \bar{h}) + \frac{14}{9}(\bar{h})^{\frac{1}{3}}(h - \bar{h})^2 + O(h - \bar{h})^3 \right]_i^* \quad (34)$$

Neglecting terms of order higher than two and substituting Eq. (34) into Eq. (32) yields:

$$\left(\frac{h^{7/3}}{n_M^2} \right)_i^* = \left\{ \frac{1}{N_w} \sum_{j=1}^J \sum_{k=1}^K \left[\frac{(\bar{h})^{\frac{7}{3}} + \frac{7}{3}(\bar{h})^{\frac{4}{3}}(h_{j,k} - \bar{h}) + \frac{14}{9}(\bar{h})^{\frac{1}{3}}(h_{j,k} - \bar{h})^2}{n_M^2|_{j,k}} \right] \right\}_i^* \quad (35)$$

Substituting $h_i^*|_{j,k} = (\eta^* - z|_{j,k})_i$, and $\bar{h}_i^* = \eta_i^* - \bar{z}_i^*$ (where \bar{z}_i^* is the mean bed level of wet area of the large cell) into Eq. (35) leads to

$$\left(\frac{h^{7/3}}{n_M^2} \right)_i^* = \left\{ \frac{1}{N_w} \sum_{j=1}^J \sum_{k=1}^K \left[\frac{(\eta - \bar{z})^{\frac{7}{3}} + \frac{7}{3}(\eta - \bar{z})^{\frac{4}{3}}(\bar{z} - z|_{j,k}) + \frac{14}{9}(\eta - \bar{z})^{\frac{1}{3}}(\bar{z} - z|_{j,k})^2}{n_M^2|_{j,k}} \right] \right\}_i^* \quad (36)$$

or alternatively:

$$\left(\frac{h^{7/3}}{n_M^2} \right)_i^* = \left[(\eta - \bar{z})^{\frac{7}{3}} T_d + \frac{7}{3}(\eta - \bar{z})^{\frac{4}{3}} T_e + \frac{14}{9}(\eta - \bar{z})^{\frac{1}{3}} T_f \right]_i^* \quad (37)$$

where

$$T_{di}^* = \left\{ \frac{1}{N_w} \sum_{j=1}^J \sum_{k=1}^K \left[\frac{1}{n_M^2} \right]_{j,k} \right\}_i^* \quad (38)$$

$$T_{ei}^* = \left\{ \frac{1}{N_w} \sum_{j=1}^J \sum_{k=1}^K \left[\frac{(\bar{z} - z)}{n_M^2} \right]_{j,k} \right\}_i^* \quad (39)$$

$$T_{fi}^* = \left\{ \frac{1}{N_w} \sum_{j=1}^J \sum_{k=1}^K \left[\frac{(\bar{z} - z)^2}{nm^2} \right]_{j,k} \right\}_i^* \quad (40)$$

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. (37) into Eq. (33), which yields:

$$\bar{q}_i^{\eta+1} = \bar{q}_i^* / \sigma_i^*, \quad \sigma_i^* = \left[1 + \frac{g \|\bar{q}\| \Delta t}{(\eta - \bar{z})^3 T_d + \frac{7}{3} (\eta - \bar{z})^3 T_e + \frac{14}{9} (\eta - \bar{z})^3 T_f} \right]_i^* \quad (41)$$

4.4 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 \bar{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 efforts 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 sub-grid model (herein denoted by SG) 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). At 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, 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 section, pre-processing took no longer than 0.1% of the simulation time. Figure 5 represents the structure of the model, illustrating how the pre-processing links to other components of the model. The SG algorithm was implemented in C++ programming language and the executable used in this paper was compiled with Intel.

5 Test cases

In this section, the accuracy and computational performance of the sub-grid model presented in the previous sections are tested through a set of stringent idealised and real-world test cases. This is done by comparing the results of the SG against the so-called 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 solved for a single mesh. The 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.

5.1 Test Case 1

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. Since the channel and bend are represented only at fine-resolution, this problem tests the ability of the proposed upscaling methods to capture the influence of sub-grid scale topography on the flow. Figure 6 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 substantially coarser than the width of the channel. The results of this simulation are compared against those obtained with the T model runs with resolutions of 2, 4, 8 and 16 m. The bed elevation in the white small cells (main channel) is $z_c = 0$ m and two scenarios are simulated in which the elevation of green cells (floodplains) is set to $z_f = 1.0$ m and $z_f = 1.5$ m. The value of Manning's coefficient $n_M = 0.01 \text{ s m}^{-1/3}$ is constant across the domain. Initially, the domain is entirely dry. A constant discharge $Q = 1.6 \text{ m}^3 \text{ s}^{-1}$ is uniformly introduced along the edges of white cells on the left side of the domain and a constant water depth 0.5 m is set at the edges of the white cells at the lowest 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 was set to 0.75.

Figure 7 presents the longitudinal profiles of water surface elevation obtained along the red line shown in Fig. 6. These profiles are presented for the different model types, resolutions and two floodplain elevations (i.e. the green area scenarios in Fig. 6) of 1m (Fig. 7a) and 1.5 m (Fig. 7b). 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 the two floodplain elevations differ substantially. Namely, while for $z_f = 1.0$ m the elevation of free surface

1
2
3 is less than 1.0 m for all resolutions tested, the increase in floodplain height to $z_f = 1.5$ m results in
4 substantially higher values (~50% higher for the coarser resolution). The reason for these different
5 solutions is related to how the bed elevation was assigned to these cells. The coarse resolution cell
6 elevation was obtained from averaging of 2 m resolution bed elevation over the area of the coarsened
7 cell. Therefore, in the coarsened grid the elevation of the region representing the channel depends on
8 the values of z_f and z_c . The results in Fig. 7 also illustrate how the accuracy of solutions is reduced, as
9 the grid is made coarser for simulations performed with a traditional mode. On the other hand, results
10 obtained with the sub-grid model at resolution 16/2 are very close to the results of the finely resolved
11 traditional model. The improved accuracy of the SG model over the traditional model is evidenced by
12 the Root Mean Square Error (RMSE) = 0.05 m for SG (16/2) against RMSE = 0.21 m and RMSE =
13 0.51 m of T (16) at floodplain scenarios of $z_f = 1.0$ m and $z_f = 1.5$ m, respectively. Since all
14 simulations in this test case involve only a small number of cells, runtime was extremely short and is
15 not analysed here.

22 23 24 5.2 Test Case 2

25
26 The second test case is aimed at assessing the performance of the developed sub-grid model for the
27 solution of problems involving relatively large variation of topography, including areas of adverse
28 (negative) bed slope. This variation of depths and velocities within computational cells is a
29 particularly stringent test for the estimation of the friction source term by the upscaling method
30 proposed in section 4.3. The test case was created using the method described by Macdonald (1996).
31 The technique uses an inverse solution for the one-dimensional steady non-uniform flow whereby the
32 bed longitudinal profile corresponding to the pre-defined depth profile is obtained. A 10 m wide, 3 km
33 long prismatic rectangular channel with constant roughness $n_M = 0.05$ s m^{-1/3} and the flow discharge
34 $Q = 180$ m³ s⁻¹ is adopted in this test case. The water depth varies periodically according to the
35 expression $h = 8 + 1.5 \sin^5 \left(\frac{\pi x}{500} \right)$, which produces values of the Froude number between 0.2 and 0.35
36 along the channel. The solution to the inverse problem leads to the bed profile shown in Fig. 8a.
37 Simulations were performed with grid resolutions of 20, 50, 100 and 200 m with both traditional and
38 sub-grid (coarse mesh), while 1 m² small cells were used for all sub-grid simulations. The value of
39 CFL for all simulations was set to 0.9.

40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
Figures 8a to 8c present the results of selected simulations (for clarity, only the coarsest grids
solutions are shown) at the centre of the channel, along with the corresponding analytical solution.
Figure 8d illustrates the absolute error $|E|$ between analytical and numerical water surface solutions.
Further error analysis is provided by Table 1, where the maximum and RMS errors between the two
models and the analytical solution are listed. At fine resolution (e.g. 20 m), the accuracy of both
models is within the order of cm, with the SG model delivering the best accuracy. The improved
accuracy of the SG over the T model becomes apparent as the grid resolution is coarsened. For

1
2
3 example, at 100 m resolution, the maximum absolute errors of T and SG models are 0.26 and 0.06 m,
4 respectively. It is important to observe that at the resolution of 200 m, the results of the proposed sub-
5 grid model are more accurate than the results of the Traditional model run at 50 m resolution. The
6 improved accuracy of the SG model is also shown in Fig. 8d.
7
8
9

10 5.3 Test Case 3

11
12 Test Case 3 is aimed at testing the ability of the proposed upscaling techniques to model the effects of
13 friction and topography in problems involving flow over bed with substantial variation of Manning's
14 coefficient and bed elevations. The domain is a 1200 m long prismatic and compound channel under a
15 steady flow rate of $Q = 20 \text{ m}^3 \text{ s}^{-1}$. The main channel and floodplains are rectangular. The base width
16 of the main channel is 4.0 m, and the width of both floodplains is 8 m. The elevation of the main
17 channel is 2 m below the elevation of the adjacent floodplains. The constant bed slope is 0.001. Two
18 roughness scenarios are modelled, (1) using the values of Manning's coefficient of $0.04 \text{ s m}^{-1/3}$ and
19 $0.08 \text{ s m}^{-1/3}$ for main channel and floodplains, respectively and (2) the space-averaged value of
20 scenario 1 for the whole domain, i.e. constant $n_M = 0.072 \text{ s m}^{-1/3}$ for the both main channel and
21 floodplain is adopted. A constant value of $\eta = 2.5 \text{ m}$ is imposed as downstream boundary condition
22 at $x = 1200 \text{ m}$. The model is run from an arbitrary initial condition for a duration long enough for the
23 steady flow condition to be reached. The small cell size is 2 m and the resolutions for the both
24 Traditional and Sub-grid models (coarse mesh) were 2 m, 4 m and 20 m. The solution of T (2) is
25 taken as the benchmark solution, against which all other solutions are compared.
26
27
28
29
30
31
32

33 Figure 9 shows the water surface elevation at the channel centreline obtained with the two
34 models at different resolutions and both Manning coefficients scenarios. In these figures, no
35 substantial differences can be observed between the free surface profiles predicted by the SG (4/2)
36 and T (4), which also coincide (visually) with the benchmark solution T (2). On the other hand, at the
37 coarsest resolution, T (20) and SG (20/2) results in over- and underestimated solutions, respectively
38 for the both scenarios. The results of SG (20/2) are much closer to the benchmark solution than those
39 of T (20).
40
41
42
43

44 Figure 9 also shows that differences between the coarse and benchmark solutions are higher
45 in scenario 1 (varied roughness) than in scenario 2 (constant roughness). This shows that simple
46 averaging of roughness coefficients across a computational cell may produce results that substantially
47 overestimate the effect of friction. Table 2 compares the maximum absolute $|E|$ and RMS errors (i.e.
48 relative to T (2) simulation) of the two models run at the computational cell resolutions of 4 m and 20
49 m. At 4 m resolution (in which case both the topographic and roughness data of cells in the main
50 channel and floodplain are the same as those in of the finest resolution grid), the results of both
51 models are very close to the benchmark solutions. At the coarsest resolution (i.e. $\Delta x = 20 \text{ m}$), where
52
53
54
55
56
57
58
59
60

each computational cell covers parts of both the floodplain and main channel, errors obtained with the T model are substantially larger than those provided by the SG model.

5.4 Test Case 4

Test Case 4 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 hours 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 km² and 400 km, respectively. The average flow in the river is 267 m³ s⁻¹, while the peak flow of the 200-year return period event is approximately 3,200 m³ s⁻¹ (Morales-Hernández et al., 2016). The computational domain of study is a 6 km x 2 km area near the city of Rome (Fig. 10). 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^{-1/3} and 0.0446 m s^{-1/3}, respectively.

The initial condition corresponds to the results of a steady flow simulation with discharge of 374 m³ s⁻¹. Figure 11 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, as shown in Fig. 10. Gridded topographic data is available at the resolutions of 2 m and 5 m. In the proposed sub-grid model, these two resolutions are used 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. The cell sizes for the T model is 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.

Figures 12a and 12b 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 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. Since the sub-grid model includes the elevation of small cells near the boundaries, even the coarsest resolution simulation was performed successfully. For clarity, only the results of some simulations are shown in Fig. 12. Overall, the results of all simulations produce water elevations that are higher than the benchmark solution at the 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 at coarse resolutions, 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,

1
2
3 respectively. No significant differences were observed between the results of the SG model at small
4 cell resolutions of 2 and 5 m.
5

6 Figure 13 compares the performance of the two models in terms of accuracy. The accuracy of
7 the simulations is assessed based on the RMSE of water depth of the solutions relative to the
8 benchmark [T (4)]. Figure 13a represents the solutions at $t = 110$ hr, while Fig. 13b corresponds to
9 errors at the maximum predicted water depth at each of the cells during 113 hours of flood event. In
10 Fig. 13b, the maximum water surface level computed at each computational cell is first determined for
11 the each model/resolution. The corresponding maximum water depth is then obtained by subtracting
12 the bed elevation at 2 m resolution from the maximum water surface levels. In Fig. 13a the process is
13 similar but instead of maximum water surface level, water surface levels at $t = 110$ hr are used. The
14 results in Figs 13a and 13b show the improved accuracy provided by the SG model compared to the
15 single mesh approach (i.e. T model) at different resolutions. At resolutions of the computational grid
16 smaller than 20 m, the accuracy of the T and SG models is close and display a similar trend. However,
17 as the size of the computational cell is coarsened, the curves diverge substantially, with the SG
18 displaying a much milder increase in error than the traditional approach. The analysis shows that the
19 slope of T (...) curve in $\log(\text{RMSE})$ vs $\log(\Delta x)$ is about 1 (exactly 0.95 and 0.9 for the cases Figs.
20 13a and 13b respectively), which would be expected for a first order model. By contrast, the curves
21 slopes for SG (.../2) and SG (.../5) are not constant and always less than 1 (i.e. errors increase at a
22 much slower pace as the grid is coarsened).
23
24
25
26
27
28
29
30

31 Table 3 presents the runtimes and speedups relative to the benchmark solution. Figure 14a
32 shows the runtime t_c for different simulations performed with the two models and at different
33 computational cell sizes. As previously discussed, the computational cost follows a power law $t_c \sim \Delta x^{-p}$
34 , where $p = 3$ for traditional explicit models. Figure 14a confirms this for the T model ($p = 3.013$) and
35 also shows how p is reduced for the SG model ($p = 2.216$ and 2.294 for SG (.../2) and SG (.../5),
36 respectively). The values of p less than 3 for the SG model are mainly because the fluxes at the cell
37 edges are computed at small cell resolution.
38
39
40
41

42 Figure 14b displays RMSE at $t = 110$ hr, when the differences of η predicted by the
43 benchmark solution and other solutions are relatively large (also previously shown in Fig. 12), as a
44 function of model runtime, obtained from the simulations with the two models at different resolutions.
45 This type of analysis provides a fair assessment of the trade-offs between accuracy and computational
46 performance, since both metrics are compared regardless of the resolutions at which they have been
47 obtained. The figure shows that as the simulation time is reduced (which in general is a result of grid
48 coarsening), the accuracy gap between the two models is gradually increased, with the SG model
49 outperforming the single mesh model. For instance, Fig. 14b shows that for runtimes less than $t_c = 5$
50 hr, the SG model provides the most accurate results.
51
52
53
54
55
56
57
58
59
60

6 Discussion

While the results of the model simulations presented in the previous sections show that the upscaled equations provide improved accuracy, some implications and inherent limitations of such modelling strategy must be discussed. First, the SG model presented in this paper 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 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 paper (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.

7 Conclusions

In this paper, a new model is 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) into the solution and therefore improve accuracy. Large cells can be fully or partially wet, with the wet area updated dynamically.

New methods are presented to upscale the solution from fine to coarse mesh 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, which are then used to solve the Riemann problem. 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 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 term, which assumes a generic distribution of momentum within a large cell under the assumption that the flow is approximately parallel. In this process, expensive computations involving fine scale information are also performed at pre-processing only.

Four test cases were studied in this paper, where the results of the sub-grid (SG) model were compared against a traditional first order finite-volume method (T). This assessment showed that at

the same grid resolution, the SG model provides improved accuracy over traditional methods. This enhanced accuracy is clearest 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. The combined analysis of accuracy and computational time reveals that at coarse resolutions and when similar levels of accuracy are considered, the SG model can be substantially faster than currently available methods.

Acknowledgements and data access

The work made extensive use of the University of Southampton super-computing resources Iridis4 and Iridis5. The authors 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. The authors 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 Test Case 4. Finally, the authors would like to thank Prof. Qihua Liang and two anonymous reviewers for the insightful comments. Data and the model used in this paper may be provided upon request to the authors.

Funding

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.

Notation

CFL = Courant–Friedrichs–Lewy condition (-)

e_x, e_y = x and y components of unit vector normal to cells boundary (-)

e = unit vector normal to cells boundary (-, -)

E = flux tensor ($m^2 s^{-1}$, $m^3 s^{-2}$, $m^3 s^{-2}$ | $m^2 s^{-1}$, $m^3 s^{-2}$, $m^3 s^{-2}$)

$|E|$ = absolute errors (m)

F, G = flux vectors in x and y direction ($m^2 s^{-1}$, $m^3 s^{-2}$, $m^3 s^{-2}$)

g = gravity acceleration ($m s^{-2}$)

h = water depth (m)

J, K = number of columns and rows of the small cells in a large cell (-)

1
2
3 L_d = length of the study domain (m)

4
5 n = time level ($s \text{ m}^{-1/3}$)

6
7 n_M = Manning coefficient ($s \text{ m}^{-1/3}$)

8
9 N = number of computational cells (-)

10
11 N_w = number of submerged small cells in a large cell (-)

12
13 N_T = number of submerged small cells adjacent to the cell boundary (-)

14
15 q_x, q_y = x and y components of unit width discharge ($\text{m}^2 \text{ s}^{-1}$)

16
17 \mathbf{q} = vector of unit width discharge ($\text{m}^2 \text{ s}^{-1}, \text{m}^2 \text{ s}^{-1}$)

18
19 Q = discharge ($\text{m}^3 \text{ s}^{-1}$)

20
21 s_{0x}, s_{0y} = x and y components of bed slope (-)

22
23 s_f = (constant) magnitude of the friction gradient at all small cells adjacent to the cell boundary (-)

24
25 s_{fx}, s_{fy} = x and y components of frictional slope (-)

26
27 \mathbf{S} = vector of source terms ($\text{m s}^{-1}, \text{m}^2 \text{ s}^{-2}, \text{m}^2 \text{ s}^{-2}$)

28
29 \mathbf{S}_b = vector of bed slope source term ($\text{m s}^{-1}, \text{m}^2 \text{ s}^{-2}, \text{m}^2 \text{ s}^{-2}$)

30
31 \mathbf{S}_f = vector of friction slope source term ($\text{m s}^{-1}, \text{m}^2 \text{ s}^{-2}, \text{m}^2 \text{ s}^{-2}$)

32
33 t = time (s or hr)

34
35 t_c = runtime and time cost (s or hr)

36
37 T_a = the first term of Eq. (17) ($\text{m}^{1/3} \text{ s}^{-1}$)

38
39 T_b = the second term of Eq. (17) ($\text{m}^{4/3} \text{ s}^{-1}$)

40
41 T_c = the third term of Eq. (17) ($\text{m}^{7/3} \text{ s}^{-1}$)

42
43 T_d = the first term of Eq. (37) ($\text{m}^{2/3} \text{ s}^{-2}$)

44
45 T_e = the second term of Eq. (37) ($\text{m}^{5/3} \text{ s}^{-2}$)

46
47 T_f = the third term of Eq. (37) ($\text{m}^{8/3} \text{ s}^{-2}$)

48
49 u, v = x and y velocity components (m s^{-1})

50
51 \mathbf{U} = vector of conservative variables ($\text{m}, \text{m}^2 \text{ s}^{-1}, \text{m}^2 \text{ s}^{-1}$)

52
53 \mathbf{V} = velocity vector ($\text{m s}^{-1}, \text{m s}^{-1}$)

54
55 W_d = width of the study domain (m)

1
2
3 $x, y =$ Cartesian coordinates (m)
4

5 $z =$ bed level (m)
6

7 $\hat{z} =$ a binary elevation function giving the minimum of bed and water surface level (m)
8

9 $\Gamma =$ boundary length (m)
10

11 $\delta x, \delta y =$ dimensions of small cells (m)
12

13 $\Delta x, \Delta Y =$ dimensions of large cells (m)
14

15 $\Delta t =$ time step (s)
16

17 $\varepsilon_x, \varepsilon_y =$ x and y components of the unit vector parallel to flow (-)
18

19 $\eta =$ water surface elevation (m)
20

21 $\sigma =$ frictional correction coefficient of intermediate state unit width discharge (-)
22

23 $\Phi =$ coefficient of unit width discharge distribution (-)
24

25 $\psi_a =$ approximated cross-section's conveyance ($\text{m}^2 \text{s}^{-1}$)
26

27 $\psi_e =$ exact cross-section's conveyance ($\text{m}^2 \text{s}^{-1}$)
28

29 $\Omega =$ cell area (m^2)
30

31 $\Omega_w =$ wet area of a cell (m^2)
32
33

34 **Reference**

- 35
36 Bates, P. D. (2000). Development and testing of a sub-grid scale model for moving-boundary
37 hydrodynamic problems in shallow water. *Hydrological Processes*, 14(11-12), 2073–2088.
38
39 Begnudelli, L., Sanders, B. F., & Bradford, S. F. (2008). Adaptive Godunov-Based model for flood
40 simulation. *Journal of Hydraulic Engineering*, 134(6), 714-725.
41
42 Bruwier, M., Archambeau, P., Erpicum, S., Piroton, M., & Dewals, B. (2017). Shallow-water models
43 with anisotropic porosity and merging for flood modelling on Cartesian grids. *Journal of*
44 *Hydrology*, 554, 693-709.
45
46
47 Burguete, J., García-Navarro, P., Murillo, J. & García-Palacín, I. (2007). Analysis of the friction term
48 in the one-dimensional shallow water model. *Journal of Hydraulic Engineering*, 133(9), 1048–
49 1063.
50
51
52 Cea, L., & Bladé, E. (2015). A simple and efficient unstructured finite volume scheme for solving the
53 shallow water equations in overland flow applications. *Water resources research*, 51(7), 5464–
54 5486, doi:10.1002/2014WR016259.
55
56
57
58
59
60

- 1
2
3 Cea, L., & Vázquez-Cendón, M. E. (2010). Unstructured finite volume discretization of two-
4 dimensional depth-averaged shallow water equations with porosity. *International journal for*
5 *numerical methods in fluid*, 63(8), 903–930.
6
7
8 Chow, V. T. (1959). *Open channel hydraulics*. McGraw-Hill, New York.
9
10 Cunge, J. A., Holly, F. M., & Verwey, A. (1980). *Practical aspects of computational river hydraulics*.
11 Pitman Publishing Ltd., London.
12
13 de Almeida, G. A., Bates, P., & Ozdemir, H. (2018). Modelling urban floods at submetre resolution:
14 challenges or opportunities for flood risk management? *Journal of Flood Risk Management*, 11,
15 S855– S865. <https://doi.org/10.1111/jfr3.12276>
16
17
18 Defina, A. (2000). Two-dimensional shallow flow equations for partially dry areas. *Water Resources*
19 *research*, 36(11), 3251–3264.
20
21 Defina, A., D'Alpaos, L., & Mattichio, B. (1994). A new set of equations for very shallow water and
22 partially dry areas suitable to 2D numerical models. In: *Proceedings of the Specialty Conference*
23 *on "Modelling of Flood Propagation Over Initially Dry Areas"*, New York, USA, 72–81.
24
25
26 Duan, Y., Liu, Z., Chen, Y., & Zhu, D. (2017). Improved 2D shallow water model able to capture the
27 effects of complex bathymetric features through their subgrid modeling. *Journal of Hydraulic*
28 *Engineering*, 143(2), 04016081.
29
30
31 Fleischmann, A., Paiva, R., & Collischonn, W. (2019). Can regional to continental river
32 hydrodynamic models be locally relevant? A cross-scale comparison. *Journal of Hydrology X*,
33 3, 100027– 100045. <https://doi.org/10.1016/j.hydroa.2019.100027>.
34
35
36 Guinot, V. (2012). Multiple porosity shallow water models for macroscopic modelling of urban
37 floods. *Advances in Water Resources*, 37, 40–72.
38
39
40 Guinot, V. (2017). A critical assessment of flux and source term closures in shallow water models
41 with porosity for urban flood simulations. *Advances in Water Resources*, 109, 133–157.
42
43
44 Guinot, V., Delenne, C., Rousseau, A., & Boutron, O. (2018). Flux closures and source term models
45 for shallow water models with depth-dependent integral porosity. *Advances in Water Resources*,
46 122, 1–26. <https://doi.org/10.1016/j.advwatres.2018.09.014>.
47
48
49 Guinot, V., Sanders, B. F., & Schubert, J. E. (2017). Dual integral porosity shallow water model for
50 urban flood modelling. *Advances in Water Resources*, 103, 16–31.
51
52
53 Guinot, V., & Soares-Frazão, S., (2006). Flux and source term discretisation in two dimensional
54 shallow water models with porosity on unstructured grids. *International Journal for Numerical*
55 *Methods in Fluids*, 50(3), 309–345.
56
57
58
59
60

- 1
2
3 Hénonin, J., Hongtao, M., Zheng-Yu, Y., Hartnack, J., Havnø, K., Gourbesville, P., & Mark, O.
4 (2015), Citywide multi-grid urban flood modelling: the July 2012 flood in Beijing. *Urban Water*
5 *Journal*, 12(1), 52–66.
6
7
8 Hoch, J. M., van Beek, R., Winsemius, H. C., & Bierkens, M. F. P. (2018). Benchmarking flexible
9 meshes and regular grids for large-scale fluvial inundation modelling. *Advances in Water*
10 *Resources*, 121, 350-360. <https://doi.org/10.1016/j.advwatres.2018.09.003>
11
12
13 Horrit, M. S., & Bates, P. D. (2001). Effects of spatial resolution on a raster based model of flood flow.
14 *Journal of Hydrology*, 253(1-4), 239-249. [https://doi.org/10.1016/S0022-1694\(01\)00490-5](https://doi.org/10.1016/S0022-1694(01)00490-5).
15
16
17 Hou, J., Liang, Q., Zhang, H., & Hinkelmann, R. (2015). An efficient unstructured MUSCL scheme
18 for solving the 2D shallow water equations. *Environmental Modelling & software*, 66, 131-152.
19
20
21 Huang, C. -J., Hsu, M. -H., Teng, W. -H., & Wang, Y. -H. (2014). The impact of building coverage in
22 the metropolitan area on the flow calculation. *Water*, 6(8), 2449-2466.
23
24
25 Kesserwani, G., & Liang, Q. (2012). Locally limited and fully conserved RKDG2 shallow water
26 solutions with wetting and drying. *Journal of Scientific Computing*, 50(1), 120–144.
27
28
29 Kim, B., Sanders, B. F., Famiglietti, J. S., & Guinot, V. (2015). Urban flood modeling with porous
30 shallow-water equations: A case study of model errors in the presence of anisotropic porosity.
31 *Journal of Hydrology*, 523, 680–692.
32
33
34 Leveque, R. J. (2002). *Finite volume methods for hyperbolic systems*. Cambridge University Press,
35 Cambridge, UK.
36
37
38 Liang, Q. (2010). Flood simulation using a well-balanced shallow flow model. *Journal of Hydraulic*
39 *Engineering*, 136(9), 669-675.
40
41
42 Liang, Q. (2012). A simplified adaptive Cartesian grid system for solving the 2D shallow water
43 equations. *International Journal for Numerical Methods in Fluids*, 69(2), 442-458.
44
45
46 Liang, Q., & Borthwick, A. G. (2009). Adaptive quadtree simulation of shallow flows with wet-dry
47 fronts over complex topography. *Computers & Fluids*, 38(2), 221-234.
48
49
50 Liang, Q., & Marche, F. (2009). Numerical resolution of well-balanced shallow water equations with
51 complex source terms. *Advances in Water Resources*, 32(6), 873-884.
52
53
54 MacDonald, I. (1996). *Analysis and computation of steady open channel flow*. PhD thesis,
55 Department of Mathematics., University of Reading, Reading, U. K.
56
57
58 McMillan, H. K., & Brasington, J. (2007). Reduced complexity strategies for modelling urban
59 floodplain inundation. *Geomorphology*, 90(3-4), 226–243.
60

- 1
2
3 Morales-Hernández, M., Petaccia, G., Brufau, P., & García-Navarro, P. (2016). Conservative 1D-2D
4 coupled numerical strategies applied to river flooding: the Tiber (Rome). *Applied Mathematical*
5 *Modelling*, *40*(3), 2087-2105.
6
7
8 Neal, J. C., Fewtrell, T. J., & Trigg, M. A. (2009). Parallelisation of storage cell flood models using
9 OpenMP. *Environmental Modelling & Software*. *24*(7), 872–87.
10
11 Özgen, I., Liang, D., & Hinkelmann, R. (2016a). Shallow water equations with depth-dependent
12 anisotropic porosity for subgrid-scale topography. *Applied. Mathematical Modelling*, *40*(17-18),
13 7447-7473.
14
15
16 Özgen, I., Liang, D., & Hinkelmann, R. (2016b). Urban flood modeling using shallow water
17 equations with depth-dependent anisotropic porosity. *Journal of Hydrology*, *541*, 1165–1184.
18
19
20 Park, S., Kim, B. & Kim, D. (2019). 2D GPU-accelerated high resolution numerical scheme for
21 solving diffusive wave equations. *Water*. *11*(7), 1447.
22
23
24 Platzek, F. W., Stelling, G. S., Jankowski, J. A., Patzwahl, R., & Pietrzak, J. D. (2016). An efficient
25 semi-implicit subgrid method for free-surface flow on hierarchical grids. *International Journal*
26 *for Numerical Methods in Fluids*, *80*(12), 715–741. doi: 10.1002/flid.4172.
27
28
29 Sanders, B. F., & Schubert, J. E. (2019). PRIMo: Parallel raster inundation model. *Advances in Water*
30 *Resources*, *126*, 79–95.
31
32 Sanders, B. F., Schubert, J. E., & Gallegos, H. A. (2008). Integral formulation of shallow-water
33 equations with anisotropic porosity for urban flood modeling. *Journal of Hydrology*, *362*(1-2),
34 19–38.
35
36
37 Schubert, J. E., & Sanders, B. F. (2012). Building treatments for urban flood inundation models and
38 implications for predictive skill and modeling efficiency. *Advances in Water Resources*, *41*, 49–
39 64. <http://dx.doi.org/10.1016/j.advwatres.2012.02.012>.
40
41
42 Soares-Frazão, S., Lhomme, J., Guinot, V., & Zech, Y. (2008). Two-dimensional shallow-water
43 model with porosity for urban flood modelling. *Journal of Hydraulic Research*, *46*(1), 45–64.
44
45
46 Stelling, G. S. (2012). Quadtree flood simulations with sub-grid digital elevation models. In
47 *Proceedings of the Institution of Civil Engineers-Water Management*, *165*, 567-580. Thomas
48 Telford Ltd.
49
50
51 Toro, E. (2001). *Shock-capturing methods for free-surface shallow flows*. John Wiley, Chichester,
52 UK.
53
54
55 Toro, E. F., Spruce, M., & Speares, W. (1994). Restoration of the contact surface in the HLL-
56 Riemann solver. *Shock Waves*, *4*(1), 25–34.
57
58
59
60

- 1
2
3 Valiani, A., & Begnudelli, L. (2006). Divergence form for bed slope source term in shallow water
4 equations. *Journal of Hydraulic Engineering*, *132*(7), 652-665.
5
6 Volp, N., Van Prooijen, B., & Stelling, G. (2013). A finite volume approach for shallow water flow
7 accounting for high-resolution bathymetry and roughness data. *Water Resources Research*,
8 *49*(7), 4126–4135.
9
10
11 Wang, Y., Liang, Q., Kesserwani, G., & Hall, J. W. (2011). A positivity-preserving zero-inertia model
12 for flood simulation. *Computers & Fluids*, *46*(1), 505–511.
13
14 Xia, X. & Liang, Q. (2018). A new efficient implicit scheme for discretising the stiff friction terms in
15 the shallow water equations. *Advances in Water Resources*, *117*, 87–97.
16
17
18 Yu, D., & Lane, S. N. (2006). Urban fluvial flood modelling using a two-dimensional diffusion-wave
19 treatment, part 1: mesh resolution effects. *Journal of Hydrological Process*, *20*(7), 1541–1565.
20
21 Yu, D., & Lane, S. N. (2011). Interactions between subgrid-scale resolution, feature representation
22 and grid-scale resolution in flood inundation modelling. *Hydrological. Process*, *25*(1), 36–53.
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

List of tables

Table 1. Maximum absolute (Max. $|E|$) and RMS errors of water level relative to the analytical solution obtained in Test Case 2.

Table 2. Maximum absolute (Max. $|E|$) and RMS errors of water level relative to the benchmark solution [T (2)] for Scenarios 1 and 2 in Test Case 3.

Table 3. Simulation time and speedup relative to benchmark solution [T (4)] for the different models/resolutions (Test Case 4).

List of figures

Figure 1 Nested meshes adopted in this paper for the proposed sub-grid model.

Figure 2 Example of the distribution of bed elevation within and along the interface between two large cells.

Figure 3 Unit width discharge distribution across the edge Γ_4 of i th large cell of the domain.

Figure 4 Cross section of an arbitrary large cell illustrating the definition of \hat{z} used by the model to compute bed slope source term. (a) In fully wet cells, \hat{z} is simply equal to z but (b) in partially wet cells, \hat{z} in unsubmerged small cells is lower than z (i.e. $\hat{z} = \eta$).

Figure 5 General structure of the proposed model. M_z stores the topographic data of small cells within each large cell. $M_{z_{sorted}}$ 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.

Figure 6 Computational domain and SG grids used in Test Case 1. The results of the model are reported along the red line indicated in the figure.

Figure 7 Longitudinal water surface level profiles obtained in Test 1 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 exactly the same.

Figure 8 (a) Longitudinal bed and water surface level for Test Case 2 for $0 \leq x \leq 3000$; (b) Water surface results at the upstream and (c) downstream half of the channel; (d) Absolute error ($|E|$) between analytical and numerical solutions along the channel.

Figure 9 Longitudinal profile of water surface solutions for (a) Scenario 1 and (b) Scenario 2 in Test Case 3.

Figure 10 Plan view of the computational domain used in Test Case 4.

Figure 11 Upstream and downstream boundary conditions used in Test Case 4. (a) Upstream (western boundary) time series $Q(t)$ and (b) downstream (southern boundary) time series $\eta(t)$.

Figure 12 Measured and predicted water surface elevations at (a) cross-section S1 and (b) S2 of the River Tiber (Test Case 4).

1
2
3 Figure 13 Water depth RMSE of the models/resolutions relative to the benchmark solution against
4 their computational cell size (Δx) where, water depth is, (a) values at $t = 110$ hr and (b) the maximum
5 value computed during the flood event (Test Case 4).
6
7

8 Figure 14 (a) Runtime (t_c) against computational cell size of the models/resolutions. (b) Water depth
9 RMSE (at $t = 110$ hr) of the models/resolutions relative to the benchmark solution as a function of
10 runtime (Test Case 4).
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Table 1. Maximum absolute (Max. $|E|$) and RMS errors of water level relative to the analytical solution obtained in Test Case 2.

Cell size (m)	Max. $ E $ (m)		RMSE (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

Table 2. Maximum absolute (Max. $|E|$) and RMS errors of water level relative to the benchmark solution [T (2)] for Scenarios 1 and 2 in Test Case 3.

Scenario number	Cell size	Max. $ E $ (m)		RMSE (m)	
		T	SG	T	SG
1	4	0.008	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

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

Table 3. Simulation time and speedup relative to benchmark solution [T (4)] for the different models/resolutions (Test Case 4).

Model/resolution	Runtime (hr)	Speedup
SG (10/2)	97.5	4.2
SG (20/2)	20.8	19.8
SG (40/2)	4.6	89.6
SG (50/2)	2.7	152.6
SG (100/2)	0.6	686.8
SG (10/5)	39.9	10.3
SG (20/5)	8.7	47.4
SG (40/5)	1.8	228.9
SG (50/5)	1.0	412.1
SG (100/5)	0.2	2060.5
T (4)	412.1	1.0
T (10)	27.1	15.2
T (20)	3.3	124.9
T (40)	0.4	1030.3
T (50)	0.2	2060.5

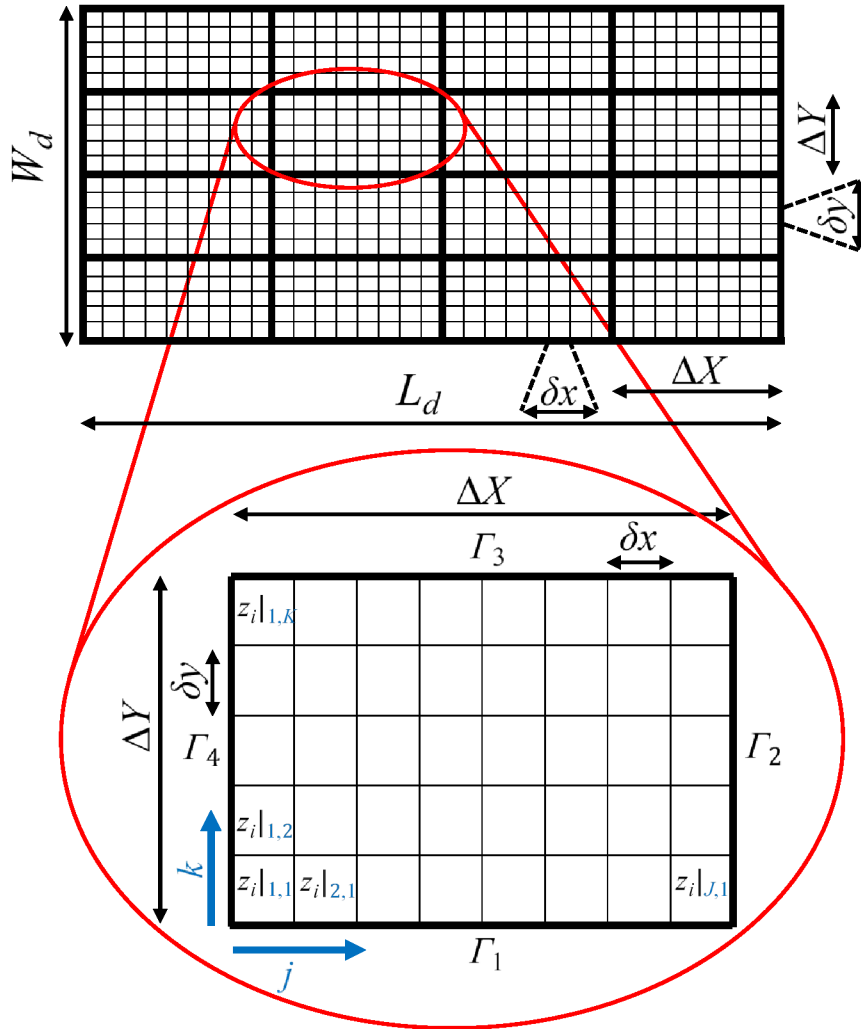


Figure 1 Nested meshes adopted in this paper for the proposed sub-grid model.

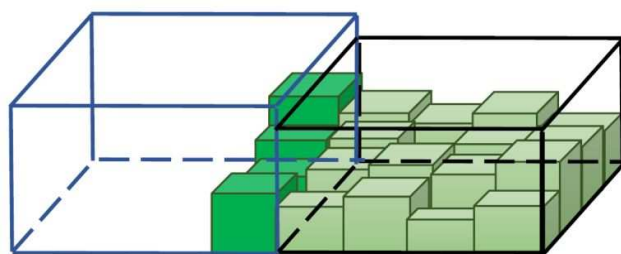


Figure 2 Example of the distribution of bed elevation within and along the interface between two large cells.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

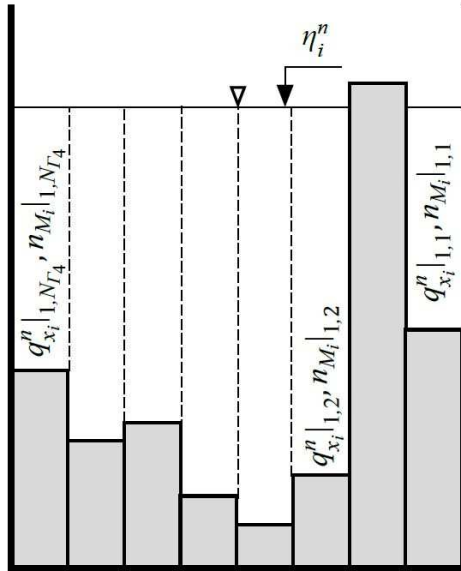


Figure 3 Unit width discharge distribution across the edge Γ_4 of i^{th} large cell of the domain.

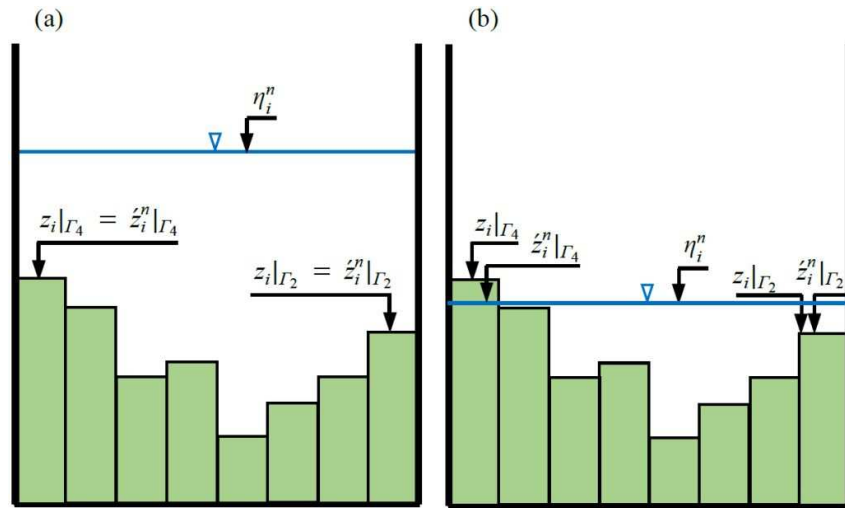


Figure 4 Cross section of an arbitrary large cell illustrating the definition of \hat{z} used by the model to compute bed slope source term. (a) In fully wet cells, \hat{z} is simply equal to z but (b) in partially wet cells, \hat{z} in unsubmerged small cells is lower than z (i.e. $\hat{z} = \eta$).

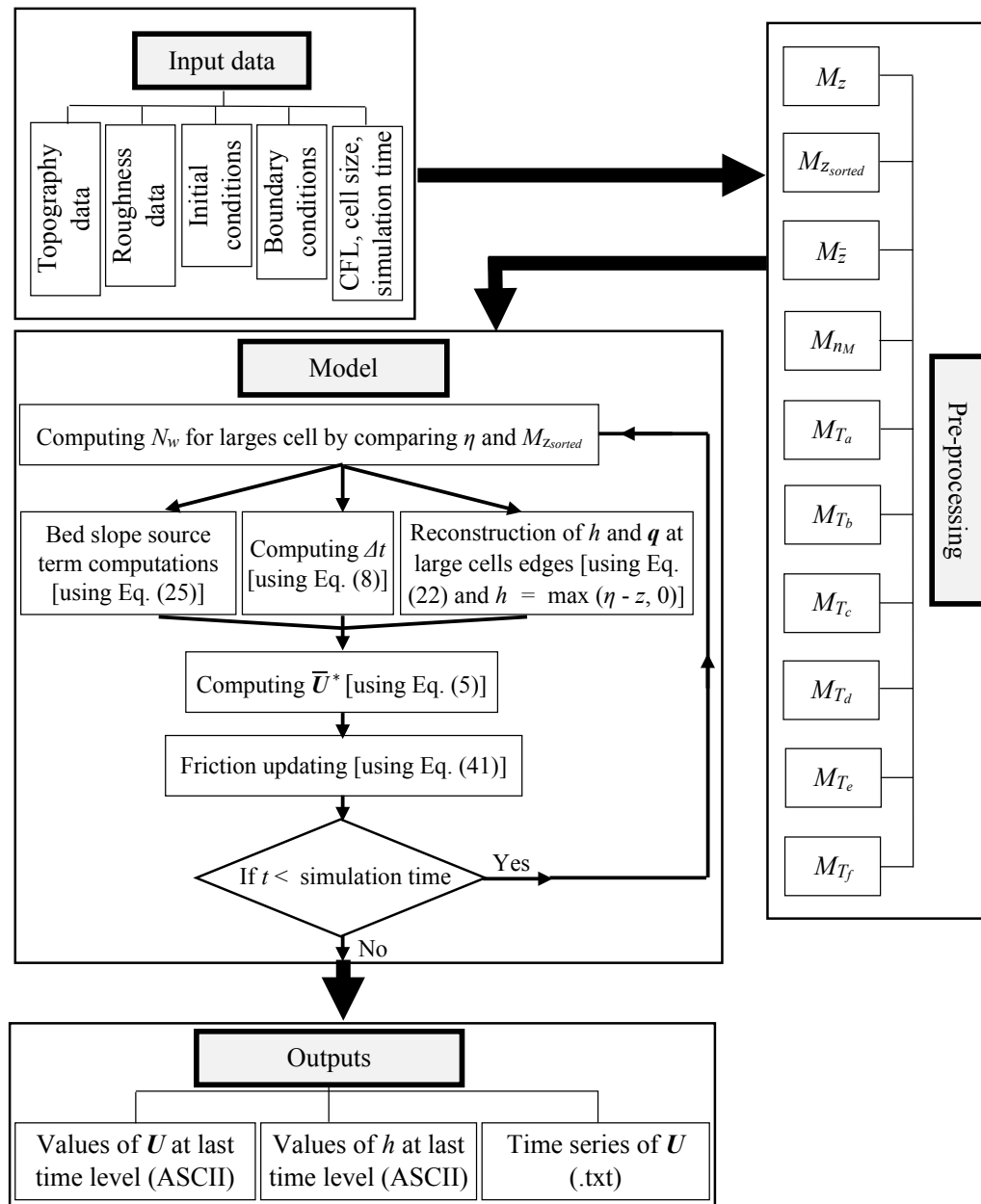


Figure 5 General structure of the proposed model. M_z stores the topographic data of small cells within each large cell. $M_{z_{sorted}}$ stores the same topographic information sorted by bed elevation. $M_{\bar{z}}$ stores the values of \bar{z} corresponding to the number of submerged small cells. M_{nM} 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.

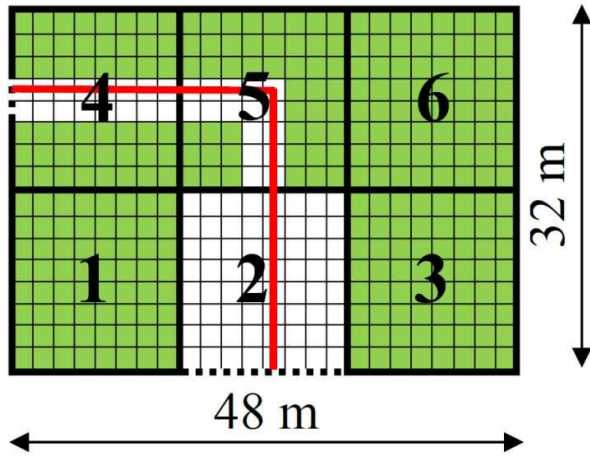


Figure 6 Computational domain and SG grids used in Test Case 1. The results of the model are reported along the red line indicated in the figure.

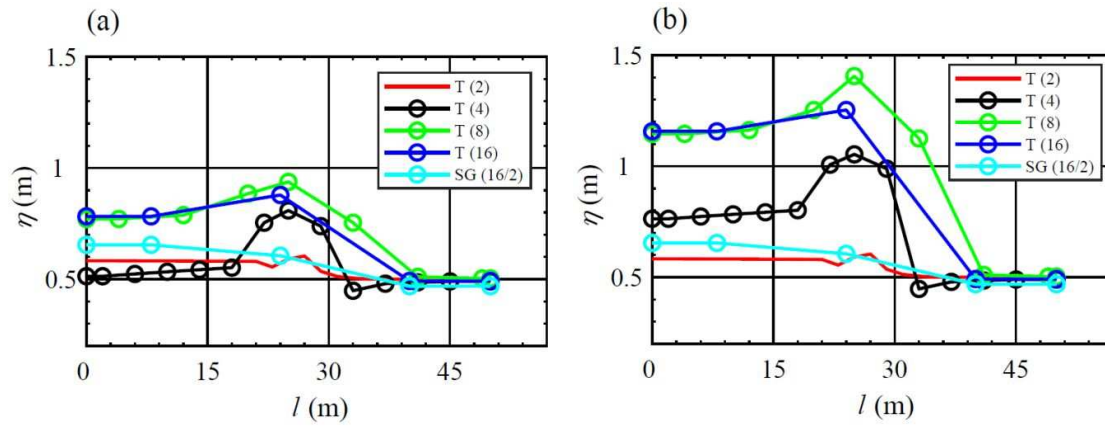


Figure 7 Longitudinal water surface level profiles obtained in Test 1 with the two models at various resolutions. (a) $z_f = 1.0$ m and (b) $z_f = 1.5$ m. Notice that the results of T (2) and SG (16/2) obtained with $z_f = 1.0$ m and $z_f = 1.5$ m are exactly the same.

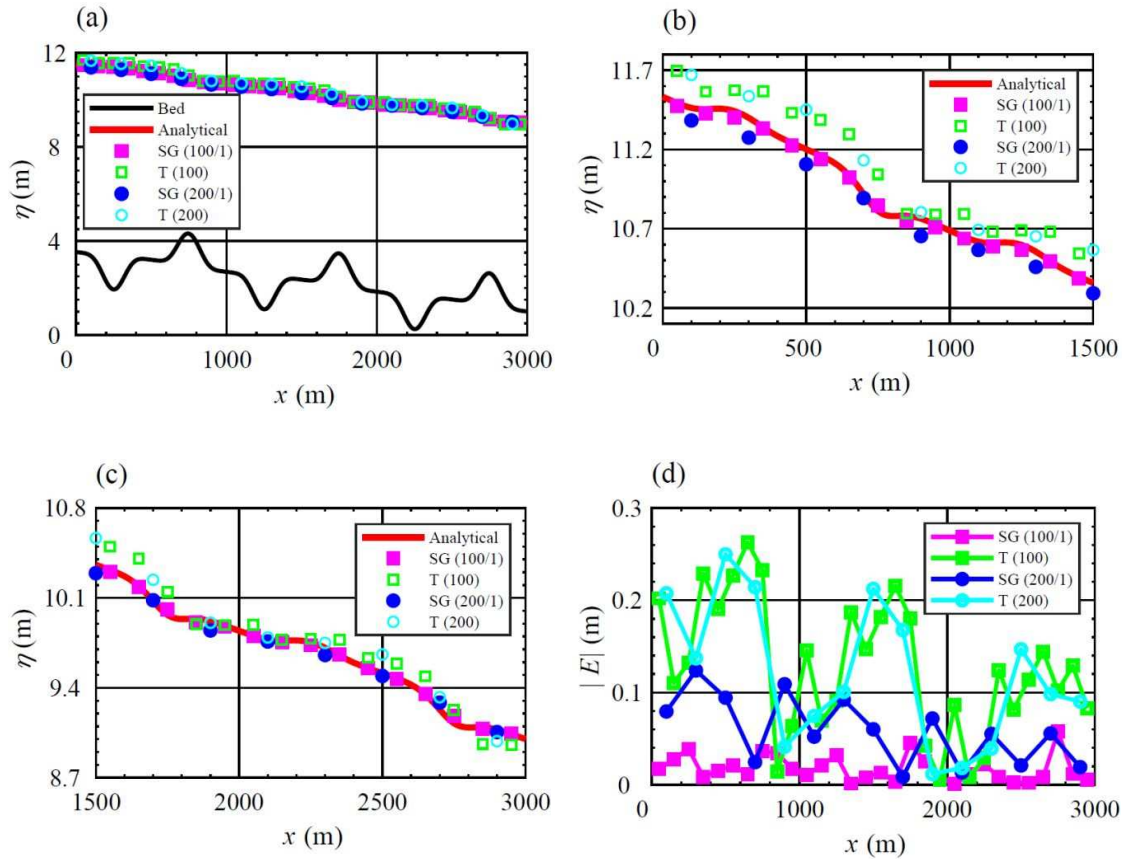


Figure 8 (a) Longitudinal bed and water surface level for Test Case 2 for $0 \leq x \leq 3000$; (b) Water surface results at the upstream and (c) downstream half of the channel; (d) Absolute error ($|E|$) between analytical and numerical solutions along the channel.

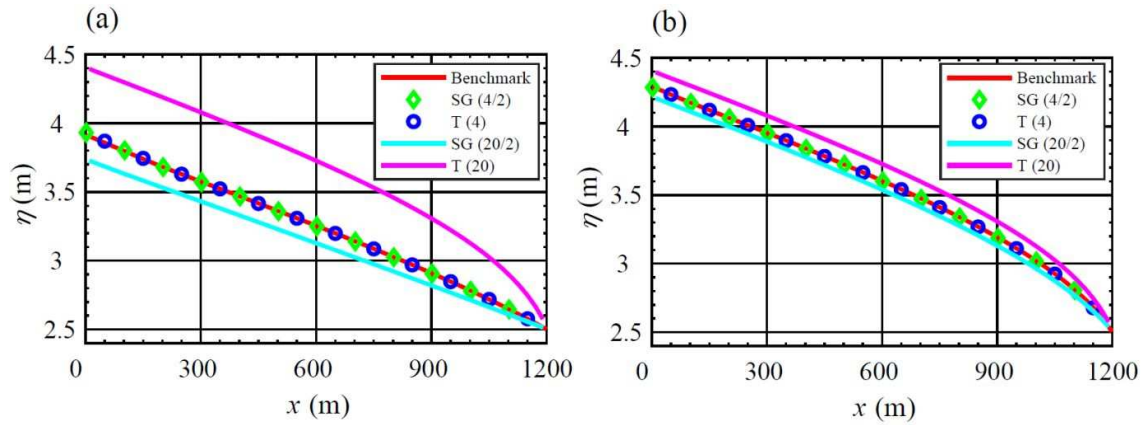


Figure 9 Longitudinal profile of water surface solutions for (a) Scenario 1 and (b) Scenario 2 in Test Case 3.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

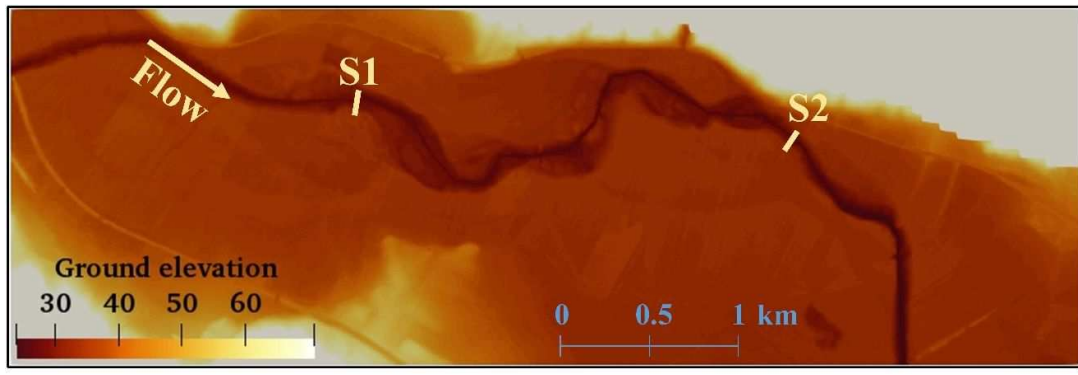


Figure 10 Plan view of the computational domain used in Test Case 4.

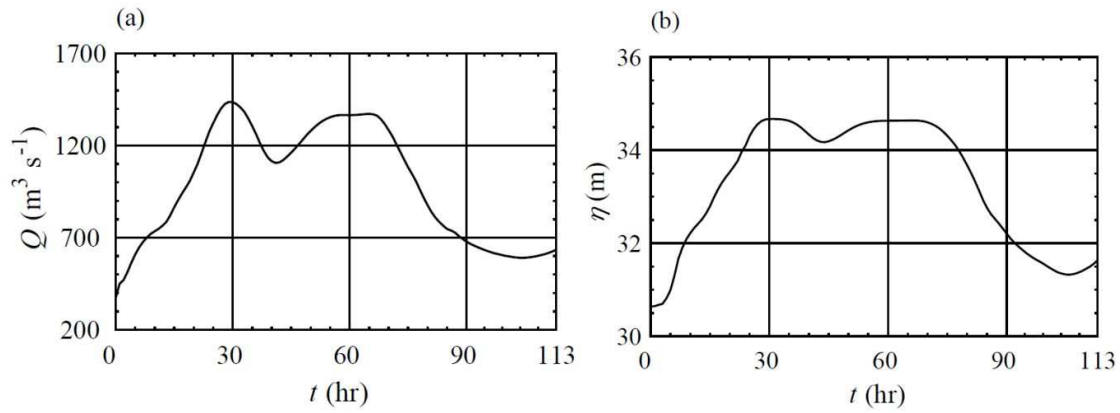


Figure 11 Upstream and downstream boundary conditions used in Test Case 4. (a) Upstream (western boundary) time series $Q(t)$ and (b) downstream (southern boundary) time series $\eta(t)$.

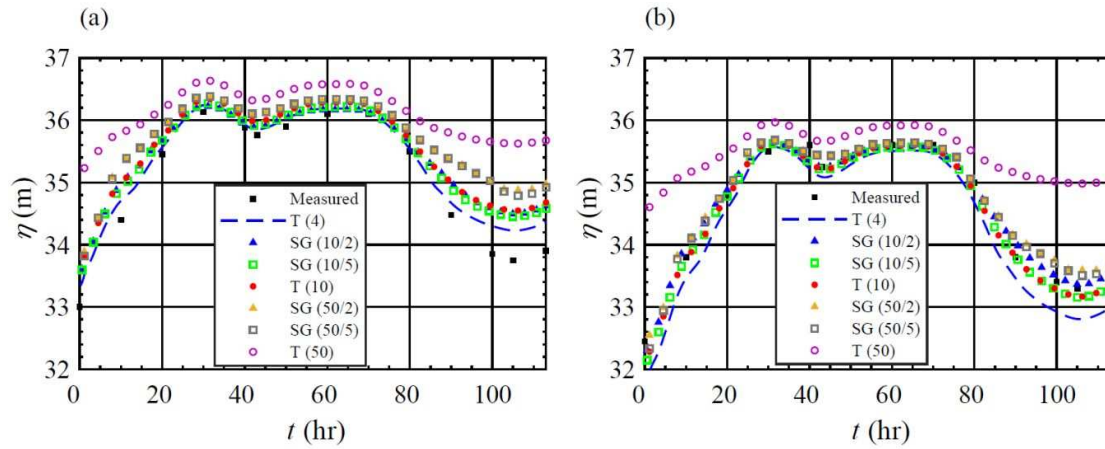


Figure 12 Measured and predicted water surface elevations at (a) cross-section S1 and (b) S2 of the River Tiber (Test Case 4).

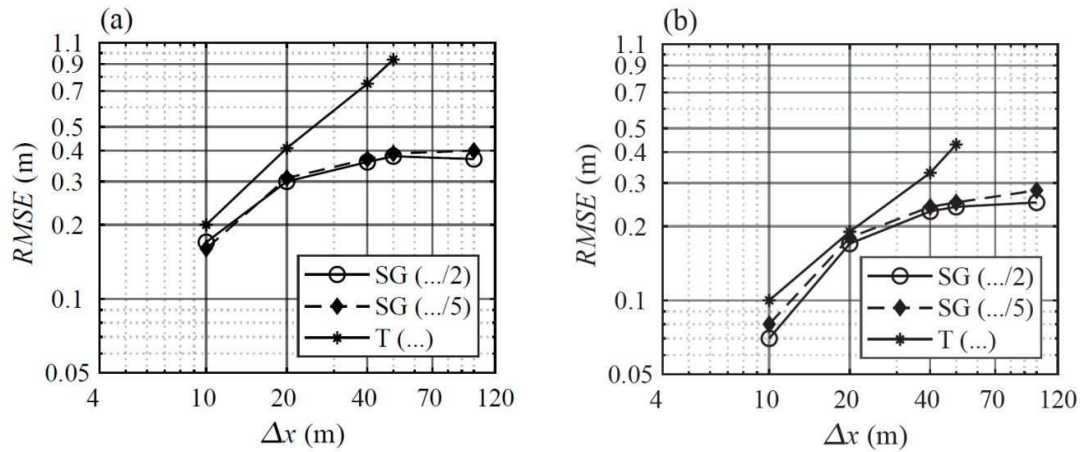


Figure 13 Water depth RMSE of the models/resolutions relative to the benchmark solution against their computational cell size (Δx) where, water depth is, (a) values at $t = 110$ hr and (b) the maximum value computed during the flood event (Test Case 4).

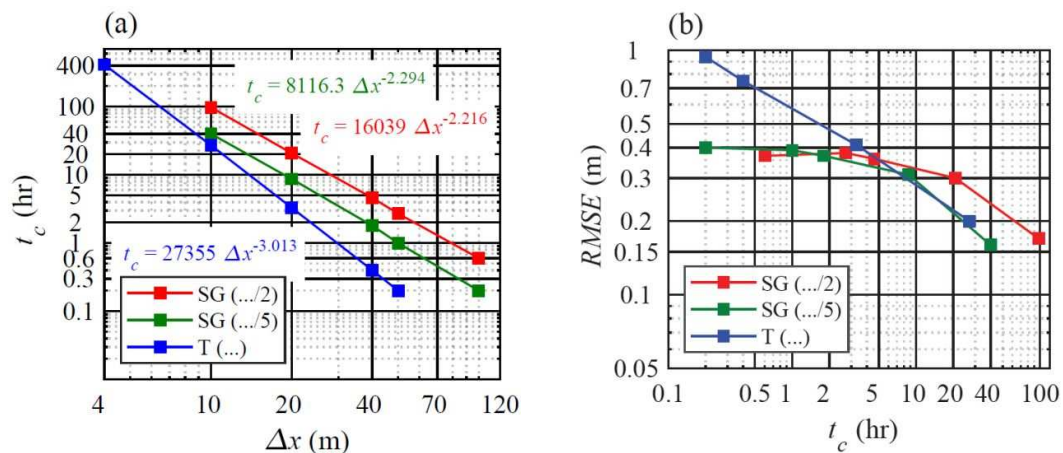
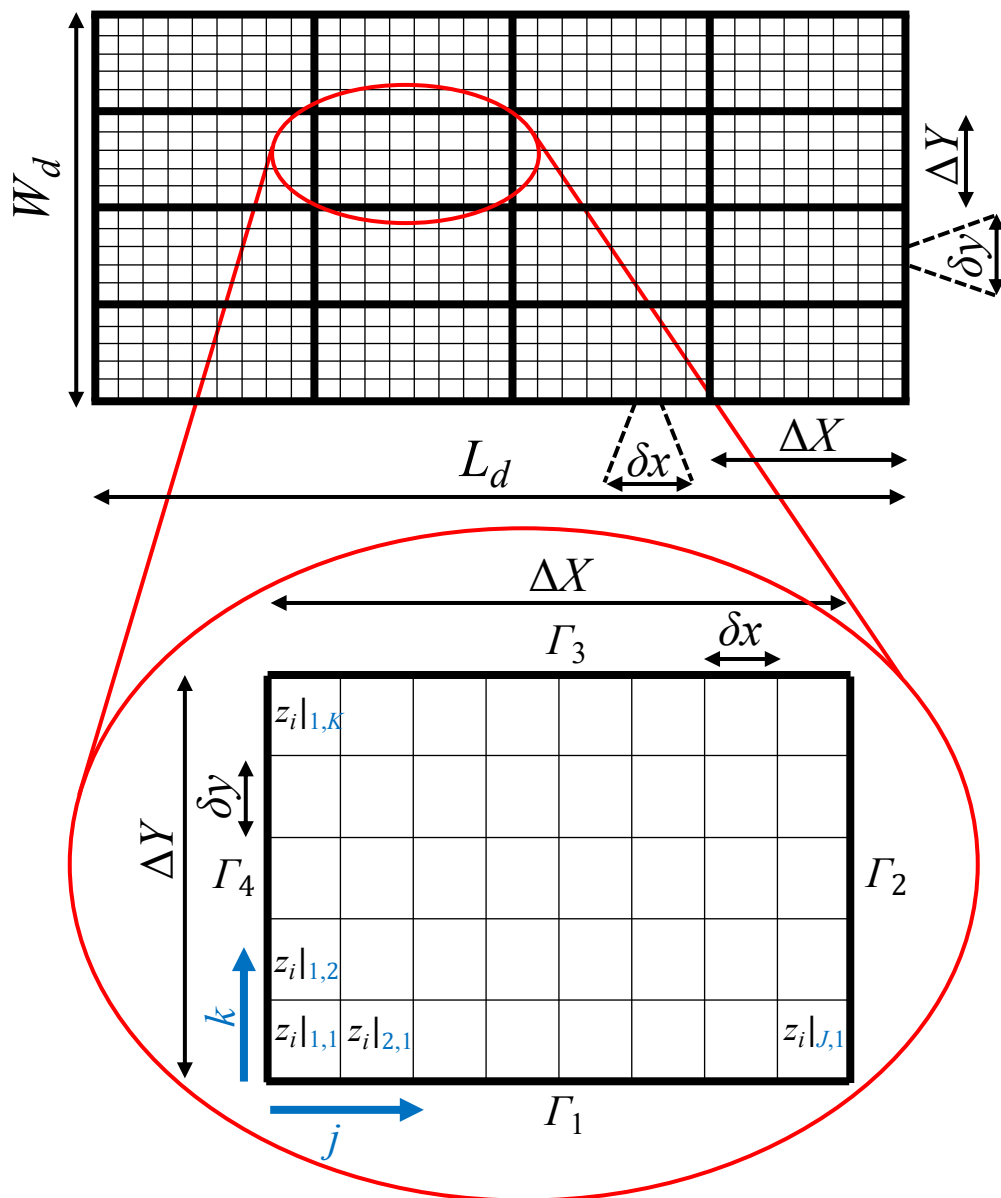
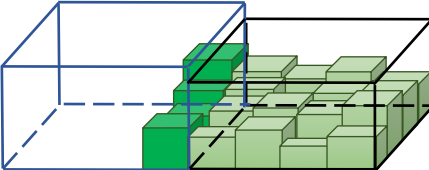
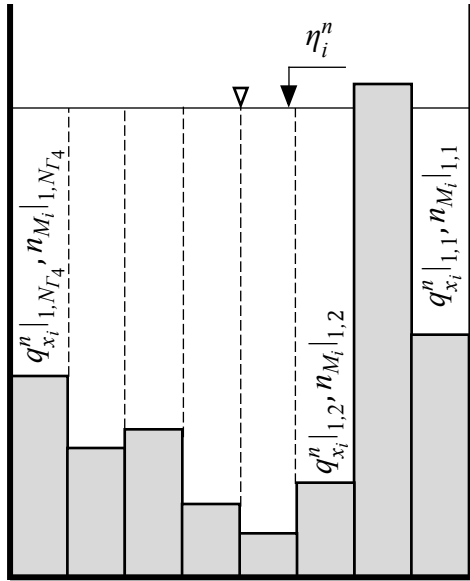


Figure 14 (a) Runtime (t_c) against computational cell size of the models/resolutions. (b) Water depth RMSE (at $t = 110$ hr) of the models/resolutions relative to the benchmark solution as a function of runtime (Test Case 4).

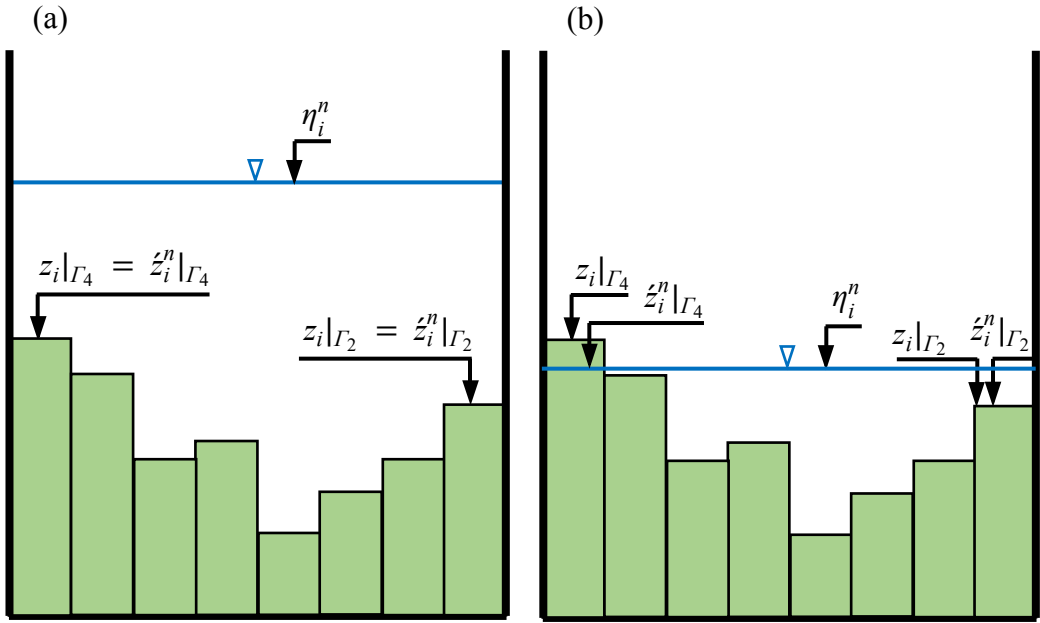


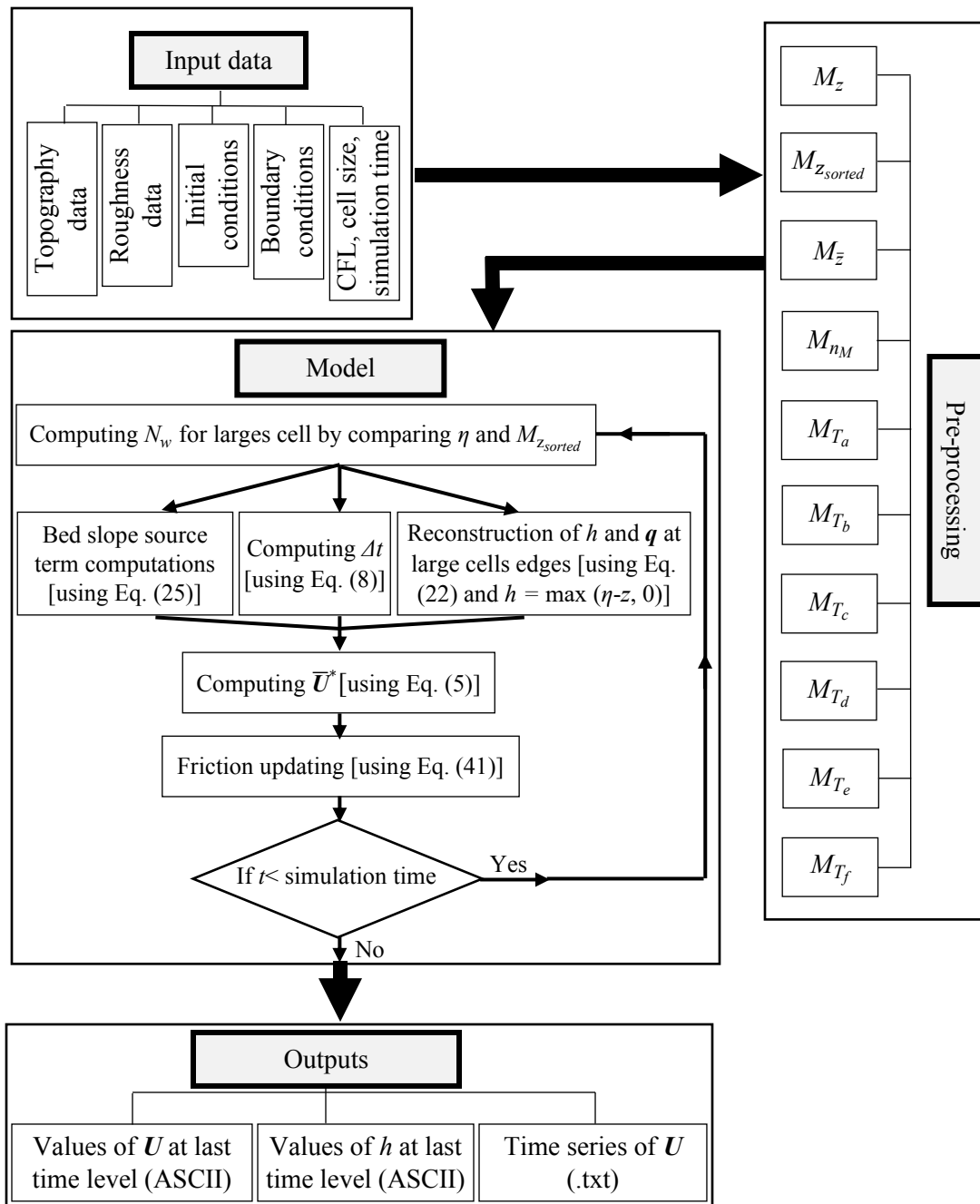
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



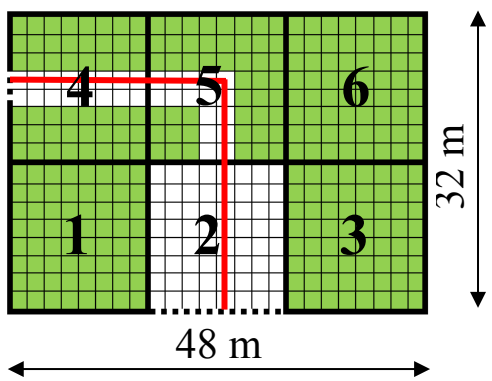


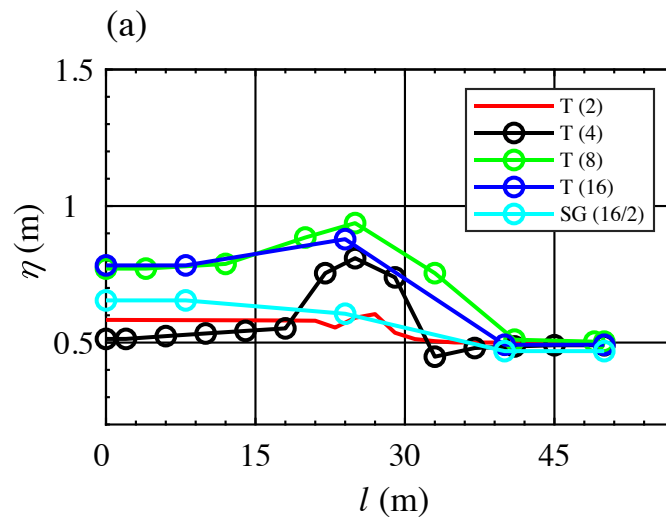
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



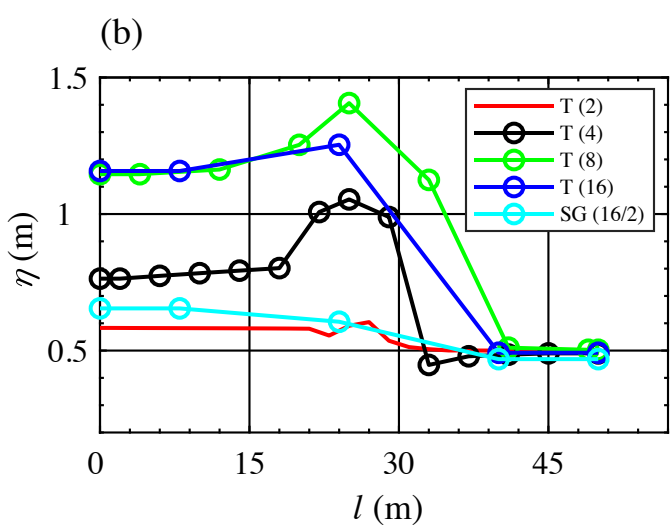


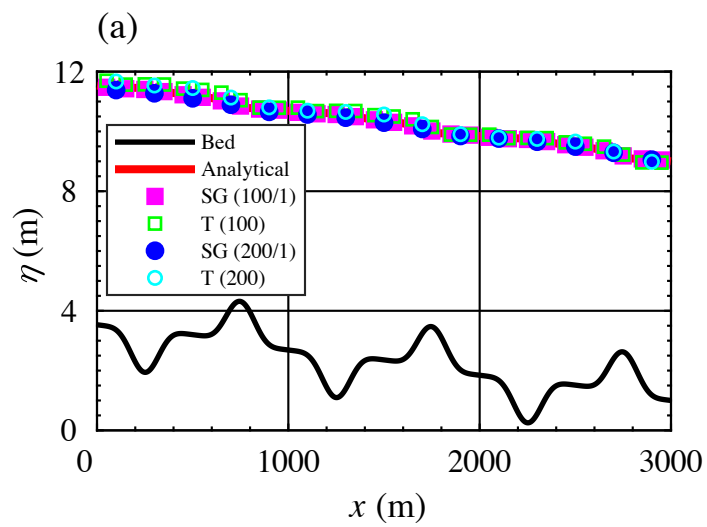
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



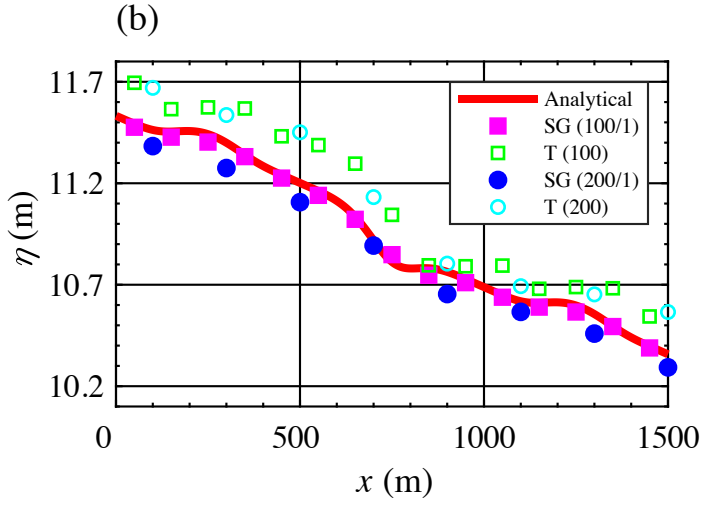


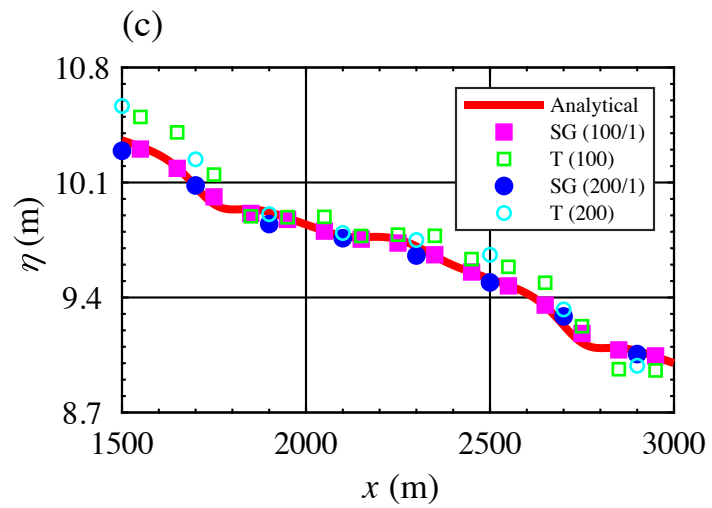
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



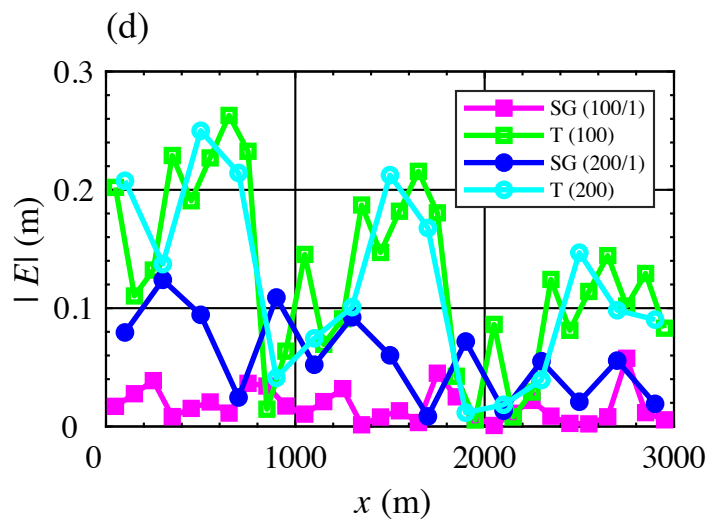


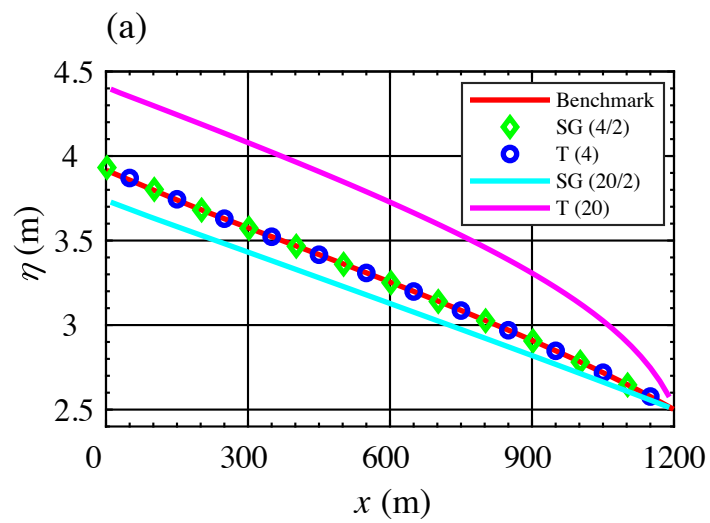
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



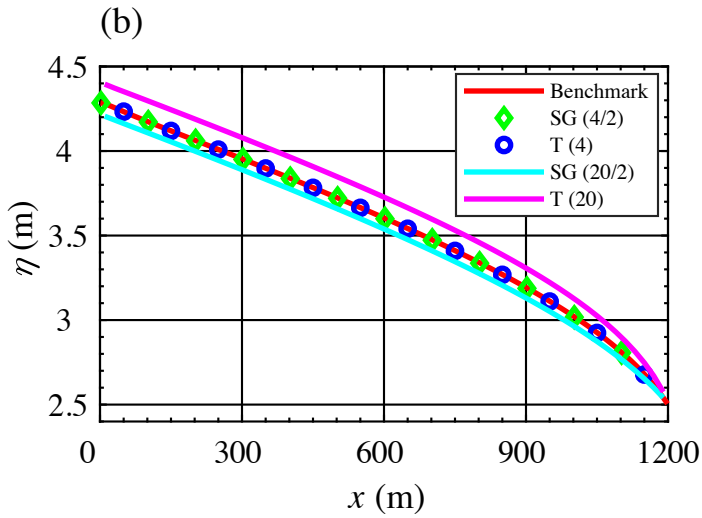


1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60





1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



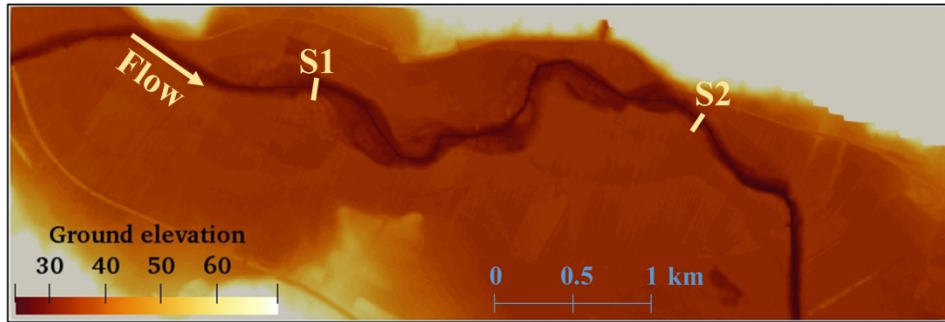
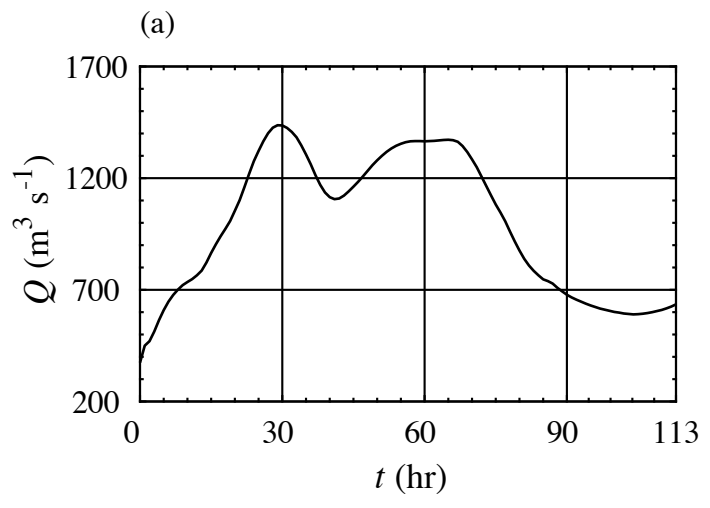
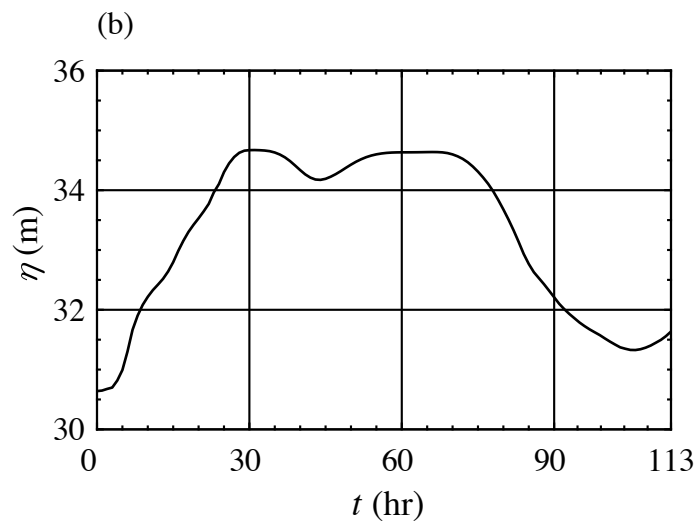


Figure 10 Plan view of the computational domain used in Test Case 4.

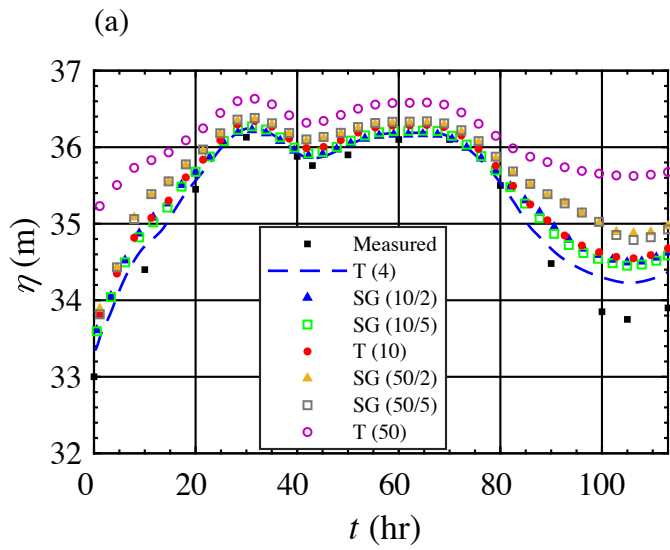
636x249mm (300 x 300 DPI)

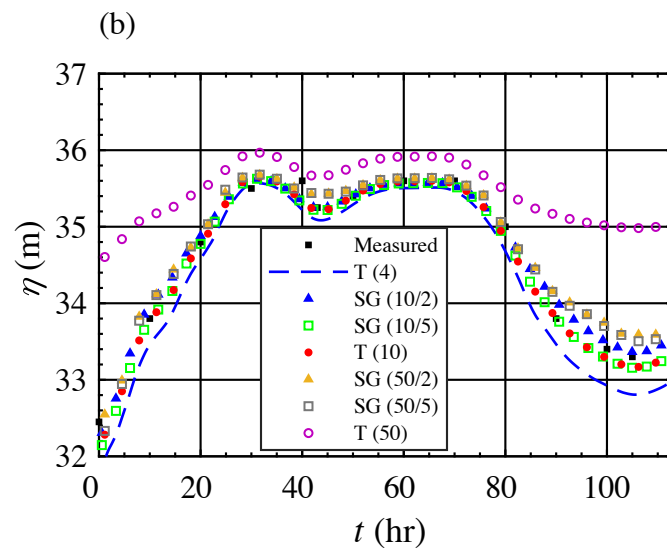
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60



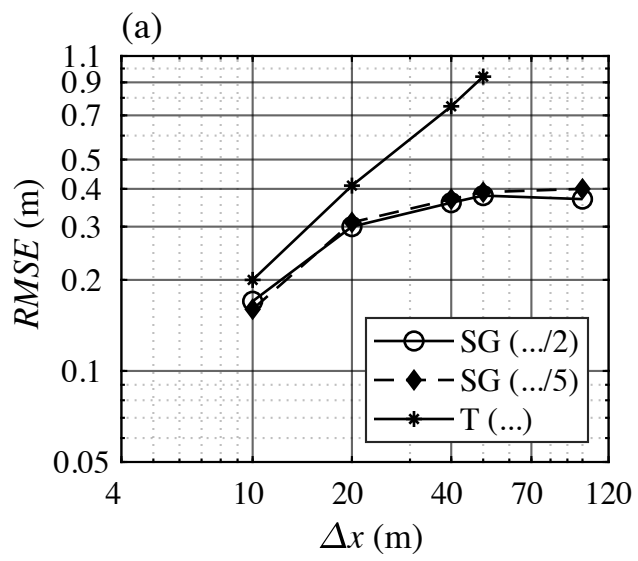


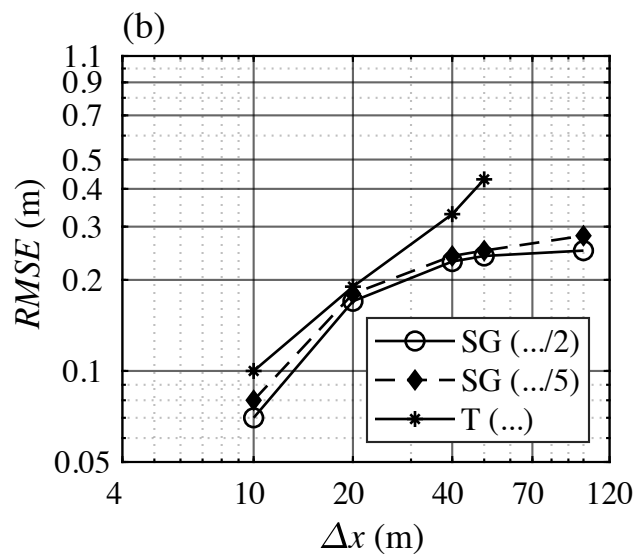
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60





1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60





1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60

