Title
Mesh type tradeoffs in 2D hydrodynamic modeling of flooding with a Godunov-based flow solver

Authors
Kim, Byunghyun
Sanders, Brett F
Schubert, Jochen E
et al.

Publication Date
2014-06-01

DOI
10.1016/j.advwatres.2014.02.013

Peer reviewed

Advances in Water Resources 68 (2014) 42–61

Contents lists available at ScienceDirect

Advances in Water Resources

journal homepage: www.elsevier.com/locate/advwatres

Mesh type tradeoffs in 2D hydrodynamic modeling of flooding with a Godunov-based flow solver

Byunghyun Kim [a,b], Brett F. Sanders [a,b,*], Jochen E. Schubert [a,b], James S. Famiglietti [a,b,c]

[a] University of California Center for Hydrologic Modeling, Irvine, CA 92697, USA
[b] Department of Civil and Environmental Engineering, University of California, Irvine, CA 92697, USA
[c] Department of Earth System Science, University of California, Irvine, CA 92697, USA

ARTICLE INFO

Article history:
Received 2 July 2013
Received in revised form 30 January 2014
Accepted 23 February 2014
Available online 16 March 2014

Keywords:
Flooding
Finite volume model
Mesh design
Unstructured grid
Cartesian grid
Porosity

ABSTRACT

The effect of mesh type on the accuracy and computational demands of a two-dimensional Godunov-type flood inundation model is critically examined. Cartesian grids, constrained and unconstrained triangular grids, constrained quadrilateral grids, and mixed meshes are considered, with and without local time stepping (LTS), to determine the approach that maximizes computational efficiency defined as accuracy relative to computational effort. A mixed-mesh numerical scheme is introduced so all grids are processed by the same solver. Analysis focuses on a wide range of dam-break type test cases, where Godunov-type flood models have proven very successful. Results show that different mesh types excel under different circumstances. Cartesian grids are 2–3 times more efficient with relatively simple terrain features such as rectilinear channels that call for a uniform grid resolution, while unstructured grids are about twice as efficient in complex domains with irregular terrain features that call for localized refinements. The superior efficiency of locally refined, unstructured grids in complex terrain is attributable to LTS; the locally refined unstructured grid becomes less efficient using global time stepping. These results point to mesh-type tradeoffs that should be considered in flood modeling applications. A mixed mesh model formulation with LTS is recommended as a general purpose solver because the mesh type can be adapted to maximize computational efficiency.

1. Introduction

Flood inundation models predict the spatial distribution of flood depths and velocities, or flooding intensity [27]. This defines the severity of the hazard and associated impacts such as threats to public safety, potential for monetary losses, disruptions to critical lifelines (water supply, sanitary, power and transportation systems), and general disruptions of commerce [82]. Flood inundation models can be used in a forecasting mode to support emergency management, and in a planning mode to identify the most effective risk reduction measures through comparative analysis of the socio-economic and environmental consequences of each alternative [1,27]. Monetary losses are primarily a function of flood depth and duration [48,56,71], while structural damage is also a function of flood velocity and the associated flood inertial momentum flux per unit width which scales as ρhV^2 where ρ is the fluid density, h is the fluid depth, and V is the fluid velocity [31,43,44,56].

Depth-averaged shallow-water models offer an excellent foundation for flood prediction even with simplistic turbulent closures that lump all momentum losses into a resistance parameter [6]. Floods occurring on steep topography such as alluvial fans, with dam-break and levee-break flooding, and with coastal flooding from tsunami and storm surge transfer significant inertial fluxes and transition between a supercritical ($Fr > 1$) and subcritical ($Fr < 1$) state, where $Fr = V/(gh)^{1/2}$ represents the Froude number. Consequently models of these flows require a full momentum balance that accounts for local acceleration, gradients in inertial fluxes (convective acceleration), pressure gradients, gravitational effects, and friction. On the other hand, on relatively flat topography, flood flows are generally subcritical and the convective acceleration can sometimes be ignored [2,6,23]. This can be exploited to yield a model capable of relatively fast execution [22], but maintaining a complete momentum balance yields a more versatile model applicable to any possible Fr including flows with shocks [23,75].

Godunov-type models are tailored to high-inertia floods because of approximate Riemann solvers that account for transcritical flows with shocks [37,75], and the literature presents Godunov-type

* Corresponding author at: Department of Civil and Environmental Engineering, University of California, Irvine, CA 92697, USA. Tel.: +1 949 824 4327; fax: +1 949 824 3672.
E-mail address: bsanders@uci.edu (B.F. Sanders).
URL: http://sanders.eng.uci.edu (B.F. Sanders).

http://dx.doi.org/10.1016/j.advwatres.2014.02.013
0309-1708/

modeling studies of dam-break floods [9,30,53,67,77,81], coastal floods [25,32] and urban floods [39,57,66,67,76]. The success of Godunov-type flood models can be attributed to approximate Riemann solvers [37,75] which are also embedded in discontinuous Galerkin finite element schemes [4,26,45] and Boussinesq models that account for non-hydrostatic flow effects [47].

Godunov-type models have generally assumed either a structured mesh of quadrilateral cells [3,13,29,35,42,86] or an unstructured mesh of triangular cells [7,15,41,70,84]. The latter mandates greater overhead to track the neighborhood of data around each cell, and makes it more challenging to compute gradients in the solution because data points do not fall on a regular grid [7,41], but the unstructured mesh is very appealing for the ease with which meshes can be generated and tailored to the unique geometry of application sites and the ability to locally refine the mesh around areas of interest [11,21,54,57,66,67,76]. Adaptive mesh approaches also allow for local refinements, and take this a step further by adjusting refinements on the fly according to flow conditions [46,52]. Cut-cell models represent a variant of quadrilateral mesh models wherein any particular quadrilateral can be bisected so as to better constrain the mesh to the site geometry [17,18]. A few studies have also used a mixed-mesh Godunov-type shallow-water model, i.e., a model that can utilize a mesh of either triangular cells, quadrilateral cells, or any combination of triangles and quadrilaterals [1,19,57].

When designing a mesh, the goal is achieve the highest degree of accuracy for a given computational cost, our alternatively, the minimum computational cost to achieve a specified level of accuracy. Several factors contribute to accuracy of flood models including: (1) structural model errors, (2) input data errors, and (3) numerical errors. Structural model errors stem from limitations of the governing equations, in this case the depth-averaged shallow-water equations, such as the assumption that pressure is hydrostatic, the velocity distribution is unidirectional (not stratified or skewed), and the bed is fixed (not erodible). Structural model errors will depend on whether the model solves the full shallow-water equations or simplified forms of the 2D equations [2,22], uses a coupled 1D/2D approach for channel flows [5,12,49,83], or introduces a sub-grid topographic model [60] or sub-grid obstruction model [38,65]. Input data errors include errors in initial conditions, boundary conditions, and parameters such as the resistance parameters and, most importantly, elevation data. Numerical errors are linked to the chosen numerical method and include errors in the spatial and temporal discretization. For example, upwind schemes promote a diffusive error that can smear out sharp fronts (e.g., hydraulic jumps) if an overly coarse discretization is used.

Mesh design affects both input data errors and numerical errors. Topographic heights and resistance parameters are sampled at the vertices (or cell centers) of the mesh, and thus mesh design controls sampling. In areas of high topographic variability, a fine mesh may be required to resolve important flow paths and thus minimize input data errors. Similarly, a relatively fine mesh can be used to resolve sharp flow features such as a hydraulic jump and thus minimize numerical errors. A recent review of flood inundation modeling indicates that uncertainties in topographic data and hydrologic data remain the primary source of uncertainty in flood predictions [6], and a St. Francis dam-break modeling study concluded that reduction of topographic errors and numerical errors were both important for model accuracy [10]. Research also indicates that model structure can affect the sensitivity of model predictions to topographic errors [2].

Mesh design also drives computational cost, which for an explicit finite volume model scales in proportion to the number of cells N_C and the number of time steps N_T as follows,

$$\mathcal{C} \sim k_1 N_C N_T \tag{1}$$

where k_1 is a factor that depends on the numerical scheme, i.e., the spatial and temporal discretization and solution update procedure. The number of time steps depends on the mesh resolution as a result of the CFL condition given by [74],

$$\alpha = \frac{\lambda_{\max}\Delta t}{\Delta s} \leqslant \alpha_{max} \tag{2}$$

where α is the Courant number, α_{max} is the maximum Courant number for stability, Δt is the time step, Δs is the grid size, and $\lambda_{\max}$ is the maximum wave speed here given by $V + (gh)^{1/2}$. The stability limit α_{max} depends on the numerical method, and Δs is not easily defined for unstructured grids [74]. Nevertheless, computational costs will generally increase with mesh resolution as follows,

$$\mathcal{C} \sim k_2 \Delta s^{-3} \tag{3}$$

because $N_C \sim \Delta s^{-2}$ and $N_T \sim \Delta s^{-1}$ based on the CFL condition (Eq. (2)), where k_2 is a factor that depends on the numerical scheme.

Clearly, computational costs and accuracy are at odds with respect to mesh resolution. But what is the role of mesh type? Flood modeling domains typically include regions where a fine resolution is required to minimize topographic errors, such as a channel or raised embankment that constrains the spreading of flood water, and other regions where added resolution is not necessary, such as a floodplain. In this case, an unstructured mesh appears to be advantageous compared to a Cartesian grid because a fine resolution can be used selectively, thus minimizing N_C and reducing $\mathcal{C}$. On the other hand, when topographic heights are uniformly gridded as a Digital Terrain Model (DTM), and flow modeling proceeds at the resolution of the DTM, Cartesian grids appear to be advantageous because fewer cells can be used. For example, when each cell of a Cartesian grid is simply divided diagonally to make a triangular grid, $\mathcal{C}$ is magnified because both N_C and N_T are increased. While the added resolution of the triangular grid may help to reduce numerical error, there is no difference in the topographic error between the two grids so overall accuracy is unlikely to differ. To date, the effect of mesh type on the computational effort and accuracy of Godunov-type flood models has not been critically examined. With increasing urbanization and climate changes that threaten more extreme flooding, the need to model flooding in detail is growing on several fronts (e.g., forecast systems, risk reduction programs) and therefore a better understanding of how to develop computationally efficient models is needed.

1.1. A study of mesh-type tradeoffs

This paper presents a study on how computational costs and accuracy are affected by mesh type using Godunov-type flood models. The study is focused on the central question of which mesh type achieves the most accurate prediction (smallest errors) for the same computational expense, i.e., the maximum computational efficiency. Alternatively, the question can be viewed as which mesh design achieves a desired level of accuracy for the minimum computational effort. A mixed-mesh Godunov-type finite volume scheme is used to compare several mesh designs including Cartesian grids, unstructured grids of triangular cells, and mixed-meshes of triangular and quadrilateral cells. Further, the effect of mesh design on accuracy and computational cost is isolated by using the same computational engine in all cases. The eliminates biases that would otherwise result from different solvers and/or model structures.

The remainder of the paper proceeds as follows: Section 2 presents the governing equations and a brief description of the numerical method. A detailed presentation of the model is included as an Appendix. Section 3 presents applications ranging from channel flow problems to large scale flooding problems, and including idealized problems with exact solutions, laboratory-scale

test cases with comprehensive monitoring data, and field-scale test problems with limited (but critical) observational data. Section 4 presents a discussion that brings to light mesh-type tradeoffs, and Section 5 presents conclusions that include a quantitative characterization of model performance relative to mesh type.

2. Governing equations and methods

2.1. Porous shallow-water equations

Porous shallow-water equations developed to systematically account for solid flow barriers (e.g., buildings in urban areas) are the basis of the model and provide a foundation for both rural and urban applications [24,38,65,67,72]. The governing equations appear as integral statements of mass and momentum conservation for an arbitrary 2D domain Ω with boundary Γ and unit outward normal vector $\mathbf{n}$ as follows [65],

$$\frac{\partial}{\partial t}\int_{\Omega} i\mathbf{U}\,d\Omega + \oint_{\Gamma} i\mathbf{E}\cdot\mathbf{n}\,ds = \oint_{\Gamma} i\mathbf{H}\cdot\mathbf{n}\,ds + \int_{\Omega} i\mathbf{S}\,d\Omega + \int_{\Omega} i\mathbf{Q}\,d\Omega \tag{4}$$

where

$$\mathbf{U} = \begin{pmatrix} h \\ uh \\ vh \end{pmatrix} \quad \mathbf{E} = \begin{pmatrix} uh & vh \\ u^2h + \frac{1}{2}gh^2 & uvh \\ uvh & v^2h + \frac{1}{2}gh^2 \end{pmatrix} \tag{5}$$

$$\mathbf{H} = \begin{pmatrix} 0 & 0 \\ \frac{1}{2}gh|_{\eta_o}^2 & 0 \\ 0 & \frac{1}{2}gh|_{\eta_o}^2 \end{pmatrix} \quad \mathbf{S} = \begin{pmatrix} 0 \\ -(c_D^f + c_D^b)uV \\ -(c_D^f + c_D^b)vV \end{pmatrix} \tag{6}$$

$$\mathbf{Q} = \begin{pmatrix} Q \\ \frac{Q-|Q|}{2}u \\ \frac{Q-|Q|}{2}v \end{pmatrix} \delta(x - x_s,\ y - y_s) \tag{7}$$

where u = x-component of velocity, v = y-component of velocity, g = gravitational constant, $V = (u^2 + v^2)^{1/2}$, c_D^f is a ground friction drag coefficient, c_D^b is a drag coefficient for sub-grid scale flow obstructions, Q represents the rate of net inflow from point sources located at (x_s, y_s) which could account for streamflow inputs or losses to the storm sewer, $\delta()$ is the 2D Dirac delta function, and $h|_{\eta_o}$ is the depth corresponding to a piecewise constant water surface elevation η_o and piecewise linear ground elevation z within Ω [78,79]. This term is introduced to transform the classical ground slope source term to a boundary integral that preserves stationary solutions [78,79]. Based on the limits of this transformation, the momentum equations appearing in Eq. (4) are restricted to numerical schemes that are first- or second order accurate in space [65,78,79].

As indicated above, porous shallow-water equations account for sub-grid scale solid features that block flow. The motivation is to parameterize blockage effects systematically, based on feature geometry, without resolving features by mesh refinement as this would significantly degrades computational efficiency [19,39,65,67,72]. The model presumes that blockage features have been mapped for a study domain $D \in \mathbf{R}^2$ and saved as a geospatial data file, as is possible by processing aerial imagery or aerial lidar data. Further filtering then proceeds to identify the sub-grid scale subset of the blockage features, D_b, which is saved as a geospatial polygon file. The subdomain D_b is taken as a mask which gives rise to a binary density function defined as follows [65],

$$i(x,y) = \begin{cases} 0 & \text{if } (x,y) \in D_b \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

which defines regions of the study domain that support flood storage and conveyance as $i(x,y) = 1$. Two mesh-dependent porosity parameters follow from the density function (Eq. (8)) as follows,

$$\phi_j = \frac{1}{\Omega_j}\int_{\Omega_j} i\,d\Omega \quad \psi_k = \frac{1}{\Gamma_k}\int_{\Gamma_k} i\,ds \tag{9}$$

where Ω_j corresponds to the two-dimensional (2D) spatial domain of the jth computational cell and Γ_k corresponds to the kth computational edge of a mesh. Note that ϕ_j represents the fraction of a cell area occupied by voids, and ψ_k represents the fraction of a cell edge occupied by voids. Consequently, these parameters affect the relative storage of cells and conveyance between cells, respectively. Guinot [39] presents a related model for obstructions that also results in two porosity parameters, but the parameters are defined differently from Sanders et al. [65]. Schubert and Sanders [67] show that there are numerous strategies to account for solid barriers in models of urban flooding and each one offers certain advantages and disadvantages. With porous shallow-water equations as the foundation of this model, any of these alternatives can be implemented. Note that the governing equations (Eq. (4)) revert to the classical shallow-water equations in the limit that $i(x,y) = 1$ across all of D.

The ground friction drag coefficient can be parameterized in several ways including a Manning n_m, Chézy C, or Darcy–Weisbach f as follows,

$$c_D^f = \frac{gn_m^2}{h^{1/3}} = \frac{g}{C^2} = \frac{f}{8} \tag{10}$$

The building drag coefficient is computed similar to vegetative drag laws (e.g., [61]) as follows,

$$c_D^b = \frac{1}{2}c_D^o ah \tag{11}$$

where a represents frontal area. The unit of a is length^{-1}, corresponding to the frontal width of obstructions in Ω normalized by Ω. c_D^o represents the dimensionless obstruction drag coefficient, and values for 2D flows are tabulated [59]. Schubert and Sanders [67] provides guidance on the calculation of a and selection of c_D^o in practical applications.

The inflow term Q appearing in Eq. (4) accommodates an arbitrary set of point sources and sinks that could be associated with tributary inflow, surcharging of storm sewers or drainage into storm sewers. This feature has been utilized to predict urban dam-break flooding [30,67] and coastal urban flooding [33].

2.2. Numerical methods

Eq. (4) are solved using a Godunov-based finite volume scheme that allows for triangular, quadrilateral, or mixed meshes. The scheme blends elements of models previously reported by the authors including an implementation of Roe's approximate Riemann solver with a critical flow fix [13], an adaptive method of variable reconstruction for uneven topography that minimizes numerical dissipation [10], a local time stepping (LTS) scheme [64], and grid based porosity parameters for hard sub-grid scale flow barriers [65]. Additionally, the model uses a new, more efficient implementation of the Volume-Free Surface-Reconstruction (VFR) technique first developed by Begnudelli and Sanders [7,8]. A complete description of the model is presented as an appendix, but a brief description of the update scheme is presented here with an emphasis on LTS to faciliate interpretation of results which follow.

The essence of the scheme is an explicit update equation for cell j with edges $k = 1,\ K_j$ (K_j = 3 or 4) given by,

$$\mathbf{U}_j^{n+m_j} = \mathbf{U}_j^n - \frac{m_j \Delta t_o}{\Omega_j}\left(\sum_{k=1,K_j} \text{flux terms}\right) + \frac{m_j \Delta t_o}{\Omega_j}\,(\text{source terms}) \quad (12)$$

where n indicates the base time level and m_j indicates the number of base time steps of size Δt_o advanced in cell j. The base time step Δt_o is specified to ensure that the CFL condition (taken here as $\alpha \leqslant 1$) is satisfied globally. This is accomplished by defining a target Courant number, α_T, and computing Δt_o based on Eq. (A-12) assuming $\alpha_j = \alpha_T$ and $m_j = 1$. Each cell is then assigned the largest possible time step ($m_j \Delta t_o$, $m_j = 1, 2, 4, 8$, etc.) that satisfies the CFL condition locally, and the assignment is adjusted as flow conditions change (see Appendix for a complete description). LTS reduces run times most effectively on variable resolution meshes, but it may also have an effect on uniform resolution meshes where the flow dynamics create spatial variability in maximum wave speeds [64]. The impact of LTS on mesh type tradeoffs is explicitly examined in this study.

Note that the CFL condition given by $\alpha_j \leqslant 1$ is only an approximation, because stability limits for nonlinear problems on unstructured grids cannot be precisely stated [74]. In a wide range of test problems beyond those considered here, stable solutions have been obtained with a Courant number slightly larger than one, but an unstable solution has never been encountered for $\alpha_j \leq 1$. Hence, $\alpha_j \leqslant 1\ \forall j$ is taken as a sufficient condition for stability.

2.3. Performance metrics

In the testing that follows, an L_1 norm is used to quantify differences between two scalar time series, w_1 and w_2, of size N as follows,

$$L_1(w_1, w_2) = \sum_{j=1,N} \frac{|(w_1)_j - (w_2)_j|}{N} \quad (13)$$

Additionally, wall-clock execution times are measured to quantify computational effort. Model runs were all completed using a 3.07 GHz Intel® CoreTM i7 CPU with 8 GB RAM. The *computational efficiency* of the alternative mesh designs is revealed by plots of L_1 errors versus run times.

3. Model applications and results

A series of model applications are presented in this section to quantitatively examine computational efficiency by mesh type. The applications begin with idealized test cases with exact solutions to characterize numerical errors, and then continue with laboratory and field-scale test cases to measure the combined effect of numerical errors, input data errors, and structural model errors. Testing emphasizes highly unsteady flows (dam-break type flows) because Godunov-schemes have proven most popular and successful in this context.

3.1. Dry-bed and wet-bed dam-break problem

These classic problems in computational hydraulics test whether a model is capable of resolving trans-critical flows with shocks without spurious oscillations or excessive numerical diffusion. Importantly, the problems have exact solutions [75], so numerical truncation errors can be measured with precision.

The test problem is configured as a 1 km long and one-cell wide channel with 1 m deep water held behind a dam located at the channel midpoint. Two variants are considered, one with a dry downstream channel condition and a second with a wet (10 cm) downstream channel condition. Cartesian and triangular mesh types are used in each case. The triangular mesh shares the same vertices as the Cartesian mesh, but uses twice the number of cells because each square is divided into two triangles. To examine the effect of refinement, the grid size (Δx and Δy) is halved and then halved again leading to a coarse, medium, and fine mesh with properties shown in Table 1. In all cases, the solution was advanced for 70 s using a target Courant number $\alpha_T = 0.8$. Additionally, separate runs were completed using LTS = 1 and 3.

The fine mesh predictions at $t = 70$ s using LTS = 1 for dry and wet cases are shown in Fig. 1(a) and (b), respectively, showing that both mesh types support a qualitatively correct solution. Errors across all meshes are listed in Table 1 and plotted in Fig. 2 versus run times to reveal the computational efficiency of the mesh design. These results indicate that:

(1) Numerical errors are a factor of 2–3 smaller in the wet-bed case than the dry bed case, regardless of mesh type. This is attibuted to the singularity in depth at the wet/dry interface.
(2) The Cartesian grid is only about 10–20% more efficient (errors relative to run times) than the triangular grid in the wet-bed problem; on the other hand, in the dry bed problem, the Cartesian grid is as much as three times more efficient (factor of 3 difference in run time for same numerical error).
(3) Use of LTS reduces run times by 20%, on average, compared to global time stepping (LTS = 3 vs. 1). Additionally, LTS reduces run times on both triangular and Cartesian grids.

To summarize, this 1D test problem shows that both triangular and quadrilateral grid models converge to exact solutions with refinement, but the quadrilateral mesh type is generally more efficient, with an advantage that varies from 10–300% across the test conditions considered here.

3.2. Uniform flow in a trapezoidal channel

This test problem illuminates a critical issue in 2D flood inundation modeling: capturing non-rectangular channel geometry with a 2D mesh. Rectangular channels are straightforward because of vertical side walls, but the sloping walls of non-rectangular channels introduce a wet/dry interface. Generally, 2D models capture topographic data at mesh vertices (or cell centers in some models), unless the model includes a sub-grid scale channel capability designed to capture more detailed topographic information [55,60,85]. In the present model, it is assumed that topography varies linearly between vertices, so it is possible to perfectly model a trapezoidal channel with three cells, but only if the cell vertices are aligned with the corners of the channel. If vertices are not constrained by the channel geometry, and fall randomly in accordance with an unconstrained mesh generation process, then the mesh design introduces an error in the topographic data (despite the fact that channel geometry is known exactly). The topographic error can be minimized through refinement, but a more elegant approach is to constrain the mesh so vertices are aligned with corners of the channel which in turn enables use of a relatively course mesh whose precise resolution is constrained by the tolerable level of numerical error.

Here, a 1000 m long trapezoidal channel is considered with a bottom width of 5 m and a side slope of 1:2. The longitudinal channel slope is 0.001, and friction is modeled with $n_m = 0.012\ \text{m}^{-1/3}$ s. A constant discharge, $Q = 10\ \text{m}^3/\text{s}$, is imposed at upstream boundary and the uniform (or normal) depth equal to 0.8 m is imposed at the downstream boundary. The exact solution to this problem is simply a constant depth of 0.8 m along the length of the channel.

Table 1
Properties of meshes, run times and L_1 norms for classical dam-break problem.

Grid	$\Delta x = \Delta y$ (m)	# Of nodes	# Of cells		Case	LTS = 1				LTS = 3				L_1 (cm)	
						Time step (s)		Run time (s)		Time step (s)		Run time (s)			
			Cartes.	Tri.		Cartes.	Tri.	Cartes.	Tri.	Cartes.	Tri.	Cartes.	Tri.	Cartes.	Tri.
Coarse	1.00	2002	1000	2000	Dry	0.1498	0.0705	0.187	0.500	0.1875	0.0791	0.125	0.375	0.320	0.280
					Wet	0.1865	0.0783	0.187	0.515	0.1610	0.0776	0.156	0.452	0.168	0.117
Medium	0.50	4002	2000	4000	Dry	0.0746	0.0352	0.592	1.670	0.0937	0.0395	0.421	1.264	0.262	0.237
					Wet	0.0932	0.0392	0.593	1.841	0.0765	0.0386	0.562	1.591	0.095	0.067
Fine	0.25	8002	4000	8000	Dry	0.0373	0.0176	2.122	5.975	0.0468	0.0198	1.482	4.384	0.122	0.097
					Wet	0.0466	0.0196	2.153	6.786	0.0400	0.0194	1.919	5.631	0.059	0.039

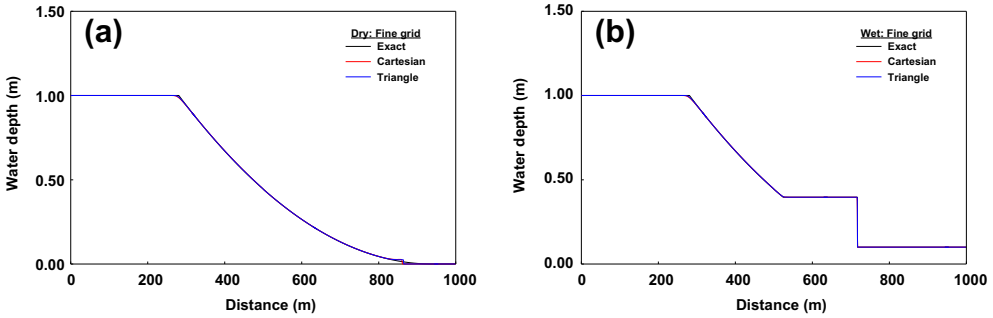

Fig. 1. Comparison of water depth predictions and exact solutions for classical dam-break problem in a (a) dry bed and (b) wet bed.

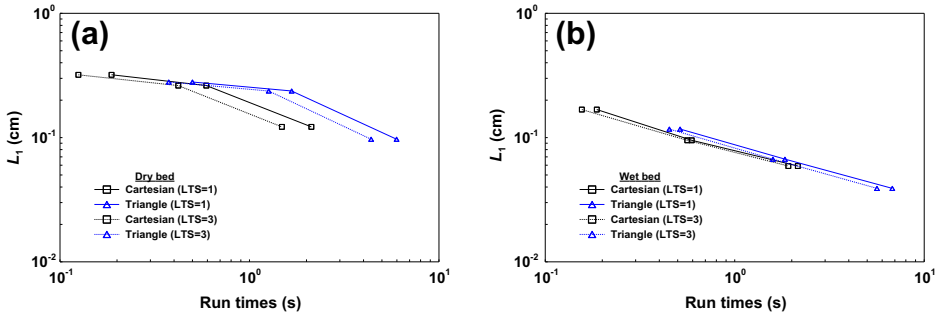

Fig. 2. Errors in classical dam-break water depth predictions versus run times in a (a) dry bed and (b) wet bed.

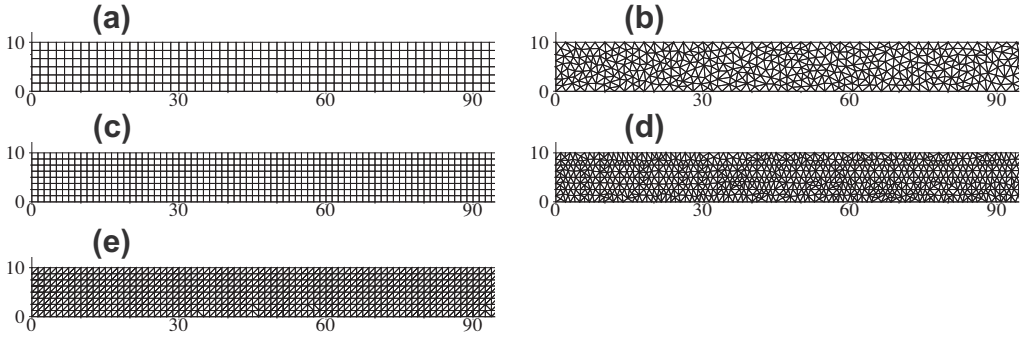

Fig. 3. Five mesh types for uniform flow test case. (a) Unconstrained Cartesian grid, (b) Unconstrained Delaunay mesh, (c) Constrained Cartesian grid, (d) Constrained Delaunay grid and (e) Constrained right-triangle grid (unit: m).

The performance of five mesh types is examined as shown in Fig. 3 and Table 2: (1) an unconstrained (vertices are not aligned with the corners of the channel) Cartesian grid, (2) an unconstrained Delaunay grid, (3) a constrained (vertices aligned with channel corners) Cartesian grid, (4) a constrained Delaunay grid and (5) a constrained right-triangle grid obtained by diving each square of the constrained Cartesian grid into two triangles. The solution was integrated using $\alpha_T = 0.8$ for a period of 1 h in all cases using LTS = 1.

Fig. 4(a) shows the longitudinal profile of the water surface predicted by the model using each of the mesh types. The constrained Cartesian grid and the constrained right triangle grid each yield the

Table 2
Properties of meshes, run times and L_1 norms for uniform flow in a trapezoidal channel.

Mesh	Grid	Resolution (m)	# Of nodes	# Of cells	Time step (s)	Run time (s)	L_1 (cm)
Unconstrained Cartesian	Coarse	3.33	1204	900	0.361	3.03	26.21
	Medium	1.67	4207	3600	0.222	12.73	6.86
	Fine	0.83	15613	14400	0.114	82.31	4.73
Unconstrained Delaunay tri.	Coarse	3.32	1371	2222	0.098	19.67	54.37
	Medium	1.66	4960	8882	0.044	127.02	48.33
	Fine	0.83	18813	35555	0.022	995.5	37.38
Constrained Cartesian	Coarse	2.5	2005	1600	0.342	4.56	4.43
	Medium	1.25	7209	6400	0.175	25.12	2.12
	Fine	0.625	27217	25600	0.085	205.17	1.45
Constrained Delaunay tri.	Coarse	2.5	2496	3996	0.116	23.82	4.80
	Medium	1.25	8864	15963	0.036	224.69	13.56
	Fine	0.622	33576	63897	0.014	3102.84	7.54
Constrained right tri.	Coarse	2.5	2005	3200	0.149	14.24	4.81
	Medium	1.25	7209	12800	0.075	89.43	1.74
	Fine	0.625	27217	51200	0.037	815.36	1.62

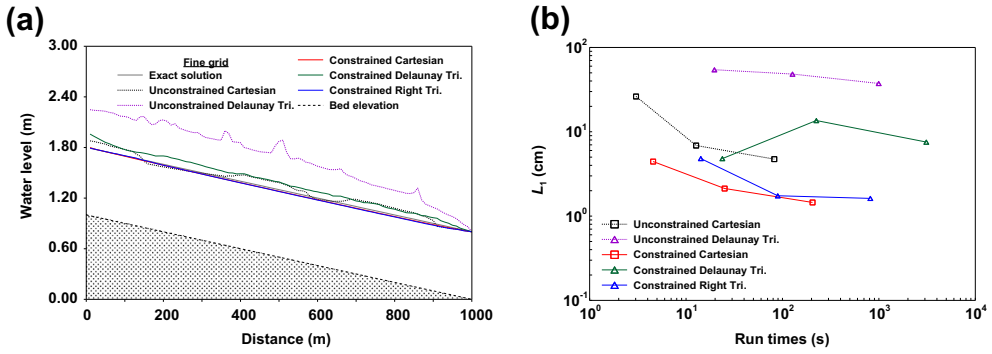

Fig. 4. Uniform flow test case. (a) longitudinal profile of the predicted water surface and (b) errors versus run times.

greatest accuracy, while the largest errors are associated with the unconstrained Delaunay grid which is attributed to the inaccurate and irregular topographical representation of the channel compared to the other grids. The unconstrained Cartesian grid also introduces errors in the topography, but the errors are regular and thus do not generate the same degree of error in the free surface height prediction. It is noted that the irregularities in the water surface predictions with unconstrained grids shown in Fig. 4(a) are small-scale transients that are attributed to partially wetted cells which transition between fully wetted and partially wetted states. Hence, an advantage of the constrained meshes is that this error is minimized.

Fig. 4(b) shows the convergence properties of each of the five mesh types considered, with errors plotted versus run times. Errors and run times are also presented in Table 2. These results show that the errors of the constrained Cartesian grid and the constrained right-triangle grid are more than an order of magnitude smaller than the unconstrained Delaunay grid, and several factors smaller than the constrained Delaunay grid. These results also show that the constrained Cartesian grid offers the greatest computational efficiency of all the mesh types considered, minimally 2–3 times better than the best results achieved with a triangular grid. Since run time scales with the resolution of the mesh, Fig. 4(b) also shows that errors are reduced with refinement using regular meshes but not necessarily with Delaunay meshes. In particular, with the constrained Delaunay mesh, the coarsest resolution actually produced smaller errors than medium and fine resolution meshes. This is attributed to the irregularity of the Delaunay meshes, which (by chance) may offer a good representation of the channel geometry and minimal errors in the free surface height.

To summarize, this test problem shows that Cartesian meshes are best suited to modeling flows in prismatic channels, and unconstrained Delaunay meshes represent the least accurate alternative. Highly accurate predictions are possible with regular triangular grids, but these are not as efficient as Cartesian grids. Importantly, that test shows the critical importance of constraining meshes to the channel geometry for accuracy purposes.

3.3. Dam-break flow in a rectangular channel with friction and an uneven bottom

Now several test cases involving a comparison to physical observations is considered. These cases introduce structural model errors in addition to input data and numerical errors. This first test involves a set of three experiments (Cases 1–3) involving dam-break flow in a horizontal hydraulic flume fitted with a triangular sill downstream of a gate (hypothetical dam), as shown in Fig. 5 [58]. Following the rapid opening of a gate that simulates a dam-break, a shock wave propagates downstream, partially reflects off the sill, forms of a hydraulic jump, and there is complex wave action associated with various reflections and wave interactions. As shown in Fig. 5, the channel is 38 m in length and the dam is located 15.5 m from the upstream end of the channel. The triangular sill is 0.4 m high, 6 m long, and located 13 m downstream from the dam. Four gages shown as G4, G10, G13 and G20 in Fig. 5 measure water levels that are taken as the reference solution. All three cases involve an initial depth of 0.75 m upstream of the dam. Case 1 involves a dry channel and a free-overfall downstream boundary condition. Case 2 involves a dry channel immediately below the dam, a flooded channel with a depth of 0.15 m downstream of

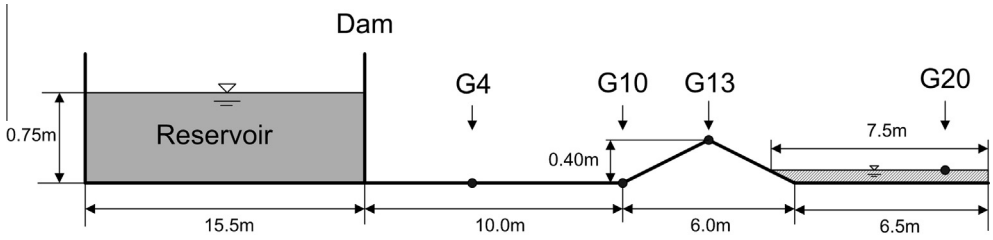

Fig. 5. Layout of dam-break flow with an uneven bottom.

the sill, and a vertical wall as the downstream boundary. Case 3 is the same as Case 2, except that a 0.15 m high weir is placed at the downstream boundary. Resistance is modeled by a Manning n_m= 0.0125 $m^{-1/3}$ s [58]. Boundary conditions are enforced with the boundary condition flux function described by Sanders [63]. The model duration in all cases is 40 s, beginning with the removal of the gate.

Each test case is modeled at three levels of refinement (Δx = 0.5, 0.25 and 0.125 m) and with Cartesian and triangular grid types, as shown in Table 3 and Fig. 6. Each test case is also modeled with an ulta-fine resolution Cartesian grid (Δx = 0.01 m) to enable measurement of the numerical errors in addition to total errors enabled by the laboratory measurements. The solution is advanced in all cases using a target Courant number, $\alpha_T = 0.8$, and global time stepping (LTS = 1). The Cartesian and triangular meshes share the same vertices at the same resolution, so the triangular meshes contain twice as many cells and require a smaller time step (roughly half the size of the rectangular mesh) to achieve the same target Courant number.

Results are qualitatively good in all cases, as indicated by Case 3 results shown in Fig. 7. The major wave forms are resolved with good agreement in both amplitude and phase, although the wave amplitude is over-predicted by about 10% near sharp fronts in several instances. Also, small oscillations are predicted by the fine, triangular mesh at Gage 10. Graphically, the results are similar across Cases 1–3, so only Case 3 results are shown in Fig. 7. Total errors (measured against measurements) and numerical errors (measured against ultra-fine grid prediction) are plotted versus run times for each mesh type in Fig. 8(a)–(c) for Cases 1–3, respectively. Errors are also presented in Table 3. The numerical error show that both mesh types converge with refinement, and that the Cartesian mesh type achieves a numerical error that is 2–3 times smaller than the triangular grid for the same computational effort. Hence, the Cartesian mesh type is 2–3 times more computationally efficient, as defined here.

Plots of total errors shown in Fig. 8 point to a structural model error or input data error, because models predictions do not converge to precisely to measurements. This is important to consider in applications, because the added work to minimize numerical errors may not be justified based on the total error.

To summarize, this test case shows that the Cartesian and triangular grids achieve similar accuracies when measured by total errors, but that the Cartesian grid minimizes numerical errors using less computational effort than the triangular grid. Importantly, this test shows that overall accuracy does not continuously improve with refinement, and computational effort, due to structural model errors.

Table 3
Properties of meshes, run times and L_1 norms for dam-break flow with an uneven bottom.

Grid	# Of nodes	# Of cells		Case	Time step (s)		Run time (s)		L_1 (cm)			
									Total error		Numerical error	
		Rect.	Tri.		Rect.	Tri.	Rect.	Tri.	Rect.	Tri.	Rect.	Tri.
Coarse	154	76	152	Case 1	0.074	0.031	0.125	0.349	3.828	3.479	1.038	1.206
				Case 2			0.131	0.359	4.446	4.656	2.169	2.108
				Case 3			0.124	0.349	3.440	3.823	1.447	1.345
Medium	306	152	304	Case 1	0.049	0.022	0.249	0.661	3.415	3.609	0.713	0.733
				Case 2			0.259	0.677	3.378	4.035	1.188	1.353
				Case 3			0.259	0.681	3.226	3.556	0.765	0.807
Fine	610	304	608	Case 1	0.024	0.012	0.483	1.372	3.324	3.257	0.409	0.342
				Case 2			0.483	1.404	3.346	3.710	0.717	0.606
				Case 3			0.484	1.404	3.274	3.561	0.449	0.381

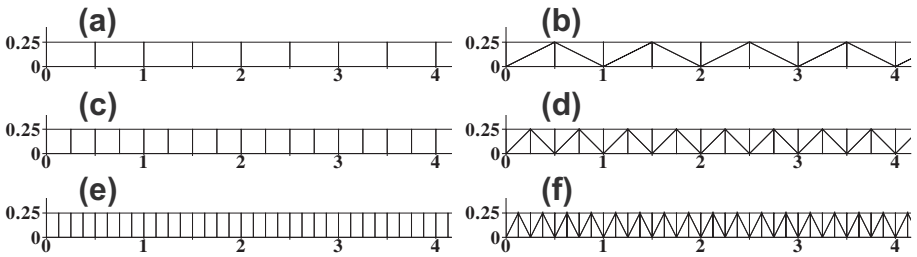

Fig. 6. Cartesian and triangle grid with (a) and (b) x = 0.50 m, (c) and (d) 0.25 m, and (e) and (f) 0.125 m (unit: m).

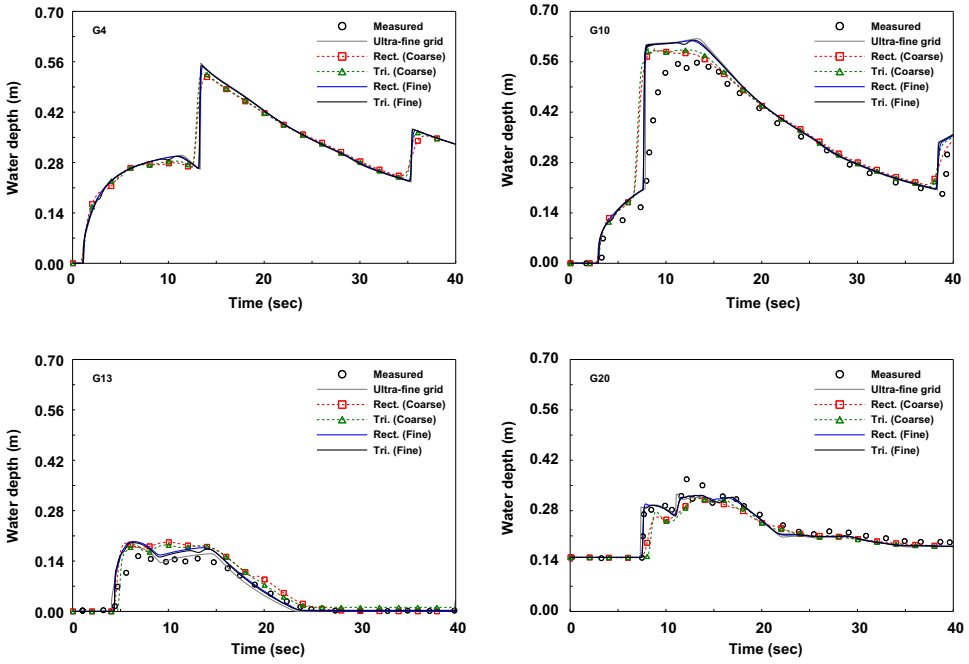

Fig. 7. Comparison of water depth predictions and measurements (Case 3) in the dam-break flow with an uneven bottom.

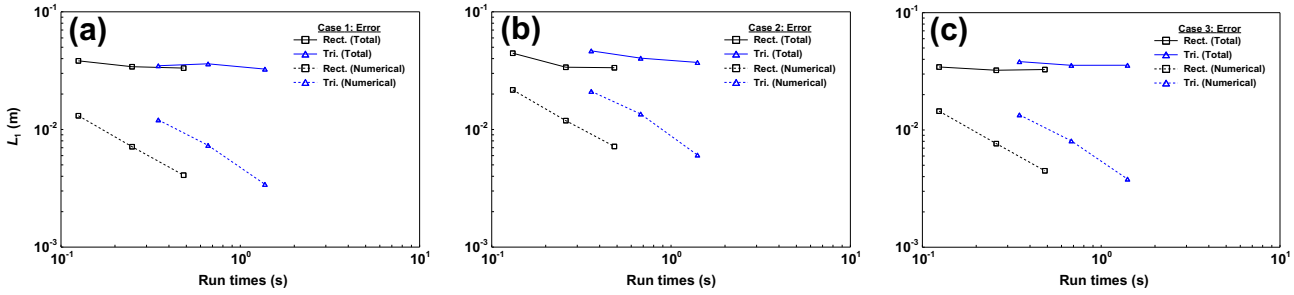

Fig. 8. Total and numerical errors versus run times for dam-break flow with an uneven bottom: (a) Case 1, (b) Case 2 and (c) Case 3.

3.4. Flash flooding of a mountainous valley

Test cases thus far have utilized engineered channel geometries, while floods of practical interest involve natural terrain and interactions with built structures. Thus, a scaled physical model study of a hypothetical, catastrophic flood in the Toce River valley in Italy is considered next [73]. The physical model captures the main topographic features of the valley, and a cluster of blocks affixed to the channel bottom simulate the blockage effects of an urban district. Discharge is controlled at the upstream boundary of the valley to simulate a flash flood, and water drains freely at the downstream boundary. An array of gages placed in the physical model measure water depth time series as shown in Fig. 9(a) and (b). As shown in Table 4, two cases termed "low flow" by Testa et al. [73] are examined here: Case 1 corresponds to aligned blocks and Case 2 corresponds to staggered blocks.

Classical shallow-water models have been applied to this test case and have achieved good overall accuracy, but a structural model error has been identified by errors as large as 50% that did not decrease with mesh refinement [38,72,65]. Thus, numerical error is trumped by structural model error in this test case. Nevertheless, a remaining goal is to realize the respectable accuracy potential of shallow-water models with the least computational effort. Use of porosity parameters has proven extraordinarily successful, yielding models that execute order of magnitude faster than classical shallow-water models without loss of accuracy [38,65,72]. The mesh design question pursued here is whether triangular grid or quadrilateral grid models support a higher level of computational efficiency.

Four meshes are created for this test case as shown in Fig. 9(c)–(f). This includes two meshes for each test case: (1) a triangular mesh constrained by the channel banks and block centroids, and (2) a mixed mesh of quadrilaterals in the center of the channel, constrained by block centroids, with triangles lining the banks to conform to the irregular bank geometry. Mesh vertices are constrained by block centroids to ensure that mesh edges span gaps between buildings and properly convey fluxes, as shown in Fig. 9. In this case, there is no need for refinement since porosity models are designed to run at a coarse resolution that matches the size of gaps between buildings [65].

Topographic heights were estimated at mesh vertices by inverse-distance weighted (IDW) interpolation from point elevation data provided by Testa et al. [73], a uniform Manning $n_m = 0.0162\ \mathrm{m}^{-1/3}\ \mathrm{s}$ was assigned to each cell, and porosity values ϕ_j and ψ_k were computed for each cell and edge, respectively, in accordance with the fraction of the area and edge, respectively, that is not covered by an obstruction [65]. To parameterize building drag using Eq. (11), a uniform drag coefficient was used $c_D^o = 1$

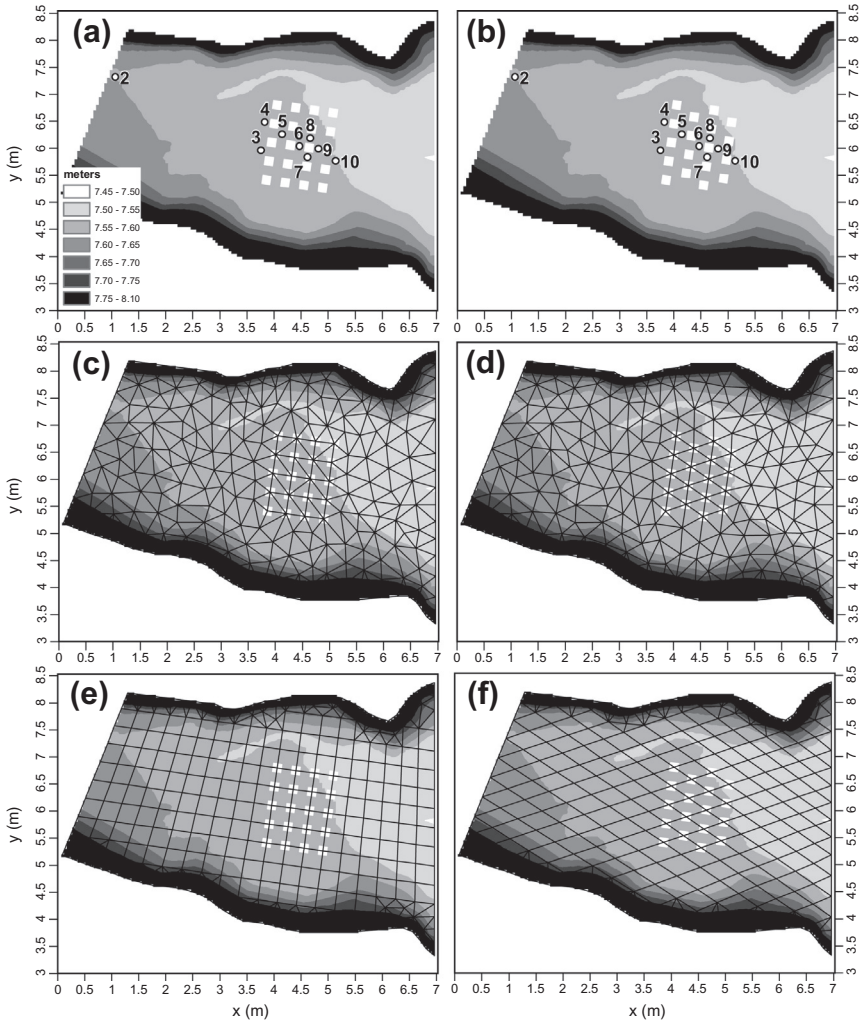

Fig. 9. Toce valley test case configuration. (a) and (b) show building configurations for the aligned and staggered scenarios respectively, as well as bed elevations and monitoring station locations. (c) and (d) show triangular and (e) and (f) show mixed gap-conforming meshes to solve the porous shallow water equations.

Table 4
Properties of meshes, run times for Toce valley test case.

Case	Configuration	Boundary condition		Mesh	# Of nodes	# Of cells			Time step (s)	Run time (s)
		Upstream	Downstream			Total	Tri.	Quad		
Case 1	Aligned	Low flow[a]	Open	Tri.	243	412	412	0	0.010	1.11
				Mixed	303	346	164	182	0.026	0.34
Case 2	Staggered	Low flow[a]	Open	Tri.	247	420	420	0	0.010	1.10
				Mixed	285	322	170	152	0.021	0.29

[a] Testa et al. [73].

and a distributed frontal area parameter a_j was computed based on block width within each cell [65]. Flow is supercritical as it enters the upstream boundary, so the discharge is specified in accordance with a "low flow" time series of discharge, and the depth is specified based on measurements at Gage 2 located at the upstream boundary (Fig. 9) [73]. At the downstream boundary condition, a "soft" boundary condition is used so the flow exits freely [63]. Table 4 summarizes mesh properties including the time step used for each.

Figs. 10 and 11 present model predictions and laboratory measurements for aligned and staggered test cases, respectively. Qualitatively, model predictions and measurements compare favorably with errors less than about 50% (e.g., Gage 5 in the aligned case), similar to previous models based on the classical shallow-water equations [38,65,72]. L_1 norms are presented in Table 5 for the aligned and staggered cases. The mixed mesh predictions generally compare better with gage data than triangular mesh predictions in the aligned case, while the mixed mesh and triangular mesh predictions are of a similar accuracy in the staggered case. However, predictions are also sensitive to the building drag parameter c_D^o [65]. Taking into consideration that this could range from nearly zero to about 2 (it is effectively a calibration parameter), then neither of the mesh designs stand out as being more accurate. Focusing now on computational efficiency, Table 4 indicates that the mixed mesh predictions execute about three to four times faster than the triangular mesh predictions. This is

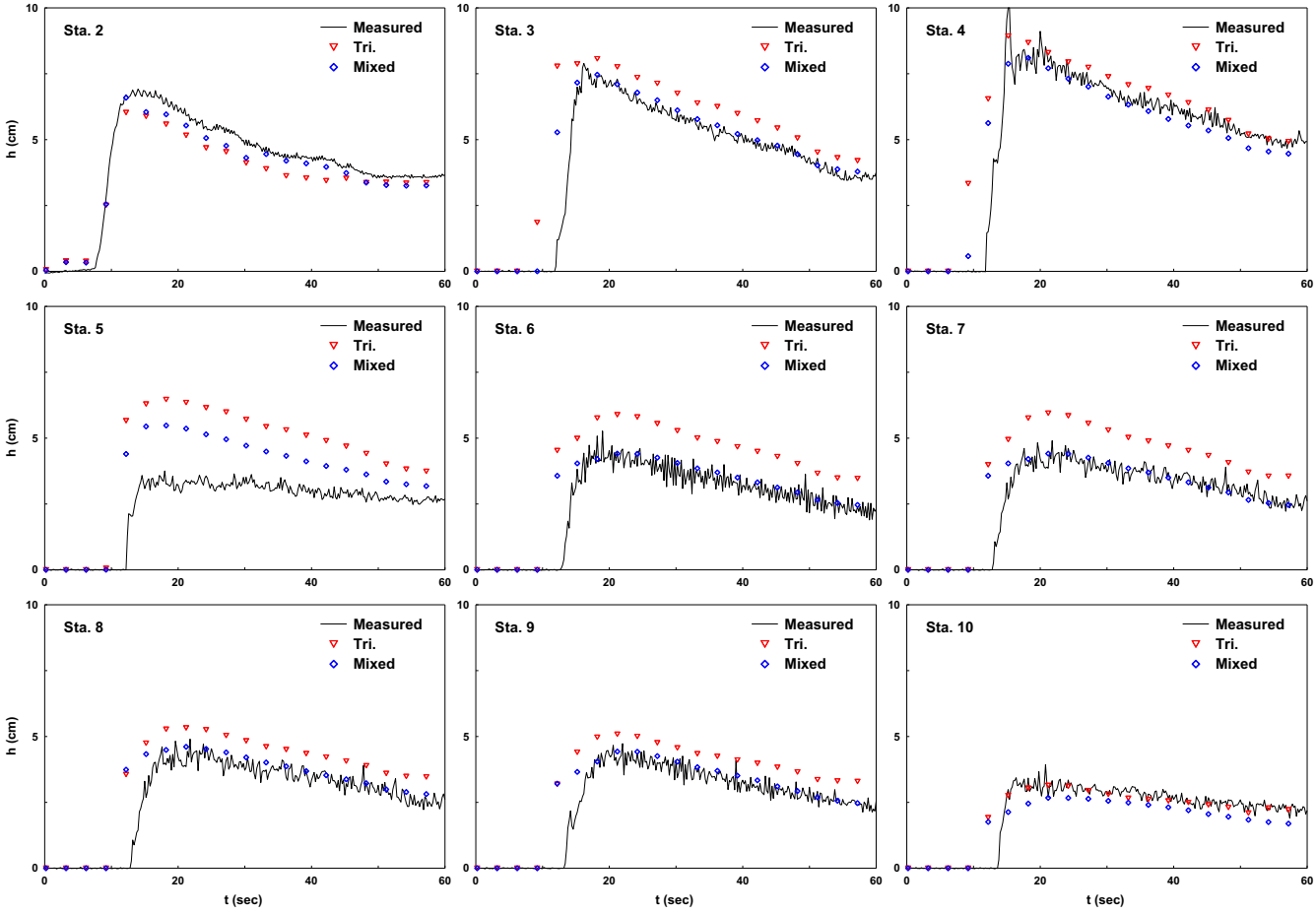

Fig. 10. Comparison of water depth predictions and measurements for aligned blocks in Toce valley test case.

attributed mainly to a differences in the time step and number of cells. For the same value of $\alpha_T = 0.8$, the mixed-mesh model uses a time step Δt_o that is at least twice as large. Additionally, the channel is discretized with about 20% fewer quadrilaterals than triangles.

These results show that a comparable level of accuracy is achieved by both mesh designs, but the mixed mesh consisting mainly of quadrilateral cells is far more computationally efficient because it executes three to four times faster. Also, it is noted that a purely quadrilateral mesh would be more difficult to set up for this test case than a mixed mesh because of the natural topography and block constraints (a Cartesian mesh would be impossible), highlighting a practical advantage of the mixed-mesh model formulation.

3.5. Malpasset dam-break flood, France

Field-scale performance is the ultimate goal of flood inundation models, and here attention turns to the Malpasset dam-break flood which is one of the most-studied historical events from a 2D modeling perspective [16,34,40,46,50,51,69,77,84]. The Malpasset dam was a concrete arch structure located near Fréjus, France and when it failed completely and catastrophically in 1959, a massive flood rushed down the Reyran valley and into Mediterranean Sea. Field validation data includes seventeen high water marks surveyed by police and the shutdown time of three electric transformers [36]. A 1:400 scale physical model study of the flood was also completed by EDF-LNH in France in 1964 and nine gages placed in the model recorded the maximum water levels and the flood arrival time [36]. Fig. 12(a) shows site topography and field and laboratory measurements locations.

Topography data used in previous modeling studies is also used here, and consists of a digitized set of 13,541 points taken from a historical 1:20,000 scale topographic map and spaced from 6 to 450 m apart [36]. A 5 m resolution raster DTM was created from these points using the terrain-to-raster tool in ArcGIS (ESRI, Redlands, CA, USA), which was found to be preferable to a TIN structure for representing the channel thalweg. The reservoir height at the time of failure was 100 m, and time measurements are relative to moment of failure so $t = 0$ corresponds to the initiation of fluid motion [36]. Friction is parameterized with a constant Manning n_m = 0.033 $m^{-1/3}$ s following previous studies [16,34,46,51,69,77,84].

Three mesh types are examined including a Cartesian grid, a constrained Delaunay mesh (triangular grid), and a mixed mesh of triangular and quadrilateral cells. Additionally, each mesh type was applied at three resolutions to measure how accuracy varies with run times and determine the approach that offers the highest computational efficiency. Total errors are measured with a comparison to field and laboratory measurements of high water marks and flood arrival times, and a "numerical" error is measured by comparison to an ultra-fine resolution (5 m) Cartesian grid prediction. Note that the "numerical" error actually reflects both numerical truncation errors and input data errors, since the sampling of topography increases with refinement. Hydrograph predictions at cross-sections shown in Fig. 12(b) are also compared to illuminate differences in flooding dynamics captured by the models.

Fig. 12(b) shows the model domain divided into five refinement zones, which are used to guide unstructured mesh generation

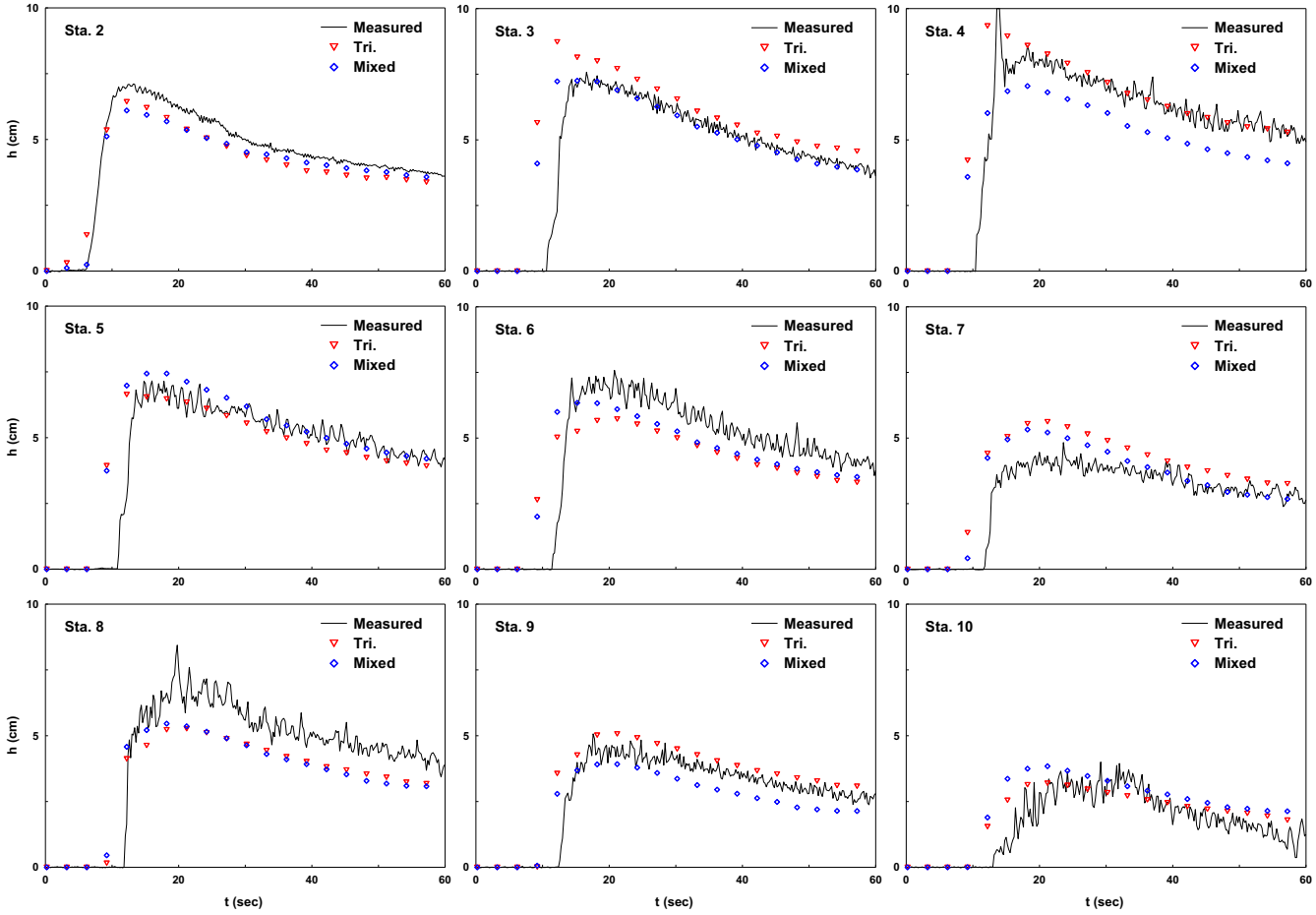

Fig. 11. Comparison of water depth predictions and measurements for staggered blocks in Toce valley test case.

Table 5
L_1 norms for Toce valley test case.

Reference	L_1 (cm): aligned			Reference	L_1 (cm): staggered		
	Measured	Tri.	Mixed		Measured	Tri.	Mixed
Mea.	0	0.970	0.616	Mea.	0	0.746	0.724
Tri.	x	0	0.631	Tri.	x	0	0.506
Mixed	x	x	0	Mixed	x	x	0

using Triangle [68] for triangular meshes and SMS (Aquaveo, Provo, Utah, USA) for mixed meshes. A uniform resolution is used within each zone. This allows a finer resolution to be used along the narrow Reyran valley, and a coarse resolution on the wide floodplain near the Mediterranean Sea, leading to the mesh properties shown in Table 6. Following mesh generation, topographic heights are assigned to vertices of the computational mesh from the 5 m DTM by IDW interpolation. Fig. 13 shows the coarse mesh versions of the three mesh types, and Table 6 summarizes key mesh and model properties. Models are all executed using a target Courant number, $\alpha_T = 0.8$. Runs are completed using LTS = 1 and LTS = 3, allowing explicit consideration of LTS with respect to the most efficient computational method. The flood was modeled for a duration of 1 h.

Fig. 14 shows model predictions of high water marks of all mesh type using LTS = 1, alongside field and scaled laboratory data. Qualitatively, model predictions are similar to results of previous modeling studies, e.g., [16,40,77], wherein flood height is generally over-predicted at P1 and P13 and under-predicted at P4, P5, P7 and P8 compared to field observations. When compared to scaled laboratory measurements, models have generally over-predicted flood height at S7 and S9 and under-predicted flood height at S11 and S12.

Fig. 15 shows L_1 errors of water height predictions versus run times for all cases, including LTS = 1 and 3. Fig. 15(a) (and Table 7) shows total errors based on a comparison to field and gage measurements, and Fig. 15(b) shows the numerical error based on a comparison to the ultra-fine Cartesian grid prediction. Focusing first on Fig. 15(a), results show that coarse resolution triangular and mixed meshes support *ca.* 25% smaller errors than a coarse Cartesian grid, and that there is little benefit to unstructured mesh refinement because total errors are not further reduced even though computational times increase. On the other hand, the Cartesian grid prediction benefits slightly from refinement from 40 to 20 m, bringing errors in line with those achieved by the unstructured meshes. Fig. 15(a) also shows that local time stepping (LTS = 3) significantly reduces unstructured grid run times compared with global time stepping (LTS = 1), with a neglible impact

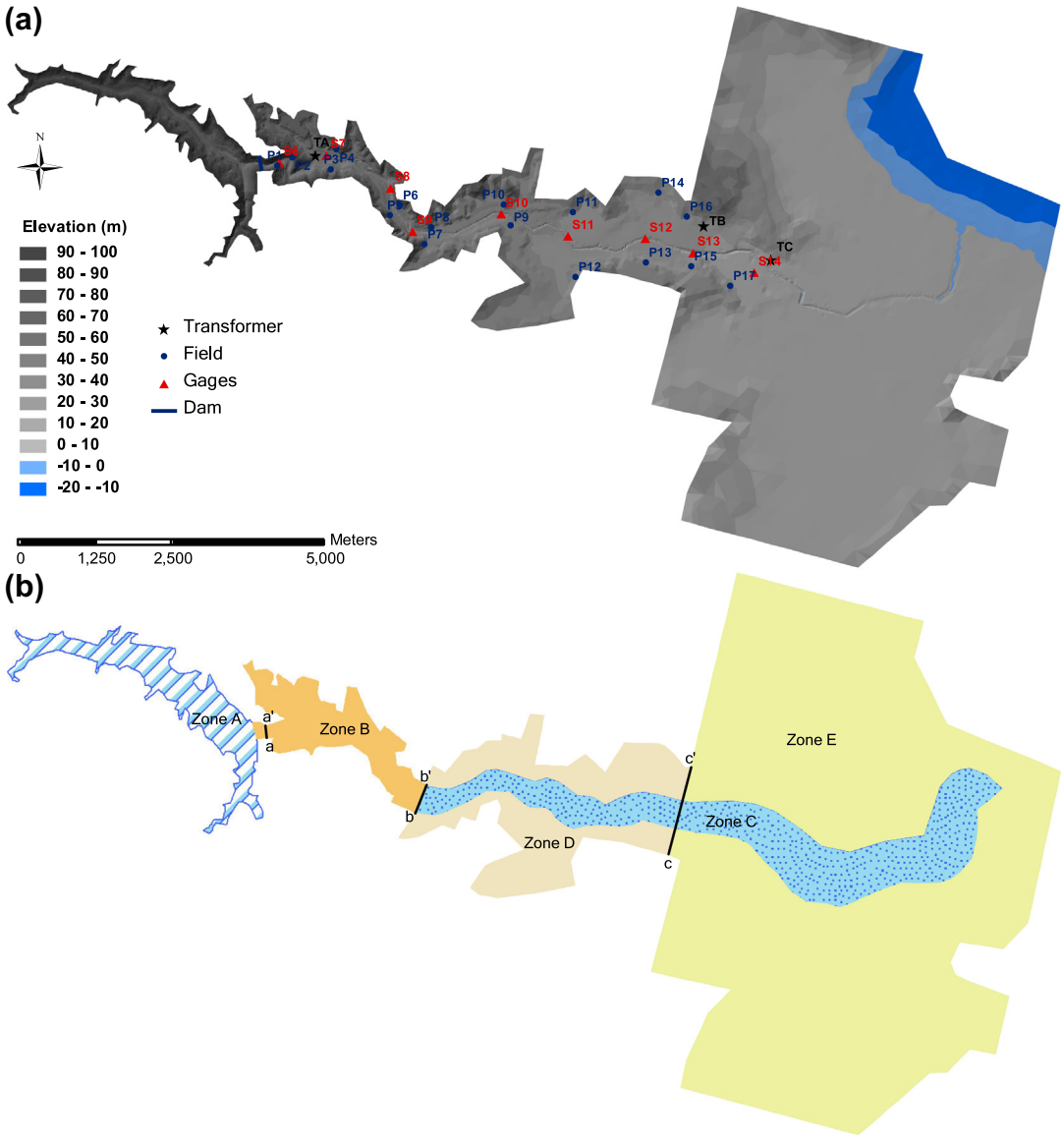

Fig. 12. Malpasset dam-break flood. (a) Topography and field and laboratory measurements locations and (b) model domain divided into five refinement zones and cross-sections for discharge predictions.

Table 6
Properties of meshes, run times for Malpasset dam-break flood.

Mesh	Grid	Zone resolution (m)					# Of nodes	# Of cells	LTS = 1		LTS = 3	
		A	B	C	D	E			Time step (s)	Run time (s)	Time step (s)	Run time (s)
Cartesian	Ultra-fine	5	5	5	5	5	2079137	2069924	–	–	–	–
Tri.	Coarse	60	28	40	80	120	17570	33983	0.218	72.34	0.215	27.54
	Medium	30	14	20	40	60	69066	135824	0.101	651.97	0.101	224.17
	Fine	15	7	10	20	30	273825	543024	0.047	5729.72	0.047	1877.97
Mixed	Coarse	60	28	40	80	120	17570	23803	0.232	47.35	0.230	19.98
	Medium	30	14	20	40	60	69066	93001	0.107	493.98	0.107	169.75
	Fine	15	7	10	20	30	273825	370939	0.049	4262.21	0.042	1583.98
Cartesian	Cartesian	40	40	40	40	40	32435	31354	0.900	15.61	0.250	20.59
	Medium	20	20	20	20	40	129909	127650	0.443	154.63	0.123	170.68
	Fine	10	10	10	10	10	519747	515170	0.211	1342.80	0.058	1472.98

on accuracy. The impact of LTS on runtimes is greater for the unstructured meshes (factor of 2–3 change) than the Cartesian grid (less than 30% change).

Focusing now on numerical errors, Fig. 15(b) (and Table 7) shows that models converge towards the fine grid prediction irrespective of mesh type or LTS level. Overall, the triangular mesh

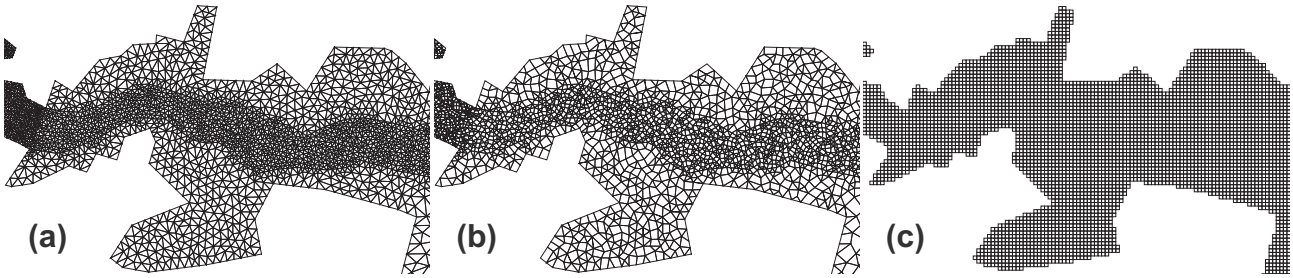

Fig. 13. Coarse mesh versions in zone C and D of Fig. 12(b). (a) Triangular mesh, (b) Mixed mesh and (c) Cartesian grid.

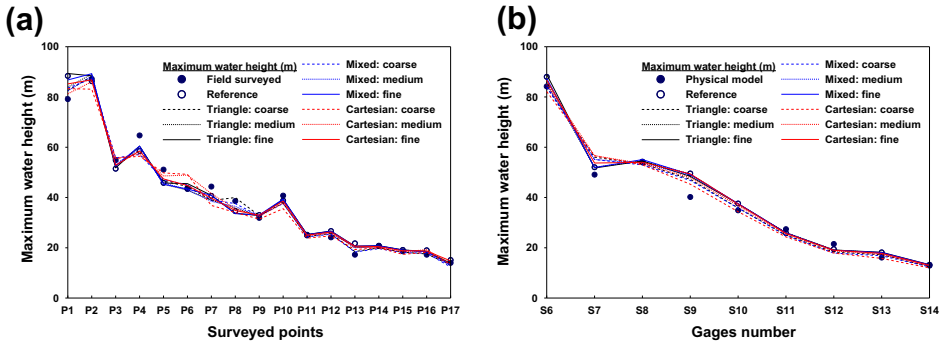

Fig. 14. Comparison of maximum water height predictions and measurements at (a) field surveyed points and (b) physical model gages.

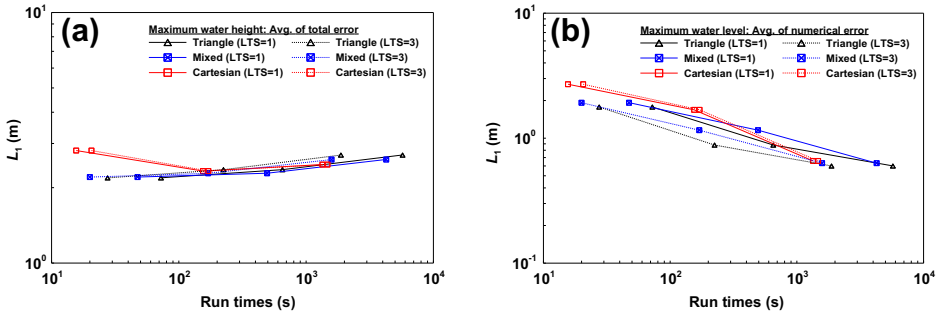

Fig. 15. (a) Total errors and (b) numerical errors of water height predictions versus run times.

Table 7
L_1 norms of maximum water height, arrival time and discharge for Malpasset dam-break flood.

Mesh	Grid	L_1 (m): maximum water height				L_1 (s): arrival time				L_1: (10^3 m^3/s): discharge		
		Field		Physical model		Electric transformer		Physical model		Numerical error		
		Total error	Numerical error	Total error	Numerical error	Total error	Numerical error	Total error	Numerical error	a–a′	b–b′	c–c′
Cartesian	Ultra-fine	2.77	–	2.79	–	66.33	–	49.09	–	–	–	–
Tri.	Coarse	2.00	1.73	2.54	1.85	34.33	76.67	121.09	73.33	2.54	2.53	2.20
	Medium	2.27	1.14	2.52	0.38	96.33	30.00	52.87	11.56	1.02	0.81	0.75
	Fine	2.65	0.72	2.79	0.36	87.00	20.67	45.20	6.11	0.25	0.41	0.42
Mixed	Coarse	2.06	1.89	2.47	1.98	49.67	107.33	136.87	89.11	3.01	3.05	2.70
	Medium	2.09	1.24	2.65	1.01	60.00	29.67	63.53	17.33	1.40	0.83	0.78
	Fine	2.49	0.81	2.78	0.30	104.33	28.00	39.98	9.11	0.76	0.78	0.71
Cartesian	Coarse	2.88	2.60	2.69	2.89	173.00	222.67	223.76	176.00	3.55	5.14	4.21
	Medium	2.13	1.79	2.71	1.49	18.00	61.67	97.42	49.67	1.48	1.81	1.48
	Fine	2.32	0.72	2.76	0.53	64.67	15.67	55.76	7.33	0.99	0.51	0.39

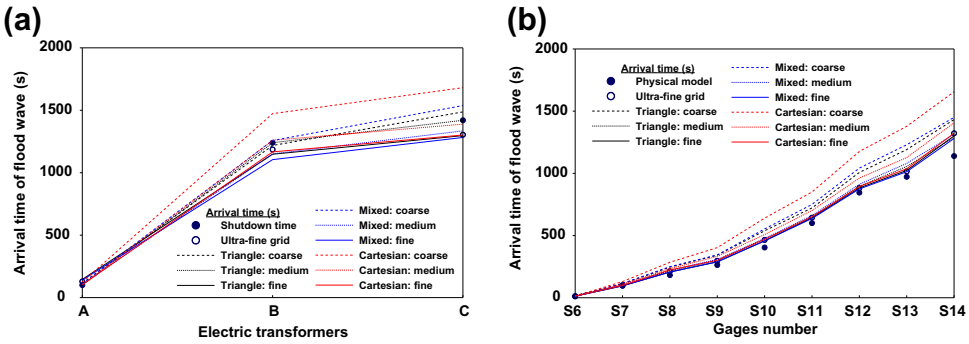

Fig. 16. Comparison of flood arrival time predictions based on (a) shutdown time of electric transformers, (b) gage measurements at physical model.

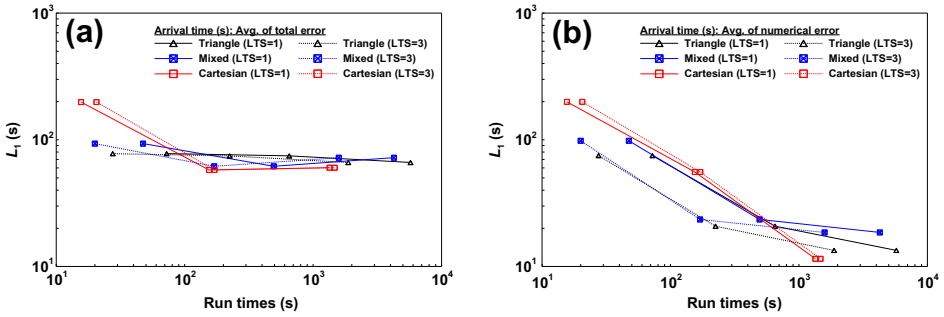

Fig. 17. (a) Total errors and (b) numerical errors of flood arrival time predictions versus run times.

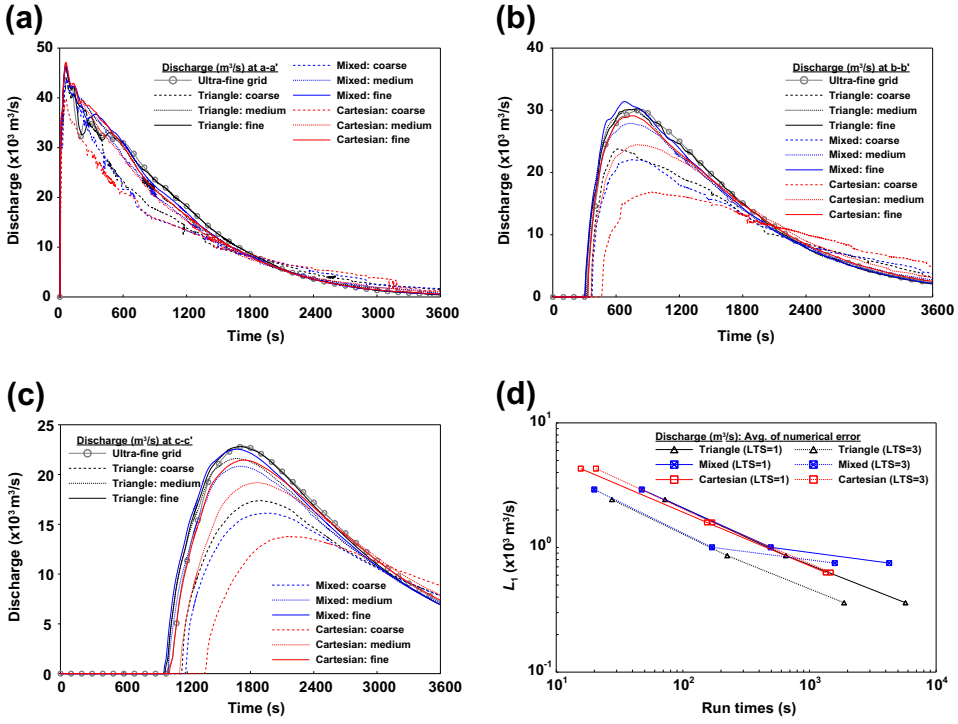

Fig. 18. Flood hydrographs at cross-section (a) a–a′, (b) b–b′, and (c) c–c′ in Fig. 12(b). (d) numerical errors of discharge predictions.

yields the smallest numerical errors for a comparable level of computational effort using LTS = 3. It offers a minimal advantage compared to the mixed mesh, with errors less than 10% smaller, but a notable advantage compared to the Cartesian grid with errors 50% smaller for a similar level of computational effort. On the other hand, using LTS = 1, the advantage of unstructured grids is marginalized. Unstructured grids are only slightly more efficient at coarse resolutions, while Cartesian grids are slightly more efficient at finer resolutions.

Fig. 16 compares flood arrival time predictions against field and laboratory measurement locations. Note, first of all, that the ultra-fine grid mesh predicts a slow-moving flood compared to the field measurements, but a relatively fast moving flood compared to the laboratory measurements. This shows not that laboratory

measurements are not consistent with the field measurements, so measurements of total errors in flood arrival times will offer little insight into the preferred mesh design. Hence, the ultra-fine resolution reference solution is particularly helpful.

Fig. 17 shows total errors and numerical errors in flood arrival time predictions, versus run times. Additionally, all results are summarized in Table 7. These results show that the coarse Cartesian grid (40 m) gives total errors in arrival times that are twice as big as the coarse unstructured grids. Additionally, the numerical error of the 40 m Cartesian grid is about twice that of the coarse unstructured grids showing that Cartesian grid numerical error is significant at this resolution. As the mesh is refined to 20 m, the total errors in Cartesian and unstructured grids are about the same, but the numerical errors of the unstructured grids remain about half the size of the Cartesian grid. At 10 m resolution, the trend does not continue and this may be because the reference solution is only a factor of two finer.

Further insight into the best mesh design is obtained by examining predictions of discharge through the cross-sections marked a–a′, b–b′ and c–c′ in Fig. 12(b). Fig. 18(a)–(c) shows that flood hydrographs are artificially attenuated by the numerical model when an overly coarse mesh is used. The worse example of numerical attenuation is the 40 m resolution (coarse) Cartesian grid, which yields a 50% error in the peak discharge at sections b–b′ and c–c′. Numerical errors shown in Fig. 18(d), plotted versus run times, indicate that the unstructured grids (triangles and mixed meshes) offer the greatest computational efficiency (with LTS = 3) since errors are approximately 50% smaller, compared to Cartesian grids, for the same computational expense.

To summarize, the Malpasset results show that all mesh designs are capable of achieving equally accurate predictions, but that unstructured meshes offer greater computational efficiency than Cartesian grids. This is attributed to the ability of unstructured meshes to locally focus computational resources (local refinement), and the sensitivity of unstructured mesh models to LTS. In fact, using global time stepping (LTS = 1), the efficiency advantage of unstructured grids is lost.

4. Discussion

The preceding results show that highly transient flooding dynamics can be accurately modeled with Godunov-type shallow-water models irrespective of the mesh type, but that mesh design is an important consideration when striving to achieve the smallest numerical errors for the same computational effort. In rectangular channel geometries, quadrilateral mesh types such as Cartesian grids minimize numerical errors more efficiently than triangular grids because these elements are simply a better fit to the geometry, enabling the domain to be discretized with fewer cells and the solution to advanced with a larger time step. Also, in rectangular channel geometries, a uniform resolution mesh works well so unstructured meshing capabilities (localized refinement) are not advantageous. However, in applications involving natural topography and irregular domain boundaries, there is a benefit to localized refinement in terms of reducing both input data errors (better sampling of topography) and reducing numerical errors, and in this context the unstructured mesh designs prove advantageous.

Two other important considerations in flood modeling include model selection, which bears on structural model errors, and the demands of model set up and parameterization. The former includes choices between model types such as 1D, 2D, and 3D approaches in addition to hybrid schemes such as coupled 1D/2D models [5,12,20,28,49,83]. The latter reflects differences in the complexity and time required to set up the model. For example, Cartesian grids are the easiest and unstructured grids are more complicated and can require hours to days for an experienced engineer to prepare with specialized meshing software. These factors are outside the scope of this study, but warrant careful consideration when approaching applications in the field.

5. Conclusions

There is no optimal mesh type for flood modeling with Godunov-type shallow-water models, because each element type (e.g., triangle and quadrilateral) is advantageous under different circumstances. Defining computatioal efficiency as accuracy per computational effort, Cartesian grids are 2–3 times more efficient than triangular grids in rectangular channels test cases, while unstructured grids (triangular grids or mixed meshes of triangular and quadrilateral elements) are about twice as efficient as Cartesian grids in the Malpasset test case where localized refinements prove important for reducing topographic and numerical errors. Additionally, unstructured meshes are required to implement the porosity sub-grid scheme used in this study, and a mixed-mesh model with quadrilateral elements proved 3–4 times more efficient than a triangular grid model. The fact that different mesh types are advantageous under different circumstances points to the utility of a mixed-mesh Godunov-type flood model. Consequently, a mixed-mesh capability is recommended for maximum versatility in practical applications.

LTS is critical for efficient flood modeling with localized refinements, irrespective of mesh type (triangular or mixed meshes). The efficiency advantage cited above is lost using global time stepping (LTS = 1). Additionally, LTS offers only a minor reduction in run times when using Cartesian grids, based on the test cases considered here.

Previous studies suggest topographic and hydrologic errors are generally greater than numerical errors in flood prediction models [6], and results here also indicate that numerical errors may be negligible, compared to structural model errors and input data errors, for the purpose of predicting maximum flood heights. However, the Malpasset dam-break test case showed that hydrograph predictions downstream of the dam are more grid sensitive than maximum flood height predictions. That is, numerical errors can be significant. Additionally, for a given level of accuracy, hydrographs can be computed with 50% less computational effort using an locally refined unstructured grid and LTS.

Acknowledgments

This work was supported by the MRPI program of the University of California Office of the President, United States the Infrastructure Management and Extreme Events program of the National Science Foundation (CMMI-1129730), United States and the National Research Foundation (NRF-2010-357-D00230), South Korea. The authors wish to thank M. Farthing for assistance with meshing.

Appendix A. Numerical scheme

A.1. Topographic and water storage model

Eq. (4) are discretized on an unstructured mesh of N_v vertices, N_c triangular and/or quadrilateral cells, and N_e edges. Elevation data are assigned to mesh vertices z_i, $i = 1, \ldots, N_v$. Elevation varies linearly along each edge of the mesh so topography within each triangular cell is modeled by a plane. For quadrilateral cells, topography is modeled by two planes with the cell bisected along a diagonal [8]. The mesh is formally a second-order accurate model

of topography which is appealing for flood modeling because topographic errors tend to dominate over truncation errors associated with the discrete solution of the shallow-water equations [6,10]. However, compared with commonly used raster grid flood models which assume piecewise constant topography, the treatment of wetting and drying is more complicated because a cell can be partially flooded. That is, the volume of water in a cell might be sufficient to flood the lowest vertex but not the highest vertex. Begnudelli and Sanders [7] introduced the Volume Free-surface Relationship (VFR) method to address this problem in the context of triangular grid models and later extended the approach to quadrilateral grid models [8]. In essence, the VFR is an algebraic equation linking the free surface elevation η_j to the volume of water in cell Ω_j expressed as $h_j = (\mathrm{Vol})_j/(\Omega_j\phi_j)$. Hence, h_j is reserved as a strict indicator of volume and not necessarily cell depth because this is poorly defined when the cell is partially flooded.

The mixed-mesh model presented here merges and simplifies VFRs devised for triangular cells [7] and quadrilateral cells [8]. The key difference is an assumption that h_j varies piecewise linearly with η_j. Begnudelli and Sanders [7,8] show that h_j can vary either as a linear, quadratic, or cubic function of η_j, and higher order polynomials (especially cubic equations) pose a drawback: an expensive, iterative, numerical method may be required to compute η_j given h_j and this step must be completed in every cell every time step. Hence, the appeal of the proposed model stems from the opportunity to use a relatively simple and efficient linear interpolation scheme instead. Assuming a quadrilateral cell with vertex elevations $z_1,\ldots,z_4$ ordered lowest to highest, the piecewise linear VFR is expressed as follows,

$$h(\eta) = \begin{cases} \frac{\eta - z_1}{z_2 - z_1}\hat{h}_2 & \text{if } z_1 \leqslant \eta < z_2 \\ \hat{h}_2 + \frac{\eta - z_2}{z_3 - z_2}(\hat{h}_3 - \hat{h}_2) & \text{if } z_2 \leqslant \eta < z_3 \\ \hat{h}_3 + \frac{\eta - z_3}{z_4 - z_3}(\hat{h}_4 - \hat{h}_3) & \text{if } z_3 \leqslant \eta < z_4 \end{cases} \tag{A-1}$$

where $\hat{h}_2$, $\hat{h}_3$, and $\hat{h}_4$ represent water storage parameters corresponding to water filled to the height of z_2, z_3, and z_4, respectively. The VFR for a triangular cell is just a simplified version of Eq. (A-1) with only three vertex elevations and two water storage parameters, $\hat{h}_2$ and $\hat{h}_3$. Begnudelli and Sanders [7] show that storage in a triangular cell can be computed as a function h_T of vertex elevations $\mathbf{z} = (z_1\, z_2\, z_3)$ (ranked lowest to highest) and η as follows,

$$h_T(\eta, \mathbf{z}) = \begin{cases} 0 & \textbf{if } \eta \leqslant z_1 \\ \frac{(\eta - z_1)^3}{3(z_2 - z_1)(z_3 - z_1)} & \textbf{if } z_1 < \eta \leqslant z_2 \\ \frac{\eta^2 + \eta z_3 - 3\eta z_1 - z_3 z_2 + z_1 z_2 + z_1^2}{3(z_3 - z_1)} & \textbf{if } z_2 < \eta \leqslant z_3 \\ \eta - \frac{z_1 + z_2 + z_3}{3} & \textbf{if } z_3 < \eta \end{cases} \tag{A-2}$$

For triangular cells, calculation of the water storage parameters is straightforward as follows,

$$\hat{h}_i = h_T(z_i, \mathbf{z}) \quad i = 2, 3 \tag{A-3}$$

while for quadrilateral cells, the procedure is slightly more complicated and involves division of the cell into two triangles (a and b), with two sets of vertex elevations $\mathbf{z}_a$ and $\mathbf{z}_b$, and the application of Eq. (A-2) to each triangle to compute the total storage. Begnudelli and Sanders [8] show there are three ways to divide a quadrilateral cell leading to the following expressions for $\mathbf{z}_a$ and $\mathbf{z}_b$,

(1) Lowest (z_1) and highest (z_4) vertices of quadrilateral are diagonally opposite.

$$\mathbf{z}_a = (z_1\, z_2\, z_4) \quad \mathbf{z}_b = (z_1\, z_3\, z_4) \tag{A-4}$$

(2) Lowest (z_1) and highest (z_4) vertices of quadrilateral located on one side, and z_1 and z_2 are on another side.

$$\mathbf{z}_a = (z_1\, z_2\, z_4) \quad \mathbf{z}_b = (z_2\, z_3\, z_4) \tag{A-5}$$

(3) Lowest (z_1) and highest (z_4) vertices of quadrilateral located on one side, and z_1 and z_3 are on another side. In this case, the two highest vertices are diagonally opposite, and the two lowest vertices are diagonally opposite.

$$\mathbf{z}_a = (z_1\, z_3\, z_4) \quad \mathbf{z}_b = (z_2\, z_3\, z_4) \tag{A-6}$$

After dividing the quadrilateral into one of these three cases, water storage parameters are computed as an area-weighted average of the two triangles as follows,

$$\hat{h}_i = \frac{\Omega_a h_T(z_i, \mathbf{z}_a) + \Omega_b h_T(z_i, \mathbf{z}_b)}{\Omega_a + \Omega_b} \quad i = 2, \ldots, 4 \tag{A-7}$$

where Ω_a and Ω_b represent the planform area of triangle a and b, respectively. Areal weighting is required to conserve volume, as depth *per se* is not a conserved property.

Water storage parameters are computed in a pre-processing step of the model, so the work of this method during time-integration is limited to linear interpolation. Importantly, VFRs can be applied in the forward direction (computing h from η) or the reverse direction (computing η from h) by linear interpolation because η, h and z monotonically increase from the lowest to highest vertex.

A.2. Solution update scheme

The solution state is resolved for each cell in the mesh, $\mathbf{U}_j$, $j = 1,\ldots,N_c$, and updated with a first-order accurate finite volume scheme that is optimized for flooding applications. The optimizations include use of a second-order topographic model, as described previously, which enables the model to achieve close to second-order convergence rates in practical applications where topographic errors tend to be larger than truncation errors in the discretized flow equations [10]. A second optimization is a method of variable reconstruction (computing variables at cell edges from cell-centered data) that adapts according to the local Froude number for improved modeling of trans-critical flows [10]. A third optimization is a local time stepping scheme whereby cells are updated with a time step of $\Delta t = \Delta t_o,\ 2\Delta t_o,\ 4\Delta t_o$ or $8\Delta t_o$ depending on which one achieves a Courant number close to one, but not greater than one [64]. Hence, the local time step is given by $m_j \Delta t_o$ where $m_j = 2^{l_j - 1}$ and $l_j \in 1,\ldots,L$ depending on the maximum number of time stepping levels, L. The time step is selected separately for each cell in the mesh, and the net effect is that some cells are updated every time step, some cells are updated every other time step, and some cells are updated every fourth or eighth time step. Mass and momentum conservation is maintained by a careful sequencing of flux calculations and solution updates, and the scheme is no less accurate than a global time stepping scheme [64]. A fourth optimization is the semi-implicit discretization of friction terms, a widely used approach that prevents the time step from ever being constrained to a value smaller than that dictated by the CFL condition [7,10,19,64,65,84].

The solution is updated in time with a fractional step method that accounts first for fluxes $\mathbf{E}$ and $\mathbf{H}$ (step n to n^*) and secondly for source terms $\mathbf{S}$ and $\mathbf{Q}$ (step n^* to $n+1$). The first step appears as follows,

$$\mathbf{U}_j^{n^*} = \mathbf{U}_j^n - \frac{m_j \Delta t_o}{\Omega_j}\left(\sum_{k=1,K_j} w_{j,k} p_{j,k} (\mathbf{E}_\perp)_k^n - \sum_{k=1,K_j} w_{j,k} p_{j,k} (\mathbf{H}_\perp)_{j,k}^n\right) \tag{A-8}$$

where K_j is the number of edges (and vertices) around cell j (either 3 or 4), k is an edge index, $w_{j,k} = \Delta s_k \psi_k / \phi_j$, $\mathbf{E}_\perp = \mathbf{E}\cdot\mathbf{n}$, $\mathbf{H}_\perp = \mathbf{H}\cdot\mathbf{n}$, $\mathbf{n} = (n_x, n_y)$ represents the unit normal vector of the edge, and $p_{j,k}$ equals either 1 or -1 depending on whether $\mathbf{n}$ is directed outward

or inward from the cell center, respectively. Note that the continuity equation is updated in all cells, but the momentum equations are only updated in completed flooded cells defined by a local depth $d > \delta_w$ at all vertices, where δ_w is a small number typically set to 10^{-6} m. The solution update is completed by accounting for friction and point-source terms in a semi-implicit manner as follows [7,10,19,64,65,84],

$$h_j^{n+m_j} = h_j^{n^*} + \frac{m_j \Delta t_o}{\phi_j \Omega_j} \hat{Q}_j^n \tag{A-9}$$

$$(uh)_j^{n+m_j} = \frac{(uh)_j^{n^*}}{1 + m_j \Delta t_o \left[(c_D)_j^n V_j^n / h_j^n - \frac{1}{2}\left(\hat{Q}_j^n - |\hat{Q}_j^n|\right) \Big/ \left(h_j^n \Omega_j\right) \right]} \tag{A-10}$$

$$(vh)_j^{n+m_j} = \frac{(vh)_j^{n^*}}{1 + m_j \Delta t_o \left[(c_D)_j^n V_j^n / h_j^n - \frac{1}{2}\left(\hat{Q}_j^n - |\hat{Q}_j^n|\right) \Big/ \left(h_j^n \Omega_j\right) \right]} \tag{A-11}$$

where $\hat{Q}_j^n$ represents the sum of all point sources in cell j (positive for inflow, negative for outflow) at time level n.

The stability of the update scheme is constrained by the Courant, Friedrichs, Lewy (CFL) condition in the first step (Eq. (A-8)), the friction step is unconditionally stable, and the point source/sink step is unconditionally stable for positive net inflow, $Q_j^{n^*}$, and conditionally stable for negative net inflow. In the case of negative net inflow, the update is stable if the volume removed during the time step is less than the volume stored in the cell. The CFL condition is given for cell j by,

$$\alpha_j = \max\left(w_{j,k}\lambda_k\right)_{k=1,K_j} \frac{m_j \Delta t_o}{\Omega_j} \leqslant 1 \quad \forall j \tag{A-12}$$

where α_j is the Courant number and $\lambda = |un_x + vn_y| + (gh)^{1/2}$ represents the maximum wave speed at each edge of the mesh.

A.3. Variable reconstruction and approximate riemann solver

The flux term $\mathbf{E}_\perp$ is computed at each edge of the mesh with an approximate Riemann solver after reconstructing the solution on both sides. The reconstruction process considers the cell-centered solution in the two neighboring cells and topographic heights at vertices of the mesh edge. Letting z_1 and z_2 represent the edge vertex heights, with $z_1 \leqslant z_2$, and η represent the water height in an adjacent cell, a preliminary edge depth d_e^* on the same side as η is reconstructed as follows [8],

$$d_e^* = \begin{cases} 0 & \text{if } \eta \leqslant z_1 \\ \frac{(\eta - z_1)^2}{2(z_2 - z_1)} & \text{if } z_1 < \eta \leqslant z_2 \\ \eta - \frac{z_1 + z_2}{2} & \text{if } z_2 < \eta \end{cases} \tag{A-13}$$

which ensures that fluxes are computed when the edge is either fully or partially flooded. A preliminary edge-based velocity estimate $\mathbf{u}_e^*$ is computed from the velocity $\mathbf{u} = (u, v)$ and discharge per unit width $\mathbf{uh} = (uh, vh)$ in the neighboring cell center as follows,

$$\mathbf{u}_e^* = \begin{cases} \mathbf{uh}/d_e^* & \text{if } \Delta z < d_e^* \\ \mathbf{u} & \text{if } \delta_w < d_e^* \leqslant \Delta z \\ 0 & \text{if } d_e^* \leqslant \delta_w \end{cases} \tag{A-14}$$

where d_e^* is computed by Eq. (A-13) and Δz represents the difference in height between the highest and lowest vertex of the cell. Hence, the reconstruction depends on whether the cell is fully flooded (Case 1), partially flooded (Case 2), or dry (Case 3). Next, an edge-based Froude number $Fr_e^2 = ||\mathbf{u}^*||^2/(gd_e^*)$ is computed to check for supercritical flow and the reconstructed depth and velocity are finalized as follows,

$$\begin{aligned} d_e &= d_e^* \quad \& \quad \mathbf{u}_e = \mathbf{u}_e^* \quad \text{if} \quad Fr_e^2 < 1 \\ d_e &= h \quad \& \quad \mathbf{u}_e = \mathbf{u} \quad \text{if} \quad Fr_e^2 \geqslant 1 \end{aligned} \tag{A-15}$$

Bradford and Sanders [14] showed that when dealing with irregular topographic data typical of flood prediction applications, a smooth reconstruction of the depth and velocity at cell edges minimizes numerical errors that cause unwanted energy dissipation. In essence, the above reconstruction assumes that **uh** and η are piecewise constant under subcritical flow conditions, and that **u** and d are piecewise constant under supercritical flow. This minimizes differences in reconstructed variables across edges [10].

Note that Eqs. (A-13)–(A-15) are applied twice to reconstruct the velocity and depth on both sides of each edge, and thus define a Riemann problem that is solved to evaluate fluxes. If $d_e \leqslant \delta_w$ on both sides of the edge then $\mathbf{E}_\perp = 0$. Otherwise, a variant of Roe's Approximate Riemann solver [62] is applied which gives the flux as follows,

$$\mathbf{E}_\perp = \frac{1}{2}\left(\mathbf{E}_\perp^L + \mathbf{E}_\perp^R - \hat{\mathbf{R}}|\hat{\mathbf{\Lambda}}|\Delta\hat{\mathbf{V}}\right) \tag{A-16}$$

where $\mathbf{E}_\perp^L$ and $\mathbf{E}_\perp^R$ denote the fluxes defined by the reconstructed data (d, u and v) on the left (superscript L) and right (superscript R) side of each face, $\mathbf{\Lambda}$ and $\mathbf{R}$ denote the matrix of eigenvalues and right eigenvectors, respectively, of the Jacobian matrix defined by $d\mathbf{F}/d\mathbf{U}$, and $\Delta\mathbf{V} = \mathbf{R}^{-1}\Delta\mathbf{U}$ represents wave strengths (differences in the solution) across the edge. The hat notation in Eq. (A-16) indicates that these matrices should be evaluated with so-called Roe-average variable values computed as follows [75],

$$\hat{d} = \sqrt{d_L d_R}, \quad \hat{c} = \sqrt{\frac{1}{2} g (d_L + d_R)} \tag{A-17}$$

$$\hat{u} = \frac{\sqrt{d_L}u_L + \sqrt{d_R}u_R}{\sqrt{d_L} + \sqrt{d_R}}, \quad \hat{v} = \frac{\sqrt{d_L}v_L + \sqrt{d_R}v_R}{\sqrt{d_L} + \sqrt{d_R}} \tag{A-18}$$

where $c = (gd)^{1/2}$ represents the speed of a simple gravity wave. Written in terms of Roe-averages, $|\hat{\mathbf{\Lambda}}|$, $\hat{\mathbf{R}}$ and $\Delta\hat{\mathbf{V}}$ are expressed as follows,

$$|\hat{\mathbf{\Lambda}}| = \begin{pmatrix} |\hat{u}_\perp - \hat{c}|^* & 0 & 0 \\ 0 & |\hat{u}_\perp| & 0 \\ 0 & 0 & |\hat{u}_\perp + \hat{c}|^* \end{pmatrix} \tag{A-19}$$

$$\hat{\mathbf{R}} = \begin{pmatrix} 1 & 0 & 1 \\ \hat{u} - \hat{c}n_x & -n_y & \hat{u} + \hat{c}n_x \\ \hat{v} - \hat{c}n_y & n_x & \hat{v} + \hat{c}n_y \end{pmatrix} \tag{A-20}$$

$$\Delta\hat{\mathbf{V}} = \begin{pmatrix} \frac{1}{2}\left(\Delta d - \frac{\hat{h}\Delta u_\perp}{\hat{c}}\right) \\ \hat{h}\Delta u_{||} \\ \frac{1}{2}\left(\Delta d + \frac{\hat{h}\Delta u_\perp}{\hat{c}}\right) \end{pmatrix} \tag{A-21}$$

where $u_\perp = un_x + vn_y$, $u_{||} = -un_y + vn_x$ is the velocity parallel to the cell edge, and Δ denotes the finite difference normal to the edge, for example $\Delta d = d_R - d_L$.

Roe's method will incorrectly predict a hydraulic jump at the critical flow point within a rarefaction wave, so a so-called *entropy fix* is required [75]. A fix proposed by Van Leer et al. [80] and later adopted by Bradford and Sanders [13] is used here. This fix is reflected by the asterisks in the matrix of eigenvalues (Eq. (A-19)), which indicate that the value of the eigenvalues may be adjusted. First consider the possibility of a left-moving rarefaction whereby $u_\perp = c$. To implement the fix, a wave speed tolerance is computed as follows,

$$\Delta\lambda_1 = 2\max\left[(\lambda_1)_R - (\lambda_1)_L, 0\right] \tag{A-22}$$

where $\lambda_1 = u_\perp - c$ and the adjusted eigenvalue is given by,

$$|\hat{\lambda}_1|^* = \begin{cases} \frac{1}{2}\left(\hat{\lambda}_1^2/\Delta\lambda_1 + \Delta\lambda_1\right) & \text{if } |\hat{\lambda}_1| < \Delta\lambda_1 \\ |\hat{\lambda}_1| & \text{if } |\hat{\lambda}_1| \geqslant \Delta\lambda_1 \end{cases} \tag{A-23}$$

The case of a right-moving rarefaction corresponds to $-u_\perp = c$ and in this case the wave speed tolerance is computed as,

$$\Delta\lambda_3 = 2\ \max\left[(\lambda_3)_R - (\lambda_3)_L, 0\right] \tag{A-24}$$

where $\lambda_3 = u_\perp + c$. The adjusted eigenvalue $|\hat{\lambda}_3|^*$ is computed using Eq. (A-23) with λ_3 in place of λ_1.

A.4. Ground slope flux terms

The flux term $\mathbf{H}_\perp$ accounts for ground slope effects and is discontinuous across the mesh edge, in contrast with $\mathbf{E}_\perp$ which is continuous across the edge. The flux is evaluated by first reconstructing the edge-based depth d_e at each of the K_j edges of cell Ω_j using Eq. (A-13) and the cell water height η_j. The source term flux of edge $k = 1, \ldots, K_j$ of cell Ω_j follows as,

$$(\mathbf{H}_\perp)_k = \begin{pmatrix} 0 \\ \frac{1}{2}g(d_e)^2 n_x \\ \frac{1}{2}g(d_e)^2 n_y \end{pmatrix}_k \tag{A-25}$$

and for consistency with the evalation of $\mathbf{E}_\perp$, $\mathbf{H}_\perp = 0$ if $d_e \leqslant \delta_w$.

A.5. Solution update procedure and data management

Preparation of a computational mesh is the first major task to model flooding at a site. This can be supported by meshing tools such as Triangle [68], the Surface Water Modeling System (SMS) (Aquaveo, Provo, Utah), or one of many other options. The meshing tool will generate a distribution of vertex locations (x_m, y_m), the connectivity of triangles and/or quadrilateral cells, and possibly arrays of the neighborhood of cells surrounding each cell. Mesh preparation continues with the assignment of topographic heights to each vertex z_m (from topographic data), a resistance parameter to each cell (e.g., Manning n_m) in accordance with the land cover, appropriate flags to account for boundary conditions and point sources, and if porosity parameters are to be used, ϕ_j to each cell and ψ_k to each edge in accordance with the shape and location of sub-grid obstructions.

At run-time, the mesh files and other relevant input are loaded into the model and a number pre-processing steps are completed such as the creation of pointers to manage the unstructured data format [7]; calculation of cell areas Ω_j, edge lengths Δs_k, and edge angles $\mathbf{n}_k$; calculation of VFR parameters; and the specification of an initial condition including the following cell-based data: h_j, η_j, uh_j, vh_j, u_j and v_j which are saved in cell-based arrays of dimension $N_c \times 1$. Once these steps are completed, time integration proceeds with the following steps,

(1) Fluxes are computed with a global sweep over all edges, $(\mathbf{E}_\perp)_k$, $k = 1, \ldots, N_v$, including boundary edges where boundary conditions are enforced. Data are saved in an edge-based array of dimension $N_v \times 3$.
(2) Ground slope fluxes are computed with a global sweep over all cells, and a nested sweep over the K_j local edges to compute the sum of the fluxes as follows,

$$\Sigma_j = \sum_{k=1,K_j} w_{j,k} p_{j,k} (\mathbf{H}_\perp)_k \tag{A-26}$$

The sum for each cell (Σ_j) is saved in a cell-based array of dimension $N_c \times 2$.

(3) Values of h_j, uh_j and vh_j are advanced to the next time level by applying Eq. (A-8) (fluxes) and Eqs. (A-9)–(A-11) (friction and source terms) with a sweep over all cells.
(4) The model checks for the possibility of a negative h_j (overdraft) with a sweep over all cells. In such cases, water is redistributed among cells sharing an edge with Ω_j to ensure positivity [7,8].
(5) The model updates η_j according to the VFR (Eq. (A-1)) and u_j and v_j are computed in fully flooded cells. Otherwise, the velocity is set to zero ($u_j = v_j = 0$).

As mentioned previously, a local time stepping (LTS) scheme is used which effectively advances a cell $m_j = 2^{l_j - 1}$ base time steps with each update, where l_j is an integer in the range $1 \leqslant l_j \leqslant L$. The LTS scheme executes a coordinated cycle of flux, source term, and solution update sweeps that repeats every $M = 2^{L-1}$ time steps to advance the solution. Some cells are updated every time step (level 1), some cells are updated every other time step (level 2), and so on. These sweeps are controlled by the level l_j so a set of integer arrays are introduced to designate the LTS level of each cell and edge of the 2D mesh. Additionally, logical checks are added to every routine in the simulation algorithm to control operations according to the LTS level.

The assignment of LTS levels to cells and edges occurs every M time steps and follows a three-step process described below. The process is controlled by a target Courant number, α_T, which is slightly less than unity (typically 0.8) to satisfy the CFL condition:

(1) The cell Courant number corresponding to the base time step Δt_o is computed as follows,

$$\alpha_j^o = \max\left(w_{j,k}\lambda_k\right)_{k=1,K_j} \frac{\Delta t_o}{\Omega_j} \tag{A-27}$$

and a preliminary cell-based LTS level $(l_c)_j$ is given by the l from the range $1, \ldots, L$ which satisfies the following inequality,

$$\frac{\alpha_T}{2^l} \leqslant \alpha_j^o < \frac{\alpha_T}{2^{l-1}} \tag{A-28}$$

with the exception of level 1 which is controlled by $\alpha_j^o \geqslant \alpha_o/2$ and level L which is controlled by $\alpha_j^o < \alpha_T/2^{L-1}$. To clarify, consider two examples based on $\alpha_T = 0.8$. If $\alpha_j^o = 0.25$, then $l_j = 2$ and $\Delta t_j = 2^{2-1}\Delta t_o = 2\Delta t_o$. If $\alpha_j^o = 0.08$, then $l_j = 4$ and $\Delta t_j = 2^{4-1}\Delta t_o = 8\Delta t_o$.
(2) The edge-based LTS level, $(l_e)_k$, is assigned as the minimum of the two neighboring cell-based LTS levels.
(3) The cell-based LTS level $(l_c)_j$ is finalized as the minimum of the K_j neighboring face-based LTS levels. This ensures that neighboring cells differ by at most one LTS level.

The cycle of M time steps proceeds with sweeps over cells and edges as described above (Steps 1–5 of solution update procedure). During each sweep, data are only operated on when the LTS level is less than or equal to a variable threshold l_o that repeats a particular sequence every M time steps. The sequence of l_o for $L = 3$ is given by 1, 2, 1 and 3. This corresponds to the following cycle of updates:

(1) Compute fluxes and source terms for all edges and cells, respectively.
(2) Advance level 1 cells from t to $t + \Delta t_o$.
(3) Update fluxes and source terms in level 1 edges/cells.
(4) Advance level 1 cells from $t + \Delta t_o$ to $t + 2\Delta t_o$ and advance level 2 cells from t to $t + 2\Delta t_o$.
(5) Update fluxes and source terms in level 1 and level 2 edges/cells.
(6) Advance level 1 cells from $t + 2\Delta t_o$ to $t + 3\Delta t_o$.

(7) Update fluxes and source terms in level 1 edges/cells.
(8) Advance level 1 cells from $t + 3\Delta t_o$ to $t + 4\Delta t_o$, advance level 2 cells from $t + 2\Delta t_o$ to $t + 4\Delta t_o$, and advance level 3 cells from t to $t + 4\Delta t_o$

The sequence for $L = 4$ is $l_o = 1, 2, 1, 3, 1, 2, 1, 4$; and for $L = 5$ the sequence is $l_o = 1, 2, 1, 3, 1, 2, 1, 4, 1, 2, 1, 3, 1, 2, 1, 5$. Note that the threshold for fluxes/source terms lags the threshold for solution updates. In addition, note that it is convenient to calculate the Courant number with (Eq. (A-27)) immediately following the first step of this process. Previous work has indicated that $L = 3$ or 4 is optimal for applications with wetting and drying [64].

References

[1] Abderrezzak KE, Paquier A, Mignot E. Modelling flash flood propagation in urban areas using a two-dimensional numerical model. Nat Hazards 2009;50:433–60. http://dx.doi.org/10.1007/s11069-008-9300-0.
[2] Aricó C, Sinagra M, Begnudelli L, Tucciarelli T. MAST-2D diffusive model for flood prediction on domains with triangular Delaunay unstructured meshes. Adv. Water Resour. 2011;34:1427–49. http://dx.doi.org/10.1016/j.advwatres.2011.08.002.
[3] Alcrudo F, Garcia-Navarro P. A high resolution Godunov-type scheme in finite volumes for the 2D shallow-water equations. Int J Numer Methods Fluids 1993;16(6):489–505. http://dx.doi.org/10.1002/fld.1650160604.
[4] Aizinger V, Dawson C. A discontinuous Galerkin method for two-dimensional flow and transport in shallow water. Adv. Water Resour. 2002;25(1):67–84. http://dx.doi.org/10.1016/S0309-1708(01)00019-7.
[5] Bates PD, Wilson MD, Horritt MS, Mason D, Holden N, Currie A. Reach scale floodplain inundation dynamics observed using airborne synthetic aperture radar imagery: data analysis and modeling. J Hydrol 2006;328:306–18. http://dx.doi.org/10.1016/j.jhydrol.2005.12.028.
[6] Bates PD. Integrating remote sensing data with flood inundation models: how far have we got? Hydrol Process 2012;26:2515–21. http://dx.doi.org/10.1002/hyp.9374.
[7] Begnudelli L, Sanders BF. Unstructured grid finite-volume algorithm for shallow-water flow and scalar transport with wetting and drying. J Hydraul Eng 2006;132(4):371–84. http://dx.doi.org/10.1061/(ASCE)0733-9429(2006)132:4(371).
[8] Begnudelli L, Sanders BF. Conservative wetting and drying methodology for quadrilateral grid finite-volume models. J Hydraul Eng 2007;133(3):312–22. http://dx.doi.org/10.1061/(ASCE)0733-9429(2007)133:3(312).
[9] Begnudelli L, Sanders BF. Simulation of the St. Francis dam-break flood. J Eng Mech 2007;133(11):1200–12. http://dx.doi.org/10.1061/(ASCE)0733-9399(2007)133:11(1200).
[10] Begnudelli L, Sanders BF, Bradford SF. An adaptive Godunov-based model for flood simulation. J Hydraul Eng 2008;134(6):714–25. http://dx.doi.org/10.1061/(ASCE)0733-9429(2008)134:6(714).
[11] Bilskie MV, Hagen SC. Topographic accuracy assessment of bare earth lidar-derived unstructured meshes. Adv Water Resour 2013;52:165–77. http://dx.doi.org/10.1016/j.advwatres.2012.09.003.
[12] Bladé E, Gómez-Valentín M, Dolz J, Aragón-Hernández JL, Corestein G, Sánchez-Juny M. Integration of 1D and 2D finite volume schemes for computations of water flow in natural channels. Adv Water Resour 2012;42:17–29. http://dx.doi.org/10.1016/j.advwatres.2012.03.021.
[13] Bradford SF, Sanders BF. Finite-volume model for shallow-water flooding of arbitrary topography. J Hydraul Eng 2002;128(3):289–98. http://dx.doi.org/10.1061/(ASCE)0733-9429(2002)128:3(289).
[14] Bradford SF, Sanders BF. Performance of high-resolution, non-level bed, shallow-water models. J Eng Mech 2005;131(10):1073–81. http://dx.doi.org/10.1061/(ASCE)0733-9399(2005)131:10(1073).
[15] Brufau P, García-Navarro P. Two-dimensional dam break flow simulation. Int J Numer Methods Fluids 2000;33:35–57. http://dx.doi.org/10.1002/(SICI)1097-0363(20000515)33:1<35::AID-FLD999>3.0.CO;2-D.
[16] Brufau P, García-Navarro P, Vázquez-Cendón. Zero mass error using unsteady wetting-drying conditions in shallow flows over dry irregular topgoraphy. Int J Numer Methods Fluids 2004;45:1047–82. http://dx.doi.org/10.1002/fld.729.
[17] Causon DM, Ingram DM, Mingham CG, Yang G, Pearson RV. Calculation of shallow water flows using a Cartesian cut cell approach. Adv Water Resour 2000;23(5):545–62. http://dx.doi.org/10.1016/S0309-1708(99)00036-6.
[18] Causon DM, Ingram DM, Mingham CG. A Cartesian cut cell method for shallow water flows with moving boundaries. Adv Water Resour 2001;24(8):899–911. http://dx.doi.org/10.1016/S0309-1708(01)00010-0.
[19] Cea L, Vázquez-Cendón ME. Unstructured finite volume discretization of two-dimensional depth-averaged shallow water equations with porosity. Int J Numer Methods Fluids 2010;63:903–30. http://dx.doi.org/10.1002/fld.210.
[20] Chen Y, Wang Z, Liu Z, Zhu D. 1D–2D coupled numerical model for shallow-water flows. J Hydraul Eng 2012;138(2):122–32. http://dx.doi.org/10.1061/(ASCE)HY.1943-7900.0000481.
[21] Cobby DM, Mason DC, Horritt MS, Bates PD. Two-dimensional hydraulic flood modelling using a finite-element mesh decomposed according to vegetation and topographic features derived from airborne scanning laser altimetry. Hydrol Process 2003;17:1979–2000. http://dx.doi.org/10.1002/hyp.1201.
[22] de Almeida GAM, Bates P, Freer JE, Souvignet M. Improving the stability of a simple formulation of the shallow water equations for 2-D flood modeling. Water Resour Res 2012;48:W05528. http://dx.doi.org/10.1029/2011WR011570.
[23] de Almeida GAM, Bates PD. Applicability of the local inertial approximation of the shallow water equations to flood modeling. Water Resour Res 2013;49(8):4833–44. http://dx.doi.org/10.1002/wrcr.20366.
[24] Defina A. Two-dimensional shallow flow equations for partially dry areas. Water Resour Res 2000;36:3251–64. http://dx.doi.org/10.1029/2000WR900167.
[25] Delis AI, Kazolea M, Kampanis NA. A robust high-resolution finite volume scheme for the simulation of long waves over complex domains. Int J Numer Methods Fluids 2008;56:419–52. http://dx.doi.org/10.1002/fld.1537.
[26] Ern A, Piperno S, Djadel K. A well-balanced Runga–Kutta discontinuous Galerkin method for shallow-water equations with flooding and drying. Int J Numer Methods Fluids 2008;58(1):1–25. http://dx.doi.org/10.1002/fld.1674.
[27] Ernst J, Dewals BJ, Detrembleur S, Archambeau P, Erpicum S, Pirotton M. Micro-scale flood risk analysis based on detailed 2D hydraulic modelling and high resolution geographic data. Nat Hazards 2010;55:181–209. http://dx.doi.org/10.1007/s11069-010-9520-y.
[28] Fernández-Nieto ED, Marin J, Monnier J. Coupling superposed 1D and 2D shallow-water models: source terms in finite volume schemes. Comput Fluids 2010;39(6):1070–82. http://dx.doi.org/10.1016/j.compfluid.2010.01.016.
[29] Fraccarollo L, Toro EF. Experimental and numerical assessment of the shallow water model for two-dimensional dam-break type problems. J Hydraul Res 1995;33(6):843–64. http://dx.doi.org/10.1080/00221689509498555.
[30] Gallegos HA, Schubert JE, Sanders BF. Two-dimensional, high-resolution modeling of urban dam-break flooding: a case study of Baldwin Hills. California Adv Water Resour 2009;32:1323–35. http://dx.doi.org/10.1016/j.advwatres.2009.05.008.
[31] Gallegos HA, Schubert JE, Sanders BF. Structural damage prediction in a high-velocity urban dam-break flood: field-scale assessment of predictive skill. J Eng Mech 2012;138(10):1249–62. http://dx.doi.org/10.1061/(ASCE)EM.1943-7889.0000427.
[32] Gallien TW, Schubert JE, Sanders BF. Predicting tidal flooding of urbanized embayments: a modeling framework and data requirements. Coastal Eng 2011;58:567–77. http://dx.doi.org/10.1016/j.coastaleng.2011.01.011.
[33] Gallien TW, Sanders BF, Flick RE. Modeling urban coastal flooding from wave-driven beach overtopping: field validation and uncertainty analysis. Coastal Eng, submitted for publication.
[34] George DL. Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: application to the Malpasset dam-break flood (France, 1959). Int J Numer Methods Fluids 2010;66(8):1000–18. http://dx.doi.org/10.1002/fld.2298.
[35] Glaister P. Approximate Reimann solutions of the 2-dimensional shallow-water equations. J Eng Math 1990;24(1):45–53. http://dx.doi.org/10.1080/00221688809499213.
[36] Goutal N. The Malpasset dam failure – an overview and test case definition. In: The proceeding of the fourth CADAM meeting, Zaragoza, Spain; Nov 18–19, 1999.
[37] Guinot V. Godunov-type schemes (An introduction for engineers). Science. Elsevier; 2003. 508p.
[38] Guinot V, Soares-Fraz S. Flux and source term discretization in shallow water models with porosity on unstructured grids. Int J Numer Methods Fluids 2006;50:309–45. http://dx.doi.org/10.1002/fld.1059.
[39] Guinot V. Multiple porosity shallow water models for macroscopic modelling of urban floods. Adv Water Resour 2012;37:40–72. http://dx.doi.org/10.1016/j.advwatres.2011.11.002.
[40] Hervouet JM, Petitjean A. Malpasset dam-break revisited with two-dimensional computations. J Hydraul Res 1999;37(6):777–88. http://dx.doi.org/10.1080/00221689909498511.
[41] Hubbard ME. Multidimensional slope limiters for MUSCL-type finite volume schemes on unstructured grids. J Comput Phys 1999;155:54–74. http://dx.doi.org/10.1006/jcph.1999.6329.
[42] Hubbard ME, Garcia-Navarro P. Flux difference splitting and the balancing of source terms and flux gradients. J Comput Phys 2000;165(1):89–125. http://dx.doi.org/10.1006/jcph.2000.6603.
[43] Kelman I, Spence R. A flood failure flowchart for buildings. In: Proceedings of the institution of civil engineers-municipal, Engineer, 156(3); 2003. p. 207–14. http://dx.doi.org/10.1680/muen.2003.156.3.207..
[44] Kelman I, Spence R. An overview of flood actions on buildings. Eng Geol 2004;73(3–4):297–309. http://dx.doi.org/10.1016/j.enggeo.2004.01.010.
[45] Kesserwani G, Liang QH. A discontinuous Galerkin algorithm for the two-dimensional shallow water equations. Comput Methods Appl Mech Eng 2010;199(49–52):3356–68. http://dx.doi.org/10.1016/j.cma.2010.07.007.
[46] Kesserwani G, Liang QH. Dynamically adaptive grid based discontinuous Galerkin shallow water model. Adv Water Resour 2012;37:23–39. http://dx.doi.org/10.1016/j.advwatres.2011.11.006.
[47] Kim DH, Lynett PJ, Socolofsky SA. A depth-integrated model for weakly dispersive, turbulent, and rotational fluid flows. Ocean Modell 2009;27(3–4):198–214. http://dx.doi.org/10.1016/j.ocemod.2009.01.005.
[48] Kreibich H, Piroth K, Seifert I, Maiwald H, Kunert U, Schwarz J, et al. Is flow velocity a significant parameter in flood damage modelling? Nat Hazards Earth Syst Sci 2009;9:1679–92. http://dx.doi.org/10.5194/nhess-9-1679-2009.

[49] Kuiry SN, Sen D, Bates PD. Coupled 1D-quasi-2D flood inundation model with unstructured grids. J Hydraul Eng 2010;136(8):493–506. http://dx.doi.org/10.1061/(ASCE)HY.1943-7900.0000211.
[50] Li S, Duffy CJ. Fully coupled approach to modeling shallow water flow, sediment transport, and bed evolution in rivers. Water Resour Res 2011;47:W03508. http://dx.doi.org/10.1029/2010WR009751.
[51] Liang D, Lin B, Falconer RA. A boundary-fitted numerical model for flood routing with shock-capturing capability. J Hydrol 2007;332(3–4):477–86. http://dx.doi.org/10.1016/j.jhydrol.2006.08.002.
[52] Liang Q, Borthwick AGL. Adaptive quadtree simulation of shallow flows with wet-dry fronts over complex topography. Comput Fluids 2008;38(2):221–34. http://dx.doi.org/10.1016/j.compfluid.2008.02.008.
[53] Liao CB, Wu MS, Liang SJ. Numerical simulation of a dam break for an actual river terrain environment. Hydrol Process 2007;21(4):447–60. http://dx.doi.org/10.1002/hyp.6242.
[54] Mason DC, Horritt MS, Hunter NM, Bates PD. Use of fused airborne scanning laser altimetry and digital map data for urban flood modelling. Hydrol Process 2007;2007(21):1436–47. http://dx.doi.org/10.1002/hyp.6343.
[55] McMillan HK, Brasington J. Reduced complexity strategies for modelling urban floodplain inundation. Geomorphology 2007;90:226–43. http://dx.doi.org/10.1016/j.geomorph.2006.10.031.
[56] Middelmann-Fernandez MH. Flood damage estimation beyond stage-damage functions: an Australian example. J Flood Risk Manage 2010;3(1):88–96. http://dx.doi.org/10.1111/j.1753-318X.2009.01058.x.
[57] Mignot E, Paquier A, Haider S. Modeling floods in a dense urban area using 2D shallow water equations. J Hydrol 2006;327(1–2):186–99. http://dx.doi.org/10.1016/j.jhydrol.2005.11.026.
[58] Morris MW. CADAM: concerted action on dambreak modeling. Final report. Rep. No. SR 571. Wallingford: HR Wallingford; 2000.
[59] Munson BR, Young DF, Okiishi TH. Fundamentals of fluid mechanics. 5th ed. John Wiley & Sons; 2006. 769p.
[60] Neal J, Schumann G, Bates P. A sub grid channel model for simulating river hydraulics and floodplain inundation over large and data sparse areas. Water Resour Res 2012;48:W11506. http://dx.doi.org/10.1029/2012WR012514.
[61] Nepf HM. Drag, turbulence and diffusion in flow through emergent vegetation. Water Resour Res 1999;35(2):479–89. http://dx.doi.org/10.1029/1998WR900069.
[62] Roe PL. Approximate Riemann solvers, parameter vectors, and difference schemes. J Comput Phys 1981;43:357–72. http://dx.doi.org/10.1016/0021-9991(81)90128-5.
[63] Sanders BF. Non-reflecting boundary flux function for finite volume shallow-water models. Adv Water Resour 2002;25:195–202. http://dx.doi.org/10.1016/S0309-1708(01)00055-0.
[64] Sanders BF. Integration of a shallow-water model with a local time step. J Hydraul Res 2008;46(8):466–75. http://dx.doi.org/10.3826/jhr.2008.3243.
[65] Sanders BF, Schubert JE, Gallegos HA. Integral formulation of shallow-water equations with anisotropic porosity for urban flood modeling. J Hydrol 2008;362:19–38. http://dx.doi.org/10.1016/j.jhydrol.2008.08.009.
[66] Schubert JE, Sanders BF, Smith MJ, Wright NG. Unstructured mesh generation and landcover-based resistance for hydrodynamic modeling of urban flooding. Adv Water Resour 2008;31:1603–21. http://dx.doi.org/10.1016/j.advwatres.2008.07.012.
[67] Schubert JE, Sanders BF. Building treatments for urban flood inundation models and implications for predictive skill and modeling efficiency. Adv Water Resour 2012;41:49–64. http://dx.doi.org/10.1016/j.advwatres.2012.02.012.
[68] Shewchuk JR. Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In: Lin MC, Manocha D, editors. Applied computational geometry: towards geometric engineering. Lecture notes in computer science, vol. 1148. Springer-Verlag; 1996. p. 20322. Available from: <http://www-2.cs.cmu. edu/_quake/triangle.html>.
[69] Singh J, Altinakar MS, Ding Y. Two-dimensional numerical modeling of dam-break flows over natural terrain using a central explicit scheme. Adv Water Resour 2011;34(10):1366–75. http://dx.doi.org/10.1016/j.advwatres.2011.07.007.
[70] Sleigh PA, Gaskell PH, Berzins M, Wright NG. An unstructured finite-volume algorithm for predicting flow in rivers and estuaries. Comput Fluids 1998;27(4):479–508. http://dx.doi.org/10.1016/S0045-7930(97)00071-6.
[71] Smith DI. Flood damage estimation – a review of urban stage-damage curves and loss functions. Water SA 1994;20(3):231–8.
[72] Soares-Frazão S, Lhomme J, Guinot V, Zech Y. Two-dimensional shallow-water model with porosity for urban flood modelling. J Hydraul Res 2008;46(1):45–64. http://dx.doi.org/10.1080/00221686.2008.9521842.
[73] Testa G, Zuccala D, Alcrudo F, Mulet J, Soares-Frazão S. Flash flood flow experiment in a simplified urban district. J Hydraul Res 2007;45(Extra Issue):37–44. http://dx.doi.org/10.1080/00221686.2007.9521831.
[74] Toulorge T, Desmet W. CFL conditions for Runge–Kutta discontinuous Galerkin methods on triangular grids. J Comput Phys 2011;230(12):4657–78. http://dx.doi.org/10.1016/j.jcp.2011.02.040.
[75] Toro EF. Shock-capturing methods for free-surface shallow flows. Chichester, UK: John Wiley & Sons; 2001. 309p.
[76] Tsubaki R, Fujita I. Unstructured grid generation using LiDAR data for urban flood inundation modelling. Hydrol Process 2010;24:1404–20. http://dx.doi.org/10.1002/hyp.7608.
[77] Valiani A, Caleffi V, Zanni A. Case study: Malpasset dam-break simulation using a two-dimensional finite volume method. J Hydraul Eng 2002;128(5):460–72. http://dx.doi.org/10.1061/(ASCE)0733-9429(2002)128:5(460).
[78] Valiani A, Begnudelli L. Divergence form for bed slope source term in shallow water equations. J Hydraul Eng 2006;132(7):652–65. http://dx.doi.org/10.1061/(ASCE)0733-9429(2006)132:7(652).
[79] Valiani A, Begnudelli L. Closure to Divergence form for bed slope source term in shallow water equations by Alessandro Valiani and Lorenzo Begnudelli. J. Hydraul Eng 2008;134(5):680–2. http://dx.doi.org/10.1061/(ASCE)0733-9429(2008)134:5(680).
[80] Van Leer B, Lee WT, Powell KG. Sonic point capturing. In: Ninth computational fluid dynamics conference. Buffalo (NY): American Institute of Aeronautics and Astronautics; 1989.
[81] Wang Y, Liang Q, Kesserwani G, Hall JW. A 2D shallow-water model for practical dam-break simulations. J Hydraul Res 2011;49(3):307–16. http://dx.doi.org/10.1080/00221686.2011.566248.
[82] World meteorological organization. Urban flood risk management – a tool for integrated flood management Version 1.0. APFM Techical Document No. 11, Flood Management Tools Series; 2008.
[83] Wright NG, Villanueva I, Bates PD, Mason DC, Wilson MD, Pender G, et al. Case study of the use of remotely sensed data for modeling flood inundation on the river Severn, UK. J Hydraul Eng 2008;134(5):533–40. http://dx.doi.org/10.1061/(ASCE)0733-9429(2008)134:5(533).
[84] Yoon TH, Kang SK. Finite volume model for two-dimensional shallow water flows on unstructured grids. J Hydraul Eng 2004;130(7):678–88. http://dx.doi.org/10.1061/(ASCE)0733-9429(2004)130:7(678).
[85] Yu D, Lane SN. Urban fluvial flood modelling using a two dimensional diffusion-wave treatment. Part 2: Development of a sub-grid-scale treatment. J Hydrol Process 2005;20(7):1567–83. http://dx.doi.org/10.1002/hyp.5936.
[86] Zhou JG, Causon DM, Mingham CG, Ingram DM. The surface gradient method for the treatment of source terms in the shallow-water equations. J Comput Phys 2001;168(1):1–25. http://dx.doi.org/10.1006/jcph.2000.6670.