(Multi)wavelets increase both accuracy and efficiency of standard Godunov-type hydrodynamic models: robust 2D approaches

Georges Kesserwani[a], Mohammad Kazem Sharifian[a,*]

[a]Department of Civil and Structural Engineering, University of Sheffield, Mappin St, Sheffield City Centre, Sheffield S1 3JD, UK

Abstract

Multiwavelets (MW) enable the compression, analysis and assembly of model data on a multiresolution grid within Godunov-type solvers based on second-order discontinuous Galerkin (DG2) and first-order finite volume (FV1) methods. Multiwavelet adaptivity has been studied extensively with one-dimensional (1D) hydrodynamic models (Kesserwani et al., 2019), revealing that MWDG2 can be 20 times faster than uniform DG2 and 2 times faster than uniform FV1, while preserving the accuracy and robustness of the underlying formulation. The potential of the MWDG2 scheme has yet to be studied for two-dimensional (2D) modelling, but this requires a design that robustly and efficiently solves the 2D shallow water equations (SWE) with complex source terms and wetting and drying. This paper presents a two-dimensional MWDG2 scheme that: (1) adopts a slope-decoupled DG2 solver as a reference scheme, for its ability to deliver well-balanced piecewise-planar solutions shaped by a simplified 3-component basis; and, (2) adapts the multiresolution analysis of multiwavelets for compatibility with the slope-decoupled DG2 basis. A scaled reformulation of slope-decoupled DG2 is presented alongside two multiwavelet approaches that yield MWDG2 schemes with similar properties, and a Haar wavelet FV1 (HFV1) variant for adapting piecewise-constant model data. The performance of the adaptive HFV1 and MWDG2 solvers is explored alongside their uniform counterparts, while analysing their accuracy, efficiency, grid-coarsening ability, reliability in handling wet-dry fronts across steep bed-slopes, and ability to capture features relevant to practical hydraulic modelling. The results indicate a particular multiwavelet approach that allows the MWDG2 scheme to exploit its grid-coarsening ability for the widest range of flow types. Results also indicate that the proposed (multi)wavelet-based adaptive schemes are even more efficient for the 2D case. Accompanying model software is openly available online.

Keywords: Scaled discontinuous Galerkin and finite volume hydraulic models, Adaptive multiresolution schemes, robust and efficient 2D approaches, Performance analyses and comparisons

1. Introduction

Explicit two-dimensional (2D) Godunov-type hydrodynamic solvers are widely used due to their robust treatment of moving wet-dry fronts and their ability to capture a wide range of transient flows (Mignot et al., 2006; Begnudelli et al., 2008; Néelz and Pender, 2013; Teng et al., 2017; Glenis et al., 2018; Guinot et al., 2018; Xia and Liang, 2018; Echeverribar et al., 2019; Sanders and Schubert, 2019; Xia et al., 2019; Qin et al., 2019; Morales-Hernández et al., 2020). These solvers

*Corresponding author
Email addresses: g.kesserwani@sheffield.ac.uk (Georges Kesserwani), m.sharifian@sheffield.ac.uk (Mohammad Kazem Sharifian)

often adopt a first-order finite volume (FV1) discretisation to solve the conservative form of the shallow water equations (SWE) by representing flow variables and terrain data as piecewise-constant fields defined on a uniform raster grid. Thanks to the locality of FV1, element-wise updates to flow variables only require data from neighbouring elements to evaluate numerical fluxes across interfaces (Toro and Garcia-Navarro, 2007). However, due to its first-order accuracy, an FV1 solver can require excessively fine horizontal resolution (< 2 m) to alleviate numerical diffusion errors (Lhomme et al., 2010). The second-order discontinuous Galerkin (DG2) discretisation extends the finite volume principle by representing flow variables and terrain data as piecewise-planar fields (Gourgue et al., 2009; Trahan and Dawson, 2012; Dawson et al., 2013; Kesserwani and Wang, 2014; Vater et al., 2019). Previous studies identified practical benefits to adopting local piecewise-planar approximations in 2D hydrodynamic models to improve the capture of free-surface flow spillage over steep topography (Buttinger-Kreuzhuber et al., 2019; Sanders and Schubert, 2019), and to improve velocity predictions on coarse meshes leading to better rating curve predictions (Kesserwani, 2013; Kesserwani and Wang, 2014; Minatti et al., 2016). While DG2 solvers can improve predictions on coarse meshes, they suffer from excessive runtime cost (> 3 times the cost of FV1) (Kesserwani, 2013). Hence, a modelling approach is still needed to attain accurate predictions at a cost that is competitive with FV1.

Multiwavelets (MW) are directly compatible with the piecewise-polynomial bases of FV and DG solvers and provide a multiresolution analysis that enables a dynamically adaptive solution that overcomes known shortcomings of conventional adaptive mesh refinement methods (Dahmen, 1997; Cohen et al., 2001; Müller, 2003; Lamby et al., 2005; Díaz Calle et al., 2005; Smith et al., 2008; Dahmen et al., 2010; Archibald et al., 2011; Hovhannisyan et al., 2013; Haleem et al., 2015; Kesserwani et al., 2015; Gerhard et al., 2015b; Gerhard and Müller, 2016; Guo and Cheng, 2016; Wang et al., 2016; Sharifian et al., 2019; Tao et al., 2019). Kesserwani et al. (2019) adopted multiwavelets and Haar wavelets to formulate adaptive one-dimensional (1D) MWDG2 and HFV1 hydrodynamic solvers, and found that the 1D-MWDG2 solver achieved the accuracy of the uniform DG2 solver for a runtime cost less than the uniform FV1 solver. The potential of (multi)wavelet-based adaptivity has yet to be explored for practical 2D hydrodynamic modelling, which requires the formulation of 2D-specific approaches for a robust 2D-MWDG2 scheme (Caviedes-Voullième and Kesserwani, 2015; Caviedes-Voullième et al., 2020).

As an adaptive 2D-MWDG2 scheme inherits the properties of its underlying uniform 2D-DG2 reference scheme (Gerhard et al., 2015a), the formulation of a robust 2D-MWDG2 scheme boils down to selecting two key ingredients. The first ingredient is a robust and efficient uniform 2D-DG2 solver serving as the underlying reference scheme. This 2D-DG2 solver must avoid spurious momentum errors at wet-dry fronts across very steep bed-slopes and reduce runtime cost (Caviedes-Voullième et al., 2020). Hence, the *slope-decoupled* 2D-DG2 solver (Kesserwani et al., 2018) is selected, in which piecewise-planar solutions are spanned by 3 scale coefficients—an average and two directionally independent slopes—on a simplified stencil. This slope-coupled solver ensures all scale coefficients are well-balanced irrespective of bed slope steepness and the presence of wet-dry fronts, and reduces the runtime cost per element by 2.6 times compared to a standard 2D-DG2 form while preserving second-order accuracy (Kesserwani et al., 2018). The second ingredient is a multiwavelet-based multiresolution analysis, which must be compatible with the scaling basis of the reference scheme (Caviedes-Voullième and Kesserwani, 2015). This paper proposes two such approaches and investigates their potential for robust and efficient hydrodynamic modelling in a practical context.

The remainder of the paper is structured as follows. Section 2 presents the 2D-DG2 reference scheme and two compatible multiwavelet-based approaches that facilitate a dynamically adaptive solution using a multiresolution analysis of piecewise-planar flow and topography variables. In section 3, the accuracy, efficiency and robustness of the resulting 2D-MWDG2 solver is validated across six standard 2D test cases. Results are compared against an adaptive finite volume solver based on a direct simplification of the 2D-MWDG2 formulation, and against 2D-DG2 and 2D-FV1 solvers on uniform grids. The paper concludes with key findings summarised in section 4. This paper includes a concise presentation of the numerical formulations that are accessible to engineers and modellers. Further theoretical details are given by Jarczewska et al. (2015), Gerhard and Müller (2016) and Kesserwani et al. (2018, 2019). A Fortran 2003 implementation of the solvers is also available to download from Zenodo (Sharifian and Kesserwani, 2020), and instructions for running these solvers are provided in Appendix B. In the rest of the paper, the notation HFV1, MWDG2, FV1 and DG2 will be used by default to refer to 2D solvers, unless explicitly specified otherwise.

2. Adaptive MWDG2 scheme

This section presents the implementation of adaptive and robust MWDG2 solvers for the 2D depth-averaged SWE. The implementation extends the 1D formulation in Kesserwani et al. (2019), by first re-scaling the slope-decoupled DG2 formulation (Kesserwani et al., 2018) and then supporting it with compatible multiwavelet bases (Section 2.2).

The conservative form of the SWE defined on a 2D spatial domain Ω, can be written as (Toro and Garcia-Navarro, 2007):

$$\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) + \partial_y \mathbf{G}(\mathbf{U}) = \mathbf{S}_b(\mathbf{U}) + \mathbf{S}_f(\mathbf{U}) \tag{1}$$

where $\mathbf{U}(x, y, t)$ is the vector of the state variables at location (x, y) and time t, $\mathbf{F}(\mathbf{U})$ and $\mathbf{G}(\mathbf{U})$ are the spatial flux vectors, $\mathbf{S}_b(\mathbf{U})$ and $\mathbf{S}_f(\mathbf{U})$ are the vectors that represent bed and friction slope terms, and ∂_t, ∂_x and ∂_y represent partial derivatives with respect to t, x and y, respectively. Each of these vectors has three physical components, and they are expressed as:

$$\begin{gathered}\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \mathbf{F} = \begin{bmatrix} hu \\ \dfrac{(hu)^2}{h} + \dfrac{1}{2}gh^2 \\ huv \end{bmatrix}, \mathbf{G} = \begin{bmatrix} hv \\ huv \\ \dfrac{(hv)^2}{h} + \dfrac{1}{2}gh^2 \end{bmatrix}, \\ \mathbf{S}_b = \begin{bmatrix} 0 \\ -gh\partial_x z \\ -gh\partial_y z \end{bmatrix}, \mathbf{S}_f = \begin{bmatrix} 0 \\ -C_f u\sqrt{u^2+v^2} \\ -C_f v\sqrt{u^2+v^2} \end{bmatrix},\end{gathered} \tag{2}$$

where $h\,(\mathrm{m})$ is the water depth, $u\,(\mathrm{m/s})$ and $v\,(\mathrm{m/s})$ are the velocity components in two Cartesian directions and $g\,(\mathrm{m/s^2})$ is the acceleration due to gravity. In $\mathbf{S}_b$, $z(x, y)$ represents the bottom topography elevation, while the term $C_f = g\,n_{\mathrm{M}}^2/h^{1/3}$ in $\mathbf{S}_f$ includes the Manning's bed roughness coefficient n_{M} ($\mathrm{s^{-1}m^{1/3}}$).

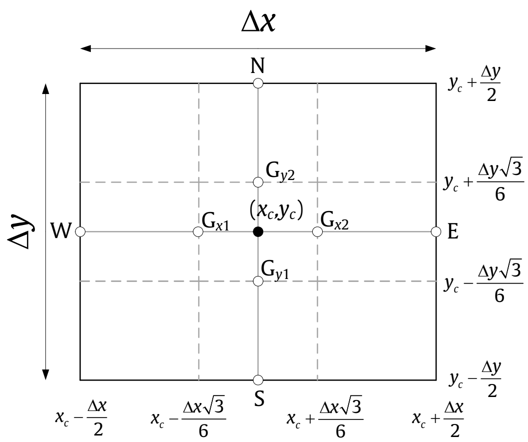

Figure 1. A rectangular element Q_c on which the slope-decoupled DG2 form (Kesserwani et al., 2018) is applied. Points E, W, N and S denote the face-centred nodes at the eastern, western, northern and southern faces. Points G$_{xi}$ and G$_{yi}$ (i = 1, 2) denote Gaussian evaluation points along the x- and y-directional centrelines.

2.1. Scaled DG2 reference solver

The domain Ω is decomposed into $M \times N$ non-overlapping and uniform rectangular elements Q_c $(c = 1, \ldots, M \times N)$. Each element has the size of $\Delta x \times \Delta y$ and is centered at (x_c, y_c), as shown in Fig. 1. An element Q_c can therefore be written as $Q_c = [x_c - \Delta x/2, x_c + \Delta x/2] \times [y_c - \Delta y/2, y_c + \Delta y/2]$. Any element Q_c can be mapped into a reference element $[-1, 1]^2 = [-1, 1] \times [-1, 1]$ via the transformation $(x, y) \in Q_c \to (\xi, \eta) \in [-1, 1]^2$ where $\xi(x) = 2(x - x_c)/\Delta x$ and $\eta(y) = 2(y - y_c)/\Delta y$; whereas, $x(\xi) = x_c + \xi(\Delta x/2)$ and $y(\eta) = y_c + \eta(\Delta y/2)$ can be used to position any element Q_c from the reference element $[-1, 1]^2$.

2.1.1. Bi-orthonormal bases and DG2 operators

In keeping with the slope-decoupled DG2 form (Kesserwani et al., 2018) and following similar reasoning as in Kesserwani et al. (2019), bi-orthonormal bases can be constructed from the 2D tensor product of the standard 1D Legendre basis functions truncated to first order monomials. This results in the following 2D orthogonal basis defined on the reference element $[-1, 1]^2$:

$$\mathbf{P} = \begin{bmatrix} P^0 & P^{1x} & P^{1y} \end{bmatrix}^{\mathrm{T}} \quad \text{with} \quad P^0(\xi,\eta) = 1, \quad P^{1x}(\xi,\eta) = \xi, \quad P^{1y}(\xi,\eta) = \eta \tag{3}$$

where superscripts 0, $1x$ and $1y$ indicate the components of $\mathbf{P}$ spanning an average and two directionally independent slopes.

The 2D basis $\mathbf{P}$ is normalised based on the L^2-norm defined by the following dot product:

$$\langle f, g \rangle = \int_{-1}^{1}\int_{-1}^{1} f(\xi,\eta) g(\xi,\eta) d\xi d\eta \tag{4}$$

This yields a scaled basis, denoted by $\hat{\mathbf{P}}$, that is also defined on $[-1, 1]^2$ as:

$$\hat{\mathbf{P}} = \begin{bmatrix} \hat{P}^0 & \hat{P}^{1x} & \hat{P}^{1y} \end{bmatrix}^{\mathrm{T}} \quad \text{with} \quad \hat{P}^0(\xi,\eta) = \frac{P^0(\xi,\eta)}{\sqrt{2}}, \quad \hat{P}^{1x}(\xi,\eta) = \frac{\sqrt{3}P^{1x}(\xi,\eta)}{\sqrt{2}}, \quad \hat{P}^{1y}(\xi,\eta) = \frac{\sqrt{3}P^{1y}(\xi,\eta)}{\sqrt{2}} \tag{5}$$

From the components of the scaled basis $\hat{\mathbf{P}}$, i.e. $\hat{P}^0$, $\hat{P}^{1x}$ and $\hat{P}^{1y}$, the components of a local primal basis $\boldsymbol{\phi}_c$ and a local dual basis $\tilde{\boldsymbol{\phi}}_c$ can be defined over an element Q_c:

$$\boldsymbol{\phi}_c = \left[\varphi_c^0(x,y) \quad \varphi_c^{1x}(x,y) \quad \varphi_c^{1y}(x,y)\right]^{\mathrm{T}} \tag{6}$$

$$\tilde{\boldsymbol{\phi}}_c = \left[\tilde{\varphi}_c^0(x,y) \quad \tilde{\varphi}_c^{1x}(x,y) \quad \tilde{\varphi}_c^{1y}(x,y)\right]^{\mathrm{T}} \tag{7}$$

with

$$\varphi_c^K(x,y) = \sqrt{2}\hat{P}^K(\xi,\eta) \quad (K = 0, 1x, 1y \text{ and } (x,y) \in Q_c) \tag{8}$$

$$\tilde{\varphi}_c^K(x,y) = \frac{\varphi_c^K(x,y)}{\Delta x \Delta y} \quad (K = 0, 1x, 1y \text{ and } (x,y) \in Q_c) \tag{9}$$

Each of the primal basis and the dual basis is compactly-supported, orthogonal and discontinuous at the faces shared by Q_c and its adjacent neighbours.

A scaled and slope-decoupled DG2 approach seeks a local planar approximate solution, $\mathbf{U}_h$, to the flow vector $\mathbf{U}$, that is expanded onto the primal basis $\boldsymbol{\phi}_c$. This leads to the following shape for $\mathbf{U}_h$ over each element Q_c:

$$\mathbf{U}_h(x(\xi), y(\eta), t)|_{Q_c} = \mathbf{U}_c^0(t) + \sqrt{3}\,\xi(x)\,\mathbf{U}_c^{1x}(t) + \sqrt{3}\,\eta(y)\,\mathbf{U}_c^{1y}(t) \tag{10}$$

where $\mathbf{U}_c^0(t)$, $\mathbf{U}_c^{1x}(t)$ and $\mathbf{U}_c^{1y}(t)$ are expansion coefficients, representing an average and two slopes that are directionally independent. These coefficients will be hereafter referred to as DG2 modes.

The DG2 modes are initialised by projecting a given initial condition $\mathbf{U}(x,y,0) = \mathbf{U}_0(x,y)$ onto each component $\tilde{\varphi}_c^K$ of the dual basis, i.e. $\mathbf{U}_c^K(0) = \left\langle \mathbf{U}_0, \tilde{\varphi}_c^K \right\rangle$, where $\langle \boxdot, \boxdot \rangle$ is the dot product defined by the L^2-norm (Eq. (4)). The DG2 modes are initialised by averaging the initial condition evaluated at the four element corners to obtain values at the eastern, western, northern and southern face centres, $\mathbf{U}_0(\mathrm{E})$, $\mathbf{U}_0(\mathrm{W})$, $\mathbf{U}_0(\mathrm{N})$ and $\mathbf{U}_0(\mathrm{S})$. Then, the initial DG2 modes can also be expressed as (Kesserwani et al., 2018):

$$\mathbf{U}_c^0(0) = \frac{1}{2}\left[\mathbf{U}_0(\mathrm{E}) + \mathbf{U}_0(\mathrm{W})\right] = \frac{1}{2}\left[\mathbf{U}_0(\mathrm{N}) + \mathbf{U}_0(\mathrm{S})\right] \tag{11a}$$

$$\mathbf{U}_c^{1x}(0) = \frac{1}{2\sqrt{3}}\left[\mathbf{U}_0(\mathrm{E}) - \mathbf{U}_0(\mathrm{W})\right] \tag{11b}$$

$$\mathbf{U}_c^{1y}(0) = \frac{1}{2\sqrt{3}}\left[\mathbf{U}_0(\mathrm{N}) - \mathbf{U}_0(\mathrm{S})\right] \tag{11c}$$

The approximate solution for the topography $z_h(x,y)$ is initialised in the same way as Eqs. (11a)-(11c) to obtain the time-invariant DG2 modes denoted z_c^0, z_c^{1x} and z_c^{1y}. This initialisation procedure ensures that z_h remains continuous at all face centres on a uniform grid.

Spatial DG2 operators $\mathbf{L}_c^K$ are applied in order to update the DG2 modes of the flow vector, $\mathbf{U}_c^K(t)$. These DG2 operators are derived by applying a standard finite element weak formulation to Eq. (1) — subject to using the dual basis $\tilde{\boldsymbol{\phi}}_c$ as a test function, replacing $\mathbf{U}$ and z by $\mathbf{U}_h$ and z_h, and exploiting the bi-orthonormality property across the primal and dual basis (Kesserwani et al., 2019). They are applied element-wise by solving the following set of ODEs within a two-stage Runge-Kutta (RK2) time stepping:

$$\partial_t \mathbf{U}_c^K(t) = \mathbf{L}_c^K \quad (K = 0, 1x, 1y) \tag{12}$$

Simplified (slope-decoupled) forms for the spatial DG2 operators $\mathbf{L}_c^K$ can be obtained (Kesserwani et al., 2018), to read:

$$\mathbf{L}_c^0 = -\frac{1}{\Delta x}\left[\tilde{\mathbf{F}}_{\mathrm{E}} - \tilde{\mathbf{F}}_{\mathrm{W}}\right] - \frac{1}{\Delta y}\left[\tilde{\mathbf{G}}_{\mathrm{N}} - \tilde{\mathbf{G}}_{\mathrm{S}}\right] + \begin{bmatrix} 0 \\ \frac{-2g\sqrt{3}}{\Delta x}\bar{h}_c^{0x}\bar{z}_c^{1x} \\ \frac{-2g\sqrt{3}}{\Delta y}\bar{h}_c^{0y}\bar{z}_c^{1y} \end{bmatrix} \tag{13a}$$

$$\mathbf{L}_c^{1x} = -\frac{\sqrt{3}}{\Delta x}\left(\tilde{\mathbf{F}}_{\mathrm{E}} + \tilde{\mathbf{F}}_{\mathrm{W}} - \mathbf{F}\left(\bar{\mathbf{U}}_c^{0x} - \bar{\mathbf{U}}_c^{1x}\right) - \mathbf{F}\left(\bar{\mathbf{U}}_c^{0x} + \bar{\mathbf{U}}_c^{1x}\right) + \begin{bmatrix} 0 \\ 2g\bar{h}_c^{1x}\bar{z}_c^{1x} \\ 0 \end{bmatrix}\right) \tag{13b}$$

$$\mathbf{L}_c^{1y} = -\frac{\sqrt{3}}{\Delta y}\left(\tilde{\mathbf{G}}_{\mathrm{N}} + \tilde{\mathbf{G}}_{\mathrm{S}} - \mathbf{G}\left(\bar{\mathbf{U}}_c^{0y} - \bar{\mathbf{U}}_c^{1y}\right) - \mathbf{G}\left(\bar{\mathbf{U}}_c^{0y} + \bar{\mathbf{U}}_c^{1y}\right) + \begin{bmatrix} 0 \\ 0 \\ 2g\bar{h}_c^{1y}\bar{z}_c^{1y} \end{bmatrix}\right) \tag{13c}$$

Note that the friction source term, $\mathbf{S}_f$, is excluded from the spatial operators in Eqs. (13a)-(13c). Rather, it is integrated separately as piecewise planar expansion $(\mathbf{S}_f)_h$, via pointwise evaluations of the standard implicit integration at each element centre and Gaussian points (Kesserwani and Liang, 2010). The terms $\tilde{\mathbf{F}}_{\mathrm{E}}$, $\tilde{\mathbf{F}}_{\mathrm{W}}$, $\tilde{\mathbf{G}}_{\mathrm{N}}$ and $\tilde{\mathbf{G}}_{\mathrm{S}}$ are inter-elemental fluxes that are evaluated via a two-argument numerical flux function, $\tilde{\mathbf{F}}$, based on the HLL approximate Riemann solver (Toro, 2001). The two arguments of the flux function are the limits of the approximate solutions $\mathbf{U}_h$, at either side of the face-centred nodes E, W, N and S (Fig. 1). These limits must be revised based on a wetting and drying condition to simultaneously ensure well-balancedness and depth-positivity preservation (Kesserwani et al., 2018). For instance, at the eastern face-centred node, E, the original limits of $\mathbf{U}_h$ are $\mathbf{U}_{\mathrm{E}}^- = \mathbf{U}_c^0(t) + \sqrt{3}\mathbf{U}_c^{1x}(t)$ and $\mathbf{U}_{\mathrm{E}}^+ = \mathbf{U}_{nei_{\mathrm{E}}}^0(t) - \sqrt{3}\mathbf{U}_{nei_{\mathrm{E}}}^{1x}(t)$, with '$nei_{\mathrm{E}}$' denoting the index of direct neighbour from the eastern side. For z_h, given its continuity property, it has equal limits at E, i.e. $z_h(\mathrm{E}^\pm) = z_{\mathrm{E}}$. Revised limits for the flow vector variables, appended with the superscript 'star', are reconstructed as: $h_{\mathrm{E}}^{\pm,*} = \max(0, h_{\mathrm{E}}^\pm)$, $(hu)_{\mathrm{E}}^{\pm,*} = h_{\mathrm{E}}^{\pm,*}u_{\mathrm{E}}^\pm$ and $(hv)_{\mathrm{E}}^{\pm,*} = h_{\mathrm{E}}^{\pm,*}v_{\mathrm{E}}^\pm$. Note that $u_{\mathrm{E}}^\pm = (hu)_{\mathrm{E}}^\pm/h_{\mathrm{E}}^\pm$ and $v_{\mathrm{E}}^\pm = (hv)_{\mathrm{E}}^\pm/h_{\mathrm{E}}^\pm$ are velocity limits evaluated from the original flow vector variable limits when $h_{\mathrm{E}}^\pm > tol_{\mathrm{dry}}$, or zeroed otherwise, and that tol_{dry} is a user-selected tolerance for defining a dry element. The continuous limit of the topography z_{E} must also be revised, i.e. $z_{\mathrm{E}}^* = z_{\mathrm{E}} - \max(0, h_{\mathrm{E}}^-)$, to preserve well-balancedness when a motionless body of water is blocked by a high wall (Liang and Marche, 2009). The revised flow vector limits $\mathbf{U}_{\mathrm{E}}^{\pm,*} = \left[h_{\mathrm{E}}^{\pm,*} \quad (hu)_{\mathrm{E}}^{\pm,*} \quad (hv)_{\mathrm{E}}^{\pm,*}\right]^{\mathrm{T}}$ are then used to evaluate flux $\tilde{\mathbf{F}}_{\mathrm{E}} = \tilde{\mathbf{F}}(\mathbf{U}_{\mathrm{E}}^{-,*}, \mathbf{U}_{\mathrm{E}}^{+,*})$ — note that fluxes $\tilde{\mathbf{F}}_{\mathrm{W}}$, $\tilde{\mathbf{G}}_{\mathrm{N}}$ and $\tilde{\mathbf{G}}_{\mathrm{S}}$ are evaluated by analogy after reconstructing revised flow vector limits $\mathbf{U}_{\mathrm{W}}^{\pm,*}$, $\mathbf{U}_{\mathrm{N}}^{\pm,*}$ and $\mathbf{U}_{\mathrm{S}}^{\pm,*}$ and their revised continuous topography limits z_{W}^*, z_{N}^* and z_{S}^*. Moreover, the limits belonging to Q_c must be reused to produce DG2 modes that are also well-balanced and depth-positivity preserving. The revised DG2 modes are appended with a 'bar' and can be generated by reapplying Eqs. (11a)-(11c) as follows:

$$\bar{\mathbf{U}}_c^{0x} = \frac{1}{2}\left[\mathbf{U}_{\mathrm{E}}^{-,*} + \mathbf{U}_{\mathrm{W}}^{+,*}\right] \tag{14a}$$

$$\bar{\mathbf{U}}_c^{0y} = \frac{1}{2}\left[\mathbf{U}_{\mathrm{N}}^{-,*} + \mathbf{U}_{\mathrm{S}}^{+,*}\right] \tag{14b}$$

$$\bar{\mathbf{U}}_c^{1x} = \frac{1}{2\sqrt{3}}\left[\mathbf{U}_{\mathrm{E}}^{-,*} - \mathbf{U}_{\mathrm{W}}^{+,*}\right] \quad \text{and} \quad \bar{z}_c^{1x} = \frac{1}{2\sqrt{3}}\left[z_{\mathrm{E}}^{*} - z_{\mathrm{W}}^{*}\right] \tag{14c}$$

$$\bar{\mathbf{U}}_c^{1y} = \frac{1}{2\sqrt{3}}\left[\mathbf{U}_{\mathrm{N}}^{-,*} - \mathbf{U}_{\mathrm{S}}^{+,*}\right] \quad \text{and} \quad \bar{z}_c^{1y} = \frac{1}{2\sqrt{3}}\left[z_{\mathrm{N}}^{*} - z_{\mathrm{S}}^{*}\right] \tag{14d}$$

Replacing the evaluated fluxes $\tilde{\mathbf{F}}_{\mathrm{E}}$, $\tilde{\mathbf{F}}_{\mathrm{W}}$, $\tilde{\mathbf{G}}_{\mathrm{N}}$ and $\tilde{\mathbf{G}}_{\mathrm{S}}$ and the revised DG2 modes $\bar{\mathbf{U}}_c^{0x} = \left[\bar{h}_c^{0x} \quad (\bar{hu})_c^{0x} \quad (\bar{hv})_c^{0x}\right]^{\mathrm{T}}$, $\bar{\mathbf{U}}_c^{0y} = \left[\bar{h}_c^{0y} \quad (\bar{hu})_c^{0y} \quad (\bar{hv})_c^{0y}\right]^{\mathrm{T}}$, $\bar{\mathbf{U}}_c^{1x} = \left[\bar{h}_c^{1x} \quad (\bar{hu})_c^{1x} \quad (\bar{hv})_c^{1x}\right]^{\mathrm{T}}$ and $\bar{\mathbf{U}}_c^{1y} = \left[\bar{h}_c^{1y} \quad (\bar{hu})_c^{1y} \quad (\bar{hv})_c^{1y}\right]^{\mathrm{T}}$ in Eqs. (13a)-(13c), leads to a robust numerical evaluation for the scaled DG2 operators without spurious momentum errors across wet-dry fronts located at very steep bed-slopes. A theoretical and diagnostic verification of these aspects can be found in Kesserwani et al. (2018).

2.1.2. *Extension to multiresolution bases*

From the same L^2-orthonormal basis $\hat{\mathbf{P}}$, a series of child bases $\left\{\hat{\mathbf{P}}^{(n)}\right\}_n$ can be defined (Gerhard and Müller, 2016), where n is a positive integer indicating the resolution level, which will be used within brackets, (n), to avoid notation confusion with other indexes. These child bases arise from the *father basis* $\hat{\mathbf{P}}^{(0)} = \hat{\mathbf{P}}$ and preserve its properties. The supports of these child bases at any resolution level (n) can be associated with a grid $g^{(n)}$ based on n dyadic subdivisions of the support $[-1,1]^2$ of $\hat{\mathbf{P}}$. Hence, $g^{(n)}$ spans $[-1,1]^2$ such that $g^{(n)} = \cup_{i,j=0}^{2^n-1} Q_{i,j}^{(n)}$, where $\left\{Q_{i,j}^{(n)}\right\}_{i,j=0,1,\ldots,2^n-1}$ is a set of non-overlapping sub-divisions of $[-1,1]^2$. A sub-division $Q_{i,j}^{(n)}$ will be referred to as a *sub-element* of $[-1,1]^2$, taking the following form:

$$Q_{i,j}^{(n)} = [\chi_{i-1/2}, \chi_{i+1/2}] \times [\Upsilon_{j-1/2}, \Upsilon_{j+1/2}] \tag{15}$$

where $\chi_{i-1/2} = -1 + \frac{2}{2^n} i$ and $\Upsilon_{j-1/2} = -1 + \frac{2}{2^n} j$ are interface points forming sub-elements $\left\{Q_{i,j}^{(n)}\right\}_{i,j=0,1,\ldots,2^n-1}$, and the indexes $i, j = 0, 1, \ldots, 2^n - 1$ represent the position of $Q_{i,j}^{(n)}$ in $g^{(n)}$, on which the components $\hat{\mathbf{P}}_{i,j}^{(n)}$ of the basis $\hat{\mathbf{P}}^{(n)}$ can be defined by translation and dilation of $\hat{\mathbf{P}}$ as follows:

$$\hat{\mathbf{P}}_{i,j}^{(n)}(\chi, \Upsilon) = 2^n\, \hat{\mathbf{P}}\left(2^n(\chi+1) - 2i - 1, 2^n(\Upsilon+1) - 2j - 1\right) \quad (\chi, \Upsilon) \in Q_{i,j}^{(n)} \tag{16}$$

From the compact-support and L^2-orthonormality properties of $\left\{\hat{\mathbf{P}}^{(n)}\right\}_n$, the grids $\left\{g^{(n)}\right\}_n$ form a hierarchy spanning $[-1,1]^2$, i.e. $\cup_n g^{(n)} = [-1,1]^2$. They are globally nested across all resolution levels while having local and non-overlapping support at each level (n). In this sense, $\left\{\hat{\mathbf{P}}^{(n)}\right\}_n$ can be referred to as *scaling bases*.

Similarly, on a grid element Q_c a hierarchy of nested grids $\left\{g_c^{(n)}\right\}_n$ can be defined such that $g_c^{(n)} = \cup_{i,j=0}^{2^n-1} Q_{i,j,c}^{(n)}$ with $\left\{Q_{i,j,c}^{(n)}\right\}_{i,j=0,1,\ldots,2^n-1}$ now denoting sub-divisions of Q_c, where $Q_{i,j,c}^{(n)}$ represents a sub-element of Q_c at a position i, j and the grid spacings $\Delta x^{(n)} = \Delta x/2^n$ and $\Delta y^{(n)} = \Delta y/2^n$ at resolution level (n). For convenience of presentation, sub-element $Q_{i,j,c}^{(n)}$ will hereafter be denoted by $Q_e^{(n)}$ where index "e" is shorthand for "i, j, c" to position it in Q_c. Thereby, sub-element $Q_e^{(n)}$ can be linked to $Q_{i,j}^{(n)}$ by translation into $[-1,1]^2$. On a sub-element $Q_e^{(n)} \in g_c^{(n)}$, bi-orthonormal dual and primal bases, denoted by $\boldsymbol{\phi}_e^{(n)}$ and $\tilde{\boldsymbol{\phi}}_e^{(n)}$, can be defined from the bases $\hat{\mathbf{P}}_{i,j}^{(n)}$ as:

$$\boldsymbol{\phi}_e^{(n)}(x, y) = \sqrt{2}\hat{\mathbf{P}}_{i,j}^{(n)}(\chi, \Upsilon) \quad (x, y) \in Q_e^{(n)} \subset Q_c \tag{17}$$

$$\tilde{\boldsymbol{\phi}}_e^{(n)}(x,y) = \frac{\boldsymbol{\phi}_e^{(n)}(x,y)}{\Delta x^{(n)}\Delta y^{(n)}} \quad (x,y) \in Q_e^{(n)} \subset Q_c \tag{18}$$

On the bi-orthonormal bases (17) and (18), the DG2 operators described in Eqs. (13a)-(13c), remain valid for any sub-element Q_e, subject to using index e instead of c, and with grid spacings $\Delta x^{(n)}$ and $\Delta y^{(n)}$ instead of Δx and Δy. These sub-elemental DG2 operators can be used to evolve the local DG2 modes, i.e. $\mathbf{U}_e^0(t)$, $\mathbf{U}_e^{1x}(t)$ and $\mathbf{U}_e^{1y}(t)$, at any sub-element Q_e within $\left\{g_c^{(n)}\right\}_n$. To ease the presentation in Section 2.2.3, these modes and those of the topography, i.e. z_e^0, z_e^{1x} and z_e^{1y}, will be considered component-wise. The scalar variable $u \in \{h, hu, hv, z\}$ will be used to represent any physical quantities in $\mathbf{U} = [h \quad hu \quad hv]^{\mathrm{T}}$ and z. The DG2 modes of any physical quantity u on any sub-element $Q_e^{(n)} \in \left\{g_c^{(n)}\right\}_n$ will therefore be denoted as $\boldsymbol{u}_e^{(n)} = [u_e^{0,(n)} \quad u_e^{1x,(n)} \quad u_e^{1y,(n)}]^{\mathrm{T}}$.

2.2. *Multiresolution analysis: 2D slope-decoupled approaches*

Multiresolution analysis is needed to analyse the behaviour of the DG2 modes over the hierarchy of nested grids $\left\{g_c^{(n)}\right\}_n$ defined on a grid element Q_c. Within this scope, the multiresolution analysis procedure requires multiwavelet bases, or *multiwavelets*, to be defined on the reference element $[-1,1]^2 = g^{(0)}$ (Jarczewska et al., 2015; Kesserwani et al., 2019), from a father basis, $\hat{\mathbf{P}}^{(0)}$, that is actually the scaled basis $\hat{\mathbf{P}}$ used to define the slope-decoupled DG2 expansion (Eq. (10), Section 2.1.1).

For the 2D case, three mother multiwavelets are needed to represent the encoded difference between the father basis $\hat{\mathbf{P}}^{(0)}$ on $g^{(0)}$ and its child basis $\hat{\mathbf{P}}^{(1)}$ on $g^{(1)}$ along the vertical, horizontal and diagonal directions (Daubechies, 1992). These mother multiwavelets will be denoted by $\boldsymbol{\Psi}_{[\alpha]}(\xi,\eta)$, $\boldsymbol{\Psi}_{[\beta]}(\xi,\eta)$ and $\boldsymbol{\Psi}_{[\gamma]}(\xi,\eta)$, respectively. They belong to $g^{(0)} \cap g^{(1)}$ and are, therefore, one level-of-resolution level higher than $\hat{\mathbf{P}}^{(0)}$. From the three mother multiwavelets, child multiwavelets $\left\{\boldsymbol{\Psi}_{[\Theta]}^{(n)}\right\}_n$, with $\Theta = \alpha, \beta, \gamma$, can be defined on the hierarchy of grids $\left\{g^{(n)}\right\}_n$ by translation and dilation as:

$$\boldsymbol{\Psi}_{[\Theta]_{i,j}}^{(n)}(\chi,\Upsilon) = 2^n\, \boldsymbol{\Psi}_{[\Theta]}(2^n(\chi+1) - 2i - 1, 2^n(\Upsilon+1) - 2j - 1) \quad (\chi,\Upsilon) \in Q_{i,j}^{(n)} \tag{19}$$

With the availability of both scaling and multiwavelet bases, there are two interchangeable ways to approximate a (given) scalar signal $s(\xi,\eta)$ defined on $[-1,1]^2$. The first way considers a single grid $g^{(n)} = \cup_{i,j=0}^{2^n-1} Q_{i,j}^{(n)}$ with the scaling bases $\hat{\mathbf{P}}^{(n)} = \left[\hat{\mathbf{P}}_{i,j}^{(n)}\right]_{i,j=0,1,\ldots,2^n-1}$ at a selected resolution (n) . It provides an approximation $s_h(\xi,\eta)$ of the signal $s(\xi,\eta)$ at resolution (n) by expanding it onto the bases $\hat{\mathbf{P}}^{(n)}$ as (Vuik and Ryan, 2014):

$$s_h(\xi,\eta) = \sum_{i,j=0}^{2^n-1} s_h(\chi,\Upsilon)|_{Q_{i,j}^{(n)}} = \sum_{i,j=0}^{2^n-1} \left\langle \mathbf{s}_{i,j}^{(n)}, \hat{\mathbf{P}}_{i,j}^{(n)} \right\rangle \tag{20}$$

where $\mathbf{s}_{i,j}^{(n)}$ denotes the vector including the local *scale coefficients* based on which $s_h(\chi,\Upsilon)|_{Q_{i,j}^{(n)}}$ is expanded onto the basis $\hat{\mathbf{P}}_{i,j}^{(n)}$. This form of approximation only involves scale coefficients from the grid $g^{(n)}$, and is therefore called *single-scale expansion*. The single-scale expansion provides a way to reconstruct the signal $s_h(\xi,\eta)$ on any single-scale grid $g^{(n)}$.

The second way to expand $s_h(\xi,\eta)$ is to involve the multiwavelet bases. By doing so, the single-scale expansion of Eq. (20) can be recursively decomposed to produce a so-called *multi-scale* expansion, which takes the form (Vuik and Ryan, 2014):

$$s_h(\xi,\eta) = s_h(\xi,\eta)|_{Q_{0,0}^{(0)}} + \sum_{l=0}^{n-1}\left(\sum_{i,j=0}^{2^n-1}\left\langle \mathbf{d}_{[\alpha]_{i,j}}^{(l)}, \mathbf{\Psi}_{[\alpha]_{i,j}}^{(l)}\right\rangle + \sum_{i,j=0}^{2^n-1}\left\langle \mathbf{d}_{[\beta]_{i,j}}^{(l)}, \mathbf{\Psi}_{[\beta]_{i,j}}^{(l)}\right\rangle + \sum_{i,j=0}^{2^n-1}\left\langle \mathbf{d}_{[\gamma]_{i,j}}^{(l)}, \mathbf{\Psi}_{[\gamma]_{i,j}}^{(l)}\right\rangle\right) \tag{21}$$

where $\mathbf{d}_{[\Theta]_{i,j}}^{(l)}$ with $\Theta = \alpha, \beta, \gamma$ are the vectors of local *details* also known as *detail coefficients* or *multiwavelet coefficients*. The multi-scale expansion, in Eq. (21), sums up the details of $s_h(\xi,\eta)$ throughout grids $g^{(0)},\ldots,g^{(n-1)}$ on top of its coarsest resolution description on $g^{(0)}$, i.e. $s_h(\xi,\eta)|_{Q_{0,0}^{(0)}}$. This makes it important to analyse the details of $s_h(\xi,\eta)$ living at successively higher resolutions. These details become increasingly significant with increasing levels of non-smoothness in $s_h(\xi,\eta)$, while remaining negligible where $s_h(\xi,\eta)$ is smooth. Therefore, the multi-scale expansion can be used to analyse the behaviour of the approximate signal $s_h(\xi,\eta)$ throughout the grids in the hierarchy $\left\{g^{(n)}\right\}_n$.

Besides the choice of the mother multiwavelets compatible with the slope-decoupled DG2 expansion (i.e. Eq. (10)), the algorithmic workflow of the multiresolution analysis follows the same procedure described in Gerhard et al. (2015a) and Kesserwani et al. (2019). The following Sections 2.2.1 and 2.2.2 will therefore focus on exploring the choice of mother multiwavelets, given its importance for preserving the robustness properties featuring in the reference DG2 solver (Section 2.1). In these sections, the subscript "i, j" will be omitted from the detail coefficients $\mathbf{d}_{[\alpha]_{i,j}}^{(l)}$, $\mathbf{d}_{[\beta]_{i,j}}^{(l)}$ and $\mathbf{d}_{[\gamma]_{i,j}}^{(l)}$ noting that they correspond to the mother multiwavelets $\mathbf{\Psi}_{[\Theta]}(\xi,\eta)$ on $g^{(0)} \cap g^{(1)}$, which are derived from the father basis $\hat{\mathbf{P}}^{(0)}$ on $g^{(0)}$. Starting from the father basis and mother multiwavelets, the so-called *two-scale transformation* (Kesserwani et al., 2019) formulae can be derived to link the coefficients across a sub-element on $g^{(0)}$ to its four child sub-elements on $g^{(1)}$ (Appendix A). Note that these same two-scale transformations remain applicable across $g^{(n)}$ and $g^{(n+1)}$ without loss of generality.

2.2.1. Approach 1: 4-component multiwavelets via full tensor product

A new approach proposed here involves an equivalent father basis $\hat{\mathbf{P}}$ formed by the 2D tensor product of the 1D basis $\hat{\mathbf{P}}_{1\mathrm{D}}$ (Eq. (5) in Kesserwani et al., 2019), subject to truncating the 4th scale coefficient associated with the nonlinear cross-dimensional slope (across the xy-direction). This truncation is needed because the full tensor product approach results in a basis that spans a superset of that spanned by the piecewise-planar DG2 basis. With this approach, $\hat{\mathbf{P}}$ is a 2×2 matrix defined as:

$$\hat{\mathbf{P}}(\xi,\eta) = \hat{\mathbf{P}}_{1\mathrm{D}}(\xi)\hat{\mathbf{P}}_{1\mathrm{D}}(\eta) \tag{22}$$

The vertical, horizontal and diagonal mother multiwavelet bases of $\hat{\mathbf{P}}$ can be defined by considering the 2D tensor products formed by $\hat{\mathbf{P}}_{1\mathrm{D}}$ and its 1D mother multiwavelet basis $\mathbf{\Psi}_{1\mathrm{D}}$ (Eq. (36) in Kesserwani et al., 2019), namely:

$$\begin{aligned}\mathbf{\Psi}_{[\alpha]}(\xi,\eta) &= \hat{\mathbf{P}}_{1\mathrm{D}}(\xi)\mathbf{\Psi}_{1\mathrm{D}}(\eta)\\ \mathbf{\Psi}_{[\beta]}(\xi,\eta) &= \mathbf{\Psi}_{1\mathrm{D}}(\xi)\hat{\mathbf{P}}_{1\mathrm{D}}(\eta)\\ \mathbf{\Psi}_{[\gamma]}(\xi,\eta) &= \mathbf{\Psi}_{1\mathrm{D}}(\xi)\mathbf{\Psi}_{1\mathrm{D}}(\eta)\end{aligned} \tag{23}$$

Therefore, Approach 1 adopts the following matrix structure for the scale coefficients, $\mathbf{s}^{(n)}$, and the detail coefficients, $\left\{\mathbf{d}_{[\Theta]}^{(n)}\right\}_{\Theta=\alpha,\beta,\gamma}$:

$$\mathbf{s}^{(n)} = \begin{bmatrix} s^{0,(n)} & s^{1x,(n)} \\ s^{1y,(n)} & s^{1xy,(n)} \end{bmatrix} \tag{24}$$

$$\mathbf{d}_{[\Theta]}^{(n)} = \begin{bmatrix} d_{[\Theta]}^{0,(n)} & d_{[\Theta]}^{1x,(n)} \\ d_{[\Theta]}^{1y,(n)} & d_{[\Theta]}^{1xy,(n)} \end{bmatrix} \tag{25}$$

In Eqs. (24) and (25), superscripts 0, $1x$, $1y$ and $1xy$ refer to components of the full DG2 expansion in $\hat{\mathbf{P}}^{(n)}$; whereas, the subscript Θ is an index to represent the horizontal, vertical and diagonal components of multiwavelets, spanning the complement of $\hat{\mathbf{P}}^{(n)}$ in $\hat{\mathbf{P}}^{(n+1)}$. Here, compatibility with the slope-decoupled DG2 expansion, i.e. in Eq. (5), is achieved by zeroing the component $s^{1xy,(n)}$ in Eq. (24), but keeping $d_{[\Theta]}^{1xy,(n)}$ active (i.e. not zeroed).

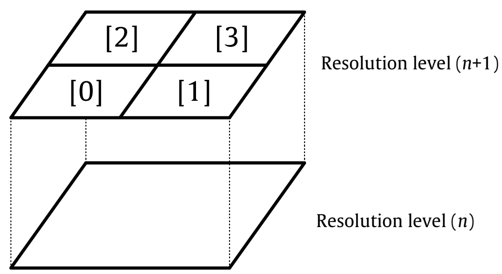

Figure 2. Index $[m]$, $m = 0, 1, 2, 3$, used to refer to the four child sub-elements at resolution level $(n + 1)$, with their vector of scale coefficients having subscripts $[m]$. For a parent sub-element at resolution level (n), the vector of scale coefficients is referred to without involving a subscript.

Given the recursive nature of the analysis, it suffices to describe the relationships linking model data between two subsequent resolution levels (n) and $(n + 1)$. From now on, the scale coefficients at resolution level (n) will be expressed without a subscript as in Eq. (24). Whereas, at resolution level $(n + 1)$, the scale coefficients of the four child sub-elements will be denoted by $\mathbf{s}_{[m]}^{(n+1)}$, with subscript m in square brackets referring to the index of each of the four child sub-elements 0, 1, 2 and 3, as shown in Fig. 2.

A key advantage of Approach 1 is that it allows the reuse of the same 2×2 (low- and high-pass) filter matrices $\mathbf{H}^{0,1}$ and $\mathbf{G}^{0,1}$ derived for the 1D case (i.e. Eqs. (34)-(35) and (40)-(41) in Kesserwani et al., 2019), which are also listed here in Eqs. (A.4)-(A.5) in Appendix A. From these filters, the following relationships for *encoding* the scale and detail coefficients at resolution level (n) from the scale coefficients of its four child sub-elements at resolution level $(n + 1)$ can be established:

$$\begin{aligned}
\mathbf{s}^{(n)} &= \mathbf{H}^0 \left[\mathbf{H}^0[\mathbf{s}_{[0]}^{(n+1)}]^{\mathrm{T}} + \mathbf{H}^1[\mathbf{s}_{[2]}^{(n+1)}]^{\mathrm{T}}\right]^{\mathrm{T}} + \mathbf{H}^1 \left[\mathbf{H}^0[\mathbf{s}_{[1]}^{(n+1)}]^{\mathrm{T}} + \mathbf{H}^1[\mathbf{s}_{[3]}^{(n+1)}]^{\mathrm{T}}\right]^{\mathrm{T}} \\
\mathbf{d}_{[\alpha]}^{(n)} &= \mathbf{H}^0 \left[\mathbf{G}^0[\mathbf{s}_{[0]}^{(n+1)}]^{\mathrm{T}} + \mathbf{G}^1[\mathbf{s}_{[2]}^{(n+1)}]^{\mathrm{T}}\right]^{\mathrm{T}} + \mathbf{H}^1 \left[\mathbf{G}^0[\mathbf{s}_{[1]}^{(n+1)}]^{\mathrm{T}} + \mathbf{G}^1[\mathbf{s}_{[3]}^{(n+1)}]^{\mathrm{T}}\right]^{\mathrm{T}} \\
\mathbf{d}_{[\beta]}^{(n)} &= \mathbf{G}^0 \left[\mathbf{H}^0[\mathbf{s}_{[0]}^{(n+1)}]^{\mathrm{T}} + \mathbf{H}^1[\mathbf{s}_{[2]}^{(n+1)}]^{\mathrm{T}}\right]^{\mathrm{T}} + \mathbf{G}^1 \left[\mathbf{H}^0[\mathbf{s}_{[1]}^{(n+1)}]^{\mathrm{T}} + \mathbf{H}^1[\mathbf{s}_{[3]}^{(n+1)}]^{\mathrm{T}}\right]^{\mathrm{T}} \\
\mathbf{d}_{[\gamma]}^{(n)} &= \mathbf{G}^0 \left[\mathbf{G}^0[\mathbf{s}_{[0]}^{(n+1)}]^{\mathrm{T}} + \mathbf{G}^1[\mathbf{s}_{[2]}^{(n+1)}]^{\mathrm{T}}\right]^{\mathrm{T}} + \mathbf{G}^1 \left[\mathbf{G}^0[\mathbf{s}_{[1]}^{(n+1)}]^{\mathrm{T}} + \mathbf{G}^1[\mathbf{s}_{[3]}^{(n+1)}]^{\mathrm{T}}\right]^{\mathrm{T}}
\end{aligned} \tag{26}$$

Given a user-defined maximum resolution level L, Eq. (26) can be applied in a descending order starting from resolution level $(L - 1)$ to *encode* (or extract) the scale and detail coefficients at any sub-element on $g^{(n)}$ from the scale coefficients of its four child sub-elements on $g^{(n+1)}$, as shown in Fig. 2. This results in a multi-scale expansion (Eq. (21)) that allows to span and access the detail coefficients within the whole hierarchy of grids $\left\{g^{(n)}\right\}_{n=0,1,\ldots,L}$.

Similarly, a relationship to *decode* (or generate) the scale coefficients at the four child sub-elements at resolution $(n + 1)$

from the scale and detail coefficients of the parent sub-element at resolution (n) can be derived to write:

$$
\begin{aligned}
\mathbf{s}_{[0]}^{(n+1)} &= [\mathbf{H}^0]^{\mathrm{T}}\left[[\mathbf{H}^0]^{\mathrm{T}}[\mathbf{s}^{(n)}]^{\mathrm{T}} + [\mathbf{G}^0]^{\mathrm{T}}[\mathbf{d}_{[\alpha]}^{(n)}]^{\mathrm{T}}\right]^{\mathrm{T}} + [\mathbf{G}^0]^{\mathrm{T}}\left[[\mathbf{H}^0]^{\mathrm{T}}[\mathbf{d}_{[\beta]}^{(n)}]^{\mathrm{T}} + [\mathbf{G}^0]^{\mathrm{T}}[\mathbf{d}_{[\gamma]}^{(n)}]^{\mathrm{T}}\right]^{\mathrm{T}} \\
\mathbf{s}_{[2]}^{(n+1)} &= [\mathbf{H}^0]^{\mathrm{T}}\left[[\mathbf{H}^1]^{\mathrm{T}}[\mathbf{s}^{(n)}]^{\mathrm{T}} + [\mathbf{G}^1]^{\mathrm{T}}[\mathbf{d}_{[\alpha]}^{(n)}]^{\mathrm{T}}\right]^{\mathrm{T}} + [\mathbf{G}^0]^{\mathrm{T}}\left[[\mathbf{H}^1]^{\mathrm{T}}[\mathbf{d}_{[\beta]}^{(n)}]^{\mathrm{T}} + [\mathbf{G}^1]^{\mathrm{T}}[\mathbf{d}_{[\gamma]}^{(n)}]^{\mathrm{T}}\right]^{\mathrm{T}} \\
\mathbf{s}_{[1]}^{(n+1)} &= [\mathbf{H}^1]^{\mathrm{T}}\left[[\mathbf{H}^0]^{\mathrm{T}}[\mathbf{s}^{(n)}]^{\mathrm{T}} + [\mathbf{G}^0]^{\mathrm{T}}[\mathbf{d}_{[\alpha]}^{(n)}]^{\mathrm{T}}\right]^{\mathrm{T}} + [\mathbf{G}^1]^{\mathrm{T}}\left[[\mathbf{H}^0]^{\mathrm{T}}[\mathbf{d}_{[\beta]}^{(n)}]^{\mathrm{T}} + [\mathbf{G}^0]^{\mathrm{T}}[\mathbf{d}_{[\gamma]}^{(n)}]^{\mathrm{T}}\right]^{\mathrm{T}} \\
\mathbf{s}_{[3]}^{(n+1)} &= [\mathbf{H}^1]^{\mathrm{T}}\left[[\mathbf{H}^1]^{\mathrm{T}}[\mathbf{s}^{(n)}]^{\mathrm{T}} + [\mathbf{G}^1]^{\mathrm{T}}[\mathbf{d}_{[\alpha]}^{(n)}]^{\mathrm{T}}\right]^{\mathrm{T}} + [\mathbf{G}^1]^{\mathrm{T}}\left[[\mathbf{H}^1]^{\mathrm{T}}[\mathbf{d}_{[\beta]}^{(n)}]^{\mathrm{T}} + [\mathbf{G}^1]^{\mathrm{T}}[\mathbf{d}_{[\gamma]}^{(n)}]^{\mathrm{T}}\right]^{\mathrm{T}}
\end{aligned}
\tag{27}
$$

A *decoding* process starting from the coarsest resolution level (0), via Eq. (27), successively sums the scale and detail coefficients at any sub-element on $g^{(n)}$ to produce the scale coefficients for each of its four child sub-elements on $g^{(n+1)}$. After decoding up to any resolution level (n), $0 \leq n \leq L$, a single-scale expansion at $g^{(n)}$ can be formed, as in Eq. (20).

2.2.2. Approach 2: 3-component multiwavelets from the slope-decoupled basis

Approach 2 follows the philosophy in Gerhard et al. (2015a) and Gerhard and Müller (2016). It relies on deriving a mother multiwavelet basis from a father basis $\hat{\mathbf{P}}$ that is initially truncated to first order monomials (i.e. similar to the basis of the scaled DG2 reference solver (Eq. (5)). Here, the multiwavelet basis in Gerhard et al. (2015a) and Gerhard and Müller (2016) is considered, but in combination with the slope-decoupled DG2 operators (Eqs. (13a)-(13c)) instead of the standard DG2 form on a conventional 2D stencil (see Kesserwani et al., 2018). Such a multiwavelet basis can be derived by generalising Alpert's construction principle (Alpert, 1993) to the 2D case based upon a Gram-Schmidt orthonormalisation procedure (details in Gerhard and Müller, 2016).

In keeping with the notation of Section 2.2.1, the size of the scale and detail coefficients in Approach 2 is one component less than in Approach 1, which are given the following vector structure:

$$
\mathbf{s}^{(n)} = \begin{bmatrix} s^{0,(n)} & s^{1x,(n)} & s^{1y,(n)} \end{bmatrix}^{\mathrm{T}} \tag{28}
$$

$$
\mathbf{d}_{[\Theta]}^{(n)} = \begin{bmatrix} d_{[\Theta]}^{0,(n)} & d_{[\Theta]}^{1x,(n)} & d_{[\Theta]}^{1y,(n)} \end{bmatrix}^{\mathrm{T}} \tag{29}
$$

In Approach 2, the scale coefficients and detail coefficients on any parent sub-element on $g^{(n)}$ can be encoded from the scale coefficients of its four child sub-elements on $g^{(n+1)}$ as:

$$
\begin{aligned}
\mathbf{s}^{(n)} &= \mathbf{HH}^0\,\mathbf{s}_{[0]}^{(n+1)} + \mathbf{HH}^1\,\mathbf{s}_{[2]}^{(n+1)} + \mathbf{HH}^2\,\mathbf{s}_{[1]}^{(n+1)} + \mathbf{HH}^3\,\mathbf{s}_{[3]}^{(n+1)} \\
\mathbf{d}_{[\alpha]}^{(n)} &= \mathbf{GA}^0\,\mathbf{s}_{[0]}^{(n+1)} + \mathbf{GA}^1\,\mathbf{s}_{[2]}^{(n+1)} + \mathbf{GA}^2\,\mathbf{s}_{[1]}^{(n+1)} + \mathbf{GA}^3\,\mathbf{s}_{[3]}^{(n+1)} \\
\mathbf{d}_{[\beta]}^{(n)} &= \mathbf{GB}^0\,\mathbf{s}_{[0]}^{(n+1)} + \mathbf{GB}^1\,\mathbf{s}_{[2]}^{(n+1)} + \mathbf{GB}^2\,\mathbf{s}_{[1]}^{(n+1)} + \mathbf{GB}^3\,\mathbf{s}_{[3]}^{(n+1)} \\
\mathbf{d}_{[\gamma]}^{(n)} &= \mathbf{GC}^0\,\mathbf{s}_{[0]}^{(n+1)} + \mathbf{GC}^1\,\mathbf{s}_{[2]}^{(n+1)} + \mathbf{GC}^2\,\mathbf{s}_{[1]}^{(n+1)} + \mathbf{GC}^3\,\mathbf{s}_{[3]}^{(n+1)}
\end{aligned}
\tag{30}
$$

where $\mathbf{HH}^{0,1,2,3}$ are 2D low-pass filters and $\mathbf{GA}^{0,1,2,3}$, $\mathbf{GB}^{0,1,2,3}$ and $\mathbf{GC}^{0,1,2,3}$ are high-pass filters. The explicit expressions of these filters, which can be directly implemented, are provided in the Appendix A (see Eqs. (A.6)-(A.9)).

In the opposite direction, to decode the scale coefficients at each of the four child sub-elements on $g^{(n+1)}$ from the scale and detail coefficients of their parent sub-element on $g^{(n)}$, the following relationship should be used:

$$\begin{aligned}
\mathbf{s}_{[0]}^{(n+1)} &= [\mathbf{HH}^0]^{\mathrm{T}}\,\mathbf{s}^{(n)} + [\mathbf{GA}^0]^{\mathrm{T}}\,\mathbf{d}_{[\alpha]}^{(n)} + [\mathbf{GB}^0]^{\mathrm{T}}\,\mathbf{d}_{[\beta]}^{(n)} + [\mathbf{GC}^0]^{\mathrm{T}}\,\mathbf{d}_{[\gamma]}^{(n)} \\
\mathbf{s}_{[2]}^{(n+1)} &= [\mathbf{HH}^1]^{\mathrm{T}}\,\mathbf{s}^{(n)} + [\mathbf{GA}^1]^{\mathrm{T}}\,\mathbf{d}_{[\alpha]}^{(n)} + [\mathbf{GB}^1]^{\mathrm{T}}\,\mathbf{d}_{[\beta]}^{(n)} + [\mathbf{GC}^1]^{\mathrm{T}}\,\mathbf{d}_{[\gamma]}^{(n)} \\
\mathbf{s}_{[1]}^{(n+1)} &= [\mathbf{HH}^2]^{\mathrm{T}}\,\mathbf{s}^{(n)} + [\mathbf{GA}^2]^{\mathrm{T}}\,\mathbf{d}_{[\alpha]}^{(n)} + [\mathbf{GB}^2]^{\mathrm{T}}\,\mathbf{d}_{[\beta]}^{(n)} + [\mathbf{GC}^2]^{\mathrm{T}}\,\mathbf{d}_{[\gamma]}^{(n)} \\
\mathbf{s}_{[3]}^{(n+1)} &= [\mathbf{HH}^3]^{\mathrm{T}}\,\mathbf{s}^{(n)} + [\mathbf{GA}^3]^{\mathrm{T}}\,\mathbf{d}_{[\alpha]}^{(n)} + [\mathbf{GB}^3]^{\mathrm{T}}\,\mathbf{d}_{[\beta]}^{(n)} + [\mathbf{GC}^3]^{\mathrm{T}}\,\mathbf{d}_{[\gamma]}^{(n)}
\end{aligned} \tag{31}$$

It is useful to note that Approach 2 entails more complicated 3×3 filter matrices, as compared to the 2×2 filter matrices involved in Approach 1 (see Appendix A).

2.2.3. Applicability of the analysis for the DG2 modes on multiresolution bases

The encoding (Eqs. (26) or (30)) and decoding (Eqs. (27) or (31)) formulae can be directly applied to the DG2 modes, $\boldsymbol{u}_e^{(n)}$, and their detail coefficients, $\mathbf{d}_{[\Theta]_e}^{(n)}$ (with $\Theta = \alpha, \beta, \gamma$), over the hierarchy $\left\{g_c^{(n)}\right\}_n$. It suffices to multiply the encoding formulae (Eqs. (26) or (30)), by a factor of $1/2$, and the decoding formulae (Eqs. (27) or (31)) by a factor of 2. This re-scaling results from repeating the analysis in Section 2.2 with $\boldsymbol{\phi}_e^{(n)}$, Eq. (17), instead of $\hat{\mathbf{P}}^{(n)}$. With Approach 2, $\boldsymbol{u}_e^{(n)}$ and its $\mathbf{d}_{[\Theta]_e}^{(n)}$ are vectors, consistent with Eqs. (28) and (29), that represent $\boldsymbol{u}_e^{(n)}$ in line with the vector structure introduced in Section 2.1.2. Here, the magnitude of the local details within $\mathbf{d}_{[\Theta]_e}^{(n)}$ will be referred to as $\left|d_e^{(n)}\right|$, calculated as:

$$\left|d_e^{(n)}\right| = max\left(\left|d_{[\Theta]_e}^{K,(n)}\right|\right)_{K=0,1x,1y\ \Theta=\alpha,\beta,\gamma} \tag{32}$$

With Approach 1, the spectral components of $\boldsymbol{u}_e^{(n)}$ and $\mathbf{d}_{[\Theta]_e}^{(n)}$ are represented with 2×2 matrices, consistent with Eqs. (24) and (25), while keeping a zero 4th component (i.e. in the "2, 2 entry" of matrix $\boldsymbol{u}_e^{(n)}$). Here, the magnitude of the local details within $\mathbf{d}_{[\Theta]_e}^{(n)}$ is calculated as:

$$\left|d_e^{(n)}\right| = max\left(\left|d_{[\Theta]_e}^{K,(n)}\right|\right)_{K=0,1x,1y,1xy\ \Theta=\alpha,\beta,\gamma} \tag{33}$$

With the re-scaled formulae, encoding of the DG2 modes across $\left\{g_c^{(n)}\right\}_n$ allows the formation of a *compressed MWDG2 solution*. This form provides access to the detail coefficients for analysis to decide the adaptive solution and grid. Whereas, decoding the remaining significant details allows the formation of an *assembled DG2 solution* on a non-uniform grid comprising (non-overlapping) sub-elements of different sizes. On this non-uniform grid, the DG2 modes can be directly updated using the scaled DG2 operators (recall Eqs. (13a)-(13c) and Section 2.1.2).

2.3. DG2 adaptive solution on a multiresolution grid

This section describes how the proposed approaches for multiresolution analysis (Section 2.2) can be combined with the scaled DG2 formulation (Section 2.1.2), in order to produce an adaptive DG2 solution on a multiresolution grid. Based on the maximum resolution level[1] L, the computational domain $\Omega = \cup_{c=1}^{M\times N} Q_c$ should be further subdivided such that each element Q_c has 2^{2L} sub-elements $\left\{Q_e^{(L)}\right\}_{e=0,\ldots,2^{2L}-1}$, i.e. $Q_c = g_c^{(L)} = \cup_{e=0}^{2^{2L}-1} Q_e^{(L)}$. This setting provides the finest uniform discretisation for Ω comprising $2^{2L} \times M \times N$ sub-elements. Without loss of generality, as the adaptive MWDG2 scheme is applicable element-wise, the coarsest grid spanning Ω is hereafter assumed to be a single parent element Q_c (i.e.

[1]For hydrodynamic modelling over a DEM, the maximum refinement level (L) is selected such that $g_c^{(L)}$ matches the resolution of the DEM.

$M = N = 1$ and $g_c^{(0)} = \Omega = Q_c$). Now, $g_c^{(L)}$ represents the finest uniform grid discretisation for Ω, comprising sub-elements $\left\{Q_e^{(L)}\right\}_{e=0,\ldots,2^{2L}-1}$. On each sub-element $Q_e^{(L)}$, DG2 modes, $\boldsymbol{u}_e^{(L)}$, can be initialised to form an assembled DG2 solution on $g_c^{(L)}$ for initial pre-processing (Section 2.3.1). The DG2 modes are initialised for all the physical variables, $u \in \{h, hu, hv, z\}$, using Eqs. (14a)-(14d) with the subscript e instead of c.

2.3.1. *Pre-processing: generation of initial detail coefficients ($t = 0$ s)*

From the initial DG2 modes of the flow variables and the topography on $g_c^{(L)}$, $\left[\boldsymbol{u}_e^{(L)}\right]_e$, the detail coefficients, $\left[\mathbf{d}_{[\Theta]_e}^{(n)}\right]_{n,e}$, at the lower resolution grids $\left\{g_c^{(n)}\right\}_{n=L-1,\ldots,1,0}$ can be encoded. This is achieved by successive application of either of the re-scaled Eqs. (26) and (30) in a descending order starting from resolution level $(L-1)$ until reaching the coarsest level (0) where both the coarsest DG2 modes, $\boldsymbol{u}_0^{(0)}$, and detail coefficients, $\mathbf{d}_{[\Theta]_0}^{(0)}$, become available. Note that encoding must be applied to the free-surface water elevation, $h + z$, instead of the water depth variable, h, to avoid producing any spurious disturbance to a flat water surface. Therefore, the set of encoded detail coefficients, $\left[\mathbf{d}_{[\Theta]_e}^{(n)}\right]_{n,e}$, is actually associated with the physical variables $u \in \{h + z, hu, hv, z\}$.

From the set of detail coefficients, $\left[\mathbf{d}_{[\Theta]_e}^{(n)}\right]_{n,e}$, a different set of normalised detail magnitudes, denoted by $\left[\check{d}_e^{(n)}\right]_{n,e}$, is also generated. This set is needed to measure the significance of all detail coefficient combined, regardless of which physical variable, u, they represent. Namely, a normalised detail magnitude, $\check{d}_e^{(n)}$, is a scalar that is evaluated from the local magnitude, $\left|d_e^{(n)}\right|$ (see Eqs. (32) and (33)), as:

$$\check{d}_e^{(n)} = \frac{\left|d_e^{(n)}\right|}{\max\left(1, \left|\left[u_e^{0,(L)}\right]_e\right|\right)} \tag{34}$$

where $\left[u_e^{0,(L)}\right]_e$ is the maximum of the average coefficients of the DG2 modes on $g_c^{(L)}$ across all variables $u \in \{h + z, hu, hv, z\}$ —– also across the hierarchy due to variational boundness across refinement levels.

At the starting time, $t = 0$ s, all the detail coefficients $\left[\mathbf{d}_{[\Theta]_e}^{(n)}\right]_{n,e}$ for all the variables $u \in \{h+z, hu, hv, z\}$ are fully accessible on $\left\{g_c^{(n)}\right\}_{n=L-1,\ldots,1,0}$. They can be ascendingly summed upon the coarsest DG2 modes $\boldsymbol{u}_0^{(0)}$ on Q_c to form a compressed MWDG2 solution on $\left\{g_c^{(n)}\right\}_{n=L-1,\ldots,1,0}$, which is as accurate as the assembled DG2 solution on $g_c^{(L)}$. Later, when $t > 0$ s, the detail coefficients of the flow variables $u \in \{h + z, hu, hv\}$ will be subjected to constant change as these components are time-dependent, whereas the detail coefficients of the topography, z, do not change with time.

2.3.2. *Prediction, regularisation and decoding: adaptive solution generation ($t \geq 0$ s)*

By analysing the magnitude of the normalised details, $\left[\check{d}_e^{(n)}\right]_{n,e}$, on $\left\{g_c^{(n)}\right\}_{n=L-1,\ldots,1,0}$ an adaptive grid at a present time t, denoted by $g_c^A(t)$, can be formed by selecting a set of (non-overlapping) sub-elements:

$$g_c^A(t) \subset \left\{Q_e^{(n)} \in \left\{g_c^{(n)}\right\}_n, \; 0 \leq n \leq L, \; 0 \leq e \leq 2^{2L} - 1 \quad \text{and} \quad \Omega = \cup_{n,e} Q_e^{(n)}\right\} \tag{35}$$

To select the sub-element $Q_e^{(n)}$ forming $g_c^A(t)$, *prediction* is applied, which is the act of measuring the normalised detail magnitudes. The prediction procedure involves four subsequent steps that are summarised as follows.

First, a parameter ε needs to be prescribed such that $0 < \varepsilon < 1$. This parameter is user-chosen and is, in fact, the *error threshold* below which detail coefficients can be ignored. Although there is no unique choice for ε, a range of choices exists to keep the assembled DG2 solution on $g_c^A(t)$ as accurate as the assembled solution on $g_c^{(L)}(t)$. This optimal range for ε is expected to be somewhere between 10^{-4} and 10^{-2} within the scope of modelling shallow water flows (Kesserwani et al., 2019; Caviedes-Voullième et al., 2020), but is rather context-specific (Sharifian et al., 2019). An analysis of the choice for ε with the proposed adaptive HFV1 and MWDG2 schemes is carried out later in Section 3.1.1.

Second, the normalised detail magnitudes, $\left[\breve{d}_e^{(n)}\right]_{n,e}$ are compared to ε for identifying the significant details. In doing so, the set of normalised detail magnitudes is scanned level-wise, in ascending order $n = 0, 1, \ldots, L-1$, while comparing the magnitudes to level-dependent error thresholds $\varepsilon^{(n)} = \varepsilon 2^{n-L}$. A detail coefficient $\mathbf{d}_{[\Theta]_e}^{(n)}$ on $Q_e^{(n)}$ is classified as *significant* if:

$$\breve{d}_e^{(n)} > \varepsilon^{(n)} \tag{36}$$

Meanwhile, sub-elements $Q_e^{(n)}$ with significant details are flagged as *active*, meaning that they are plausible candidates for inclusion in $g_c^A(t)$.

Third, re-flagging of active sub-elements is needed for *regularisation*, which ensures that significant details can be accessed within a tree structure: when any of the child details $\mathbf{d}_{[\Theta]_e}^{(n)}$ on $g_c^{(n)}$ are significant, the parent details $\mathbf{d}_{[\Theta]_e}^{(n-1)}$ on $g_c^{(n-1)}$ must also be significant. Regularisation is therefore the act of ensuring that such sub-elements, $Q_e^{(n-1)}$, are also flagged as active (to enable access to $\mathbf{d}_{[\Theta]_e}^{(n-1)}$ during decoding when generating an assembled DG2 solution on $g_c^A(t)$). Note that, for generality, when the coarsest grid spanning Ω comprises more than one parent element ($M > 1$ or $N > 1$), regularisation should also activate sub-elements located at the boundaries across the parent elements to ensure that water waves can propagate across different parent elements.

Fourth, all significant detail coefficients, $\mathbf{d}_{[\Theta]_e}^{(n)}$, at a present time t, are revisited to predict whether their significance is likely to remain or increase at time $t + \Delta t$, with Δt denoting the time-step restricted by the CFL condition with a Courant number, Cr, not exceeding 0.3 (Cockburn and Shu, 2001). Such a detail is referred to as *extra-significant* and can be identified by:

$$\breve{d}_e^{(n)} \geq 2^{\bar{m}+1} \varepsilon^{(n)} \tag{37}$$

where $\bar{m}$ is the order-of-accuracy of the prediction operator, chosen to be 1.5 (Kesserwani et al., 2019). When a detail coefficient $\mathbf{d}_{[\Theta]_e}^{(n)}$ is extra-significant, the set of active sub-elements is enlarged to include, in addition to $Q_e^{(n)}$, its four child sub-elements $Q_{[m]_e}^{(n+1)}$ with $m = 0, 1, 2, 3$ (Fig. 2). This step is necessary to ensure that the significant features of the assembled DG2 solution on $g_c^A(t)$ will also be preserved on $g_c^A(t + \Delta t)$ when generating the future detail coefficients (Section 2.3.4).

Finally, a DG2 solution on $g_c^A(t)$ can be decided and assembled by inspecting the tree of details in ascending order. The process starts from the coarsest detail coefficients $\mathbf{d}_{[\Theta]_0}^{(0)}$ and DG2 modes $\boldsymbol{u}_0^{(0)}$, while decoding (using either of the re-scaled Eqs. (27) and (31)). Inspection of details is aborted under two circumstances:

(i) When a detail coefficient $\mathbf{d}_{[\Theta]_e}^{(n)}$ switches status to becoming insignificant for the first time, with its DG2 modes $\boldsymbol{u}_e^{(n)}$ already decoded for inclusion in the generation of the assembled DG2 solution on $g_c^A(t)$, or

(ii) When inspection and decoding reached $g_c^{(L-1)}$ with certain detail coefficients $\mathbf{d}_{[\Theta]_e}^{(L-1)}$ remaining significant, and their DG2 modes $\boldsymbol{u}_e^{(L-1)}$ already decoded. Then, a final round of decoding is applied to generate the DG2 modes, $\boldsymbol{u}_e^{(L)}$, at the four child sub-elements on $g_c^{(L)}$ while generating the assembled DG2 solution on $g_c^A(t)$.

The adaptive DG2 solution is a series of carefully-selected DG2 modes forming an assembled DG2 solution on the non-uniform grid $g_c^A(t)$. These DG2 modes should then be updated by applying the scaled DG2 reference solver (Section

2.1), considering the main flow vector variables $u \in \{h, hu, hv\}$. Hence, the DG2 modes representing the water depth, h, should be restored, by subtracting the modes representing the topography, z, from those of the free-surface elevation, $h + z$.

2.3.3. *Update of the DG2 modes on $g_c^A(t)$*

To update the DG2 modes $\boldsymbol{u}_e^{(n)}$ on $g_c^A(t)$, the scaled DG2 operators (Eqs. (13a)-(13c)) are applied alongside specific treatments to ensure: (i) stability around shock-like solution gradients via double localisation and limiting; (ii) conservative evaluation of the input arguments of the HLL Riemann solver at the faces shared by sub-elements of different sizes; and, (iii) well-balanced and depth-positivity preserving DG2 modes for a discontinuous topography, z_h, on $g_c^A(t)$. The specifics of these treatments are explained next.

Double localisation and limiting: This treatment is applied before evaluating the scaled DG2 operators to ensure the stability of the assembled DG2 solution around shock-like gradients. It is aimed at restricting the operation of the slope limiter to the portions of the assembled DG2 solution where very sharp discontinuities are about to occur. The first localisation step is to flag only the active slope coefficients on $g_c^{(L)}$ for possible limiting, i.e. $u_e^{1x,(L)}$ and $u_e^{1y,(L)}$ with $u \in \{h, hu, hv\}$. These slope coefficients can only be active at refinement level (L) when sustained by a tree of significant details (Section 2.3.2). This means that at least one of these slope coefficients is attributed to a local flow feature, whether or not it is a very sharp discontinuity, i.e. a shock wave, or any other form of solution kinks, e.g. head or toe of a depression wave or a wet-dry front. The second localisation step is applied to remove the need for slope limiting around any other form of solution kinks. This can be achieved by further subjecting the slope coefficients $u_e^{1x,(L)}$ and $u_e^{1y,(L)}$ to a local shock detector (Krivodonova et al., 2004). That is, to detect if $u_e^{1x,(L)}$ needs slope limiting, the normalised magnitude of the solution discontinuity across the eastern face-centred node of $Q_e^{(L)}$, $\mathrm{DS}_{e_{\mathrm{E}}}$, first needs to be calculated as:

$$\mathrm{DS}_{e_{\mathrm{E}}} = \frac{\left|u_e^{+,(L)} - u_e^{-,(L)}\right|}{\frac{\Delta x^{(L)}}{2} \max\left(\left|u_{e_{\mathrm{Gx1}}}^{(L)}\right|, \left|u_{e_{\mathrm{Gx2}}}^{(L)}\right|\right)} \tag{38}$$

where $u_e^{-,(L)} = u_e^{0,(L)} + \sqrt{3} u_e^{1x,(L)}$ and $u_e^{+,(L)} = u_{nei_{\mathrm{E}}}^{0,(L)} - \sqrt{3} u_{nei_{\mathrm{E}}}^{1x,(L)}$ represent the limits of u_h from the side of sub-element $Q_e^{(L)}$ and its eastern neighbour $Q_{nei_{\mathrm{E}}}^{(L)}$, respectively, and $u_{e_{\mathrm{Gx1}}}^{(L)} = u_e^{0,(L)} - u_e^{1x,(L)}$ and $u_{e_{\mathrm{Gx2}}}^{(L)} = u_e^{0,(L)} + u_e^{1x,(L)}$ are local solution evaluations at the Gaussian points located on the x-directional centreline (see Fig. 1). Similarly, the solution discontinuity across the western face-centred node of $Q_e^{(L)}$, $\mathrm{DS}_{e_{\mathrm{W}}}$, can be calculated. When $\min\left(\mathrm{DS}_{e_{\mathrm{E}}}, \mathrm{DS}_{e_{\mathrm{W}}}\right) > 10$ or any higher threshold, at least one strong inter-elemental discontinuity is present in the x-direction. Hence, the slope coefficient $u_e^{1x,(L)}$ is limited, via the generalised minmod limiter (Cockburn and Shu, 2001), to produce a limited slope coefficient, $\hat{u}_e^{1x,(L)}$, as:

$$\hat{u}_e^{1x,(L)} = \mathrm{minmod}\left(u_e^{1x,(L)}, u_{nei_{\mathrm{E}}}^{0,(L)} - u_e^{0,(L)}, u_e^{0,(L)} - u_{nei_{\mathrm{W}}}^{0,(L)}\right) \tag{39}$$

Note that the process of shock detection and limiting for the slope component of the water depth variable, $h_e^{1x,(L)}$, is avoided. Instead, the process is applied for the free-surface elevation $h_e^{1x,(L)} + z_e^{1x,(L)}$ to first get a limited slope coefficient for the free-surface water elevation, and then deduce $h_e^{1x,(L)}$. This will ensure that the presence of terrain steps will not falsely trigger slope limiting. In a similar way, shock detection and limiting is applied to potentially limit $u_e^{1y,(L)}$.

Conservative evaluation at the faces shared by dissimilar sub-elements: This treatment is particularly important when coping with a multiresolution grid $g_c^A(t)$ that includes faces separating sub-elements of different sizes. The use of multiresolution analysis makes it possible to readily overcome at least two major limitations of classical adaptive mesh refinement methods (e.g. Li, 2010; Liang, 2012; Kesserwani and Liang, 2012a; Zhou et al., 2013; Liang et al., 2015). Firstly, it unconditionally allows any gap in resolution level across two adjacent sub-elements (Caviedes-Voullième et al., 2020), as opposed to standard quadtree approaches that are constrained by the 2:1 rule (Liang, 2012). Secondly, the availability of the encoding and decoding formulae (i.e. Eqs. (26) and (30) or (27) and (31)) negates the need to aggregate Riemann fluxes at higher resolution sub-elements, sharing a face with lower resolution sub-elements (Kesserwani and Liang, 2012a).

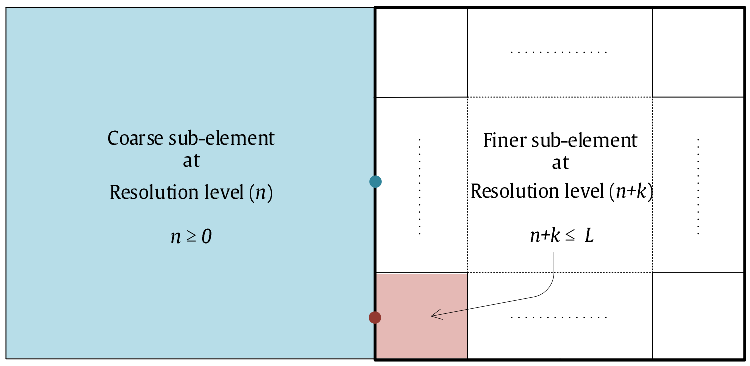

Figure 3. Schematic layout of two adjacent sub-elements: a coarse sub-element (in blue) at arbitrary resolution level (n) with $n \geq 0$ (left), a fine sub-element (in red) at resolution level ($n + k$), with $k \geq 1$ and $n + k \leq L$ (right). The red and blue points denote the center of faces belonging to fine and coarse sub-elements, respectively. At these points the limits of the local planar solution are evaluated to form the input arguments for the HLL Riemann solver subject to applying treatment (iii) (See Section 2.3.3). The thick border denotes the locally coarsened adjacent sub-element.

Without loss of generality, this treatment is explained for an elementary portion of the assembled DG2 solution on $g_c^A(t)$, which is made of a coarse sub-element, at resolution level (n), shaded in blue in Fig. 3, that is adjacent to a fine sub-element, at any higher resolution level ($n + k$), shaded in red in Fig. 3. When updating the DG2 modes on a fine sub-element, both limits of the DG2 solution at the face-centered node of the fine sub-element are directly obtainable, i.e. at the red point in Fig. 3, by using Eq. (10). For the coarse sub-element, limits at a matching resolution level (n) need to be produced, i.e. at the blue point in Fig. 3 from the side of the fine sub-element. To do so, the whole block of fine sub-elements, neighboring the coarse sub-element, is coarsened via k rounds of encoding (i.e. Eqs. (26) or (30)).

Well-balanced and depth-positivity preserving DG2 modes: With MWDG2, the approximate topography, z_h, becomes discontinuous as it gets assembled on $g_c^A(t)$ after decoding (because the coefficients used for encoding and decoding are derived from essentially discontinuous functions). Now, it is no longer possible to ensure the continuity of z_h at the face-centred nodes as was previously the case with the scaled DG2 reference solver (recall Section 2.1). This treatment has therefore been aimed to amend the wetting and drying condition of the DG2 refernce solver to accommodate potentially discontinuous topography. This has been done by involving the free-surface elevation, $h + z$, as an intermediate variable (Kesserwani and Liang, 2012b). For instance, now at the eastern face-centred node, E, the revised single topography limit is defined from the outset as (recall Section 2.1): $z_{\mathrm{E}}^{*} = \max\left(z_e^{-}, z_e^{+}\right)$, with $z_e^{-} = z_e^{0,(n)} + \sqrt{3} z_e^{1x,(n)}$ and $z_e^{+} = z_{nei_{\mathrm{E}}}^{0,(n)} - \sqrt{3} z_{nei_{\mathrm{E}}}^{1x,(n)}$. Along with this change, the *revised* limits for the water depth variable are reconstructed as: $h_{\mathrm{E}}^{+,*} = \max\left(0, \left(h_{nei_{\mathrm{E}}}^{+} + z_{nei_{\mathrm{E}}}^{+}\right) - z_{\mathrm{E}}^{*}\right)$ and $h_{nei_{\mathrm{E}}}^{-,*} = \max\left(0, \left(h_e^{-} + z_e^{-}\right) - z_{\mathrm{E}}^{*}\right)$, and the final revision of the topography limit reads $z_{\mathrm{E}}^{*} = z_{\mathrm{E}}^{*} - \max\left(0, -\left(\left(h_e^{-} + z_e^{-}\right) - z_{\mathrm{E}}^{*}\right)\right)$.

2.3.4. Truncation and encoding: forming a new compressed MWDG2 solution

This step aims to create a new set of detail coefficients from the updated DG2 modes to form a compressed MWDG2 solution on $\left\{g_c^{(n)}(t+\Delta t)\right\}_{n=L-1,\ldots,1,0}$. Now, the DG2 flow modes for the variables $u \in \{h, hu, hv\}$ are only defined for the sub-elements in $\left\{g_c^{(n)}(t)\right\}_{n=L-1,\ldots,1,0}$ spanning $g_c^{(A)}(t)$. The other sub-elements are inactive and have non-existent DG2 flow modes. *Truncation* is the process of initialising zero detail coefficients on $\left\{g_c^{(n)}(t+\Delta t)\right\}_{n=L-1,\ldots,1,0}$ to fill the inactive sub-elements and thereby keep them subject to potential activation in the next round (when re-applying the process described in Section 2.3.2). Encoding is then applied over the active sub-elements, by successively applying Eq. (26) or (30), level-wise in descending order. This generates new detail coefficients for the flow variables on $\left\{g_c^{(n)}(t+\Delta t)\right\}_{n=L-1,\ldots,1,0}$ and thereby addresses any irrelevant zeroing introduced previously by truncation. Note that encoding should be applied on the variables $u \in \{h+z, hu, hv\}$ as done in the pre-processing step (Section 2.3.1). After truncation and encoding, a full set of new details $\left[\mathbf{d}_{[\Theta]_e}^{(n)}\right]_{n,e}$ is available, from which an alternative set of normalised details $\left[\check{d}_e^{(n)}\right]_{n,e}$ can be produced (recall Section 2.3.1). With the new sets of detail coefficients in place, the process (Sections 2.3.2–2.3.4) can be repeated to evolve the adaptive DG2 solution up to a specific simulation time.

2.4. First-order variant: 2D Haar Finite Volume (HFV1) solver

An adaptive HFV1 solver can be formed by reducing the complexity of the adaptive MWDG2 solver. It suffices to neglect the slope components in the Legendre basis functions (i.e. only consider $P^0(\xi,\eta)$ in Eq. (3)). This leads to a piecewise-constant local solution $\boldsymbol{u}_e^{(n)}$ over any sub-element Q_e that can be updated by the $\mathbf{L}^0$ operator in Eq. (10) within an explicit Euler time integration. The time step is selected according to a Cr number not exceeding 0.5 to preserve stability under wetting and drying. From $P^0(\xi,\eta)$, the same the low- and high-pass filters derived for 1D case (i.e. Eq. (55) in Kesserwani et al., 2019) are usable for the 2D case, to achieve encoding and decoding formulae (i.e. Eq. (48) and Eq. (49) in Kesserwani et al., 2019) re-scaled by a factor of 2.

3. Numerical tests

The adaptive HFV1 and MWDG2 solvers are applied to reproduce selected hydrodynamic test cases with a focus on identifying their potential for 2D hydrodynamic modelling. In Section 3.1, the adaptive solvers are compared for three diagnostic test cases to explore the performance of Approach 1 *vs* Approach 2 with the MWDG2 solver (Sections 2.2.1 and 2.2.2). Comparisons include analysis of accuracy, efficiency and robustness for shallow flows over irregular terrain with wet-dry zones and fronts. In Section 3.2, three laboratory-scale hydrodynamic tests are investigated to study the potential of the adaptive solvers when simulating more realistic features relevant to practical hydrodynamic modelling. Simulations are performed under maximum Cr number considerations, 0.5 with HFV1/FV1 and 0.3 with MWDG2/DG2, and on the same desktop computer. In the rest of Section 3, the term *element* will also be used as shorthand to a sub-element to ease readability.

3.1. Comparisons and verifications

Academic test cases are employed to verify the implementation of the adaptive MWDG2 and HFV1 solvers and to explore their accuracy, efficiency and robustness properties. The classic 1D dam-break test is used to investigate adaptivity-related issues and choices, and to compare the performance of MWDG2 with Approach 1 *vs* with Approach 2. The classic 2D circular dam-break test is then investigated to particularly analyse the 2D grid prediction capability of the adaptive MWDG2 solver with either of the two approaches. The third test embraces a 2D quiescent flow over smooth and discontinuous topography blocks with wet and dry zones to verify well-balancedness of the adaptive MWDG2 solvers despite the presence of wet-dry zones and fronts positioned over steep bed-slopes.

3.1.1. 1D dam-break flow on a wet domain with shock

This test was employed by Kesserwani et al. (2019) to analyse choices for ε with the 1D adaptive MWDG2 and HFV1 solvers, suggesting $\varepsilon = 10^{-3}$ can make the adaptive solvers preserve: (i) the numerical accuracy of their uniform FV1 and DG2 counterparts on grids at the finest resolution accessible to the adaptive solvers, and (ii) a computational efficiency that is superior to the uniform FV1 solver. This test is therefore re-employed under a 2D configuration to study the same aspects with the proposed 2D version of the adaptive MWDG2 and HFV1 solvers.

A $50\,\mathrm{m} \times 25\,\mathrm{m}$ domain is assumed with an imaginary dam placed in the middle of the longer dimension, separating initial water depths of 6 m and 2 m upstream and downstream of the dam, respectively. After the dam removal, a shock wave and a rarefaction wave propagate, separated by a constant state. Simulations are run considering the adaptive MWDG2 solvers with Approach 1 and Approach 2, the adaptive HFV1 solver, and the uniform DG2 and FV1 counterparts. For the adaptive solvers, the baseline grid is composed of $M \times N = 2 \times 1$ elements and the highest resolution level is $L = 8$, resulting in a grid allowing up to $512 \times 256 = 131,072$ elements. The grid for the FV1 and DG2 solvers comprised $128 \times 64 = 8,192$ elements on a two-level coarser uniform grid at $L = 6$, and $512 \times 256 = 131,072$ elements on the finest grid at $L = 8$. The different aspects initially investigated in Kesserwani et al. (2019) are re-investigated here.

Optimal choice for the error threshold driving (multi)wavelet-adaptivity: Adaptive solver simulations are performed for different ε values considering an order-of-magnitude between 10^{-5} to 10^{-1}, to identify choices that simultaneously meet (i) and (ii). As in Kesserwani et al. (2019), accuracy analysis is performed in terms of normalised ℓ^2 error difference between the numerical water depth and the analytical water depth at $t = 2.5$ s, when both shock and rarefaction waves are still present in the computational domain, whereas computational efficiency is measured in terms of CPU runtime at $t = 40$ s.

Fig. 4a shows the errors generated by the uniform and adaptive solvers. At $\varepsilon = 10^{-1}$, the errors of the adaptive solvers exceed those of the uniform counterparts at a resolution that is is two-level coarser (i.e. the grid with 8,192 elements). The error of MWDG2 with Approach 2 exceeds that of the lower-order HFV1 solver, indicating that $10^{-1} \leq \varepsilon < 1$ is not an appropriate range. Note that the errors of the 2D solvers are slightly bigger compared to those of their 1D counterparts in Kesserwani et al. (2019). This slight increase in errors could be caused by the fact that the 2D case involves much more truncation operations on the filters' coefficients. At $\varepsilon = 10^{-2}$, the errors of the adaptive solvers start to converge towards that of their uniform counterparts, and MWDG2 with Approach 1 shows faster convergence. At $\varepsilon = 10^{-3}$, the errors of the adaptive solvers have almost reached the intended accuracy: HFV1's error is almost at the level of the uniform FV1's error and the error of MWDG2 with Approach 1 is the closest to the uniform DG2's error. At $\varepsilon \leq 10^{-3}$, the target accuracy is reached for the adaptive solvers as they all have their errors close to errors of their respective uniform solver counterparts.

Fig. 4b compares the CPU times required by the adaptive and uniform solvers to complete a 40 s simulation with

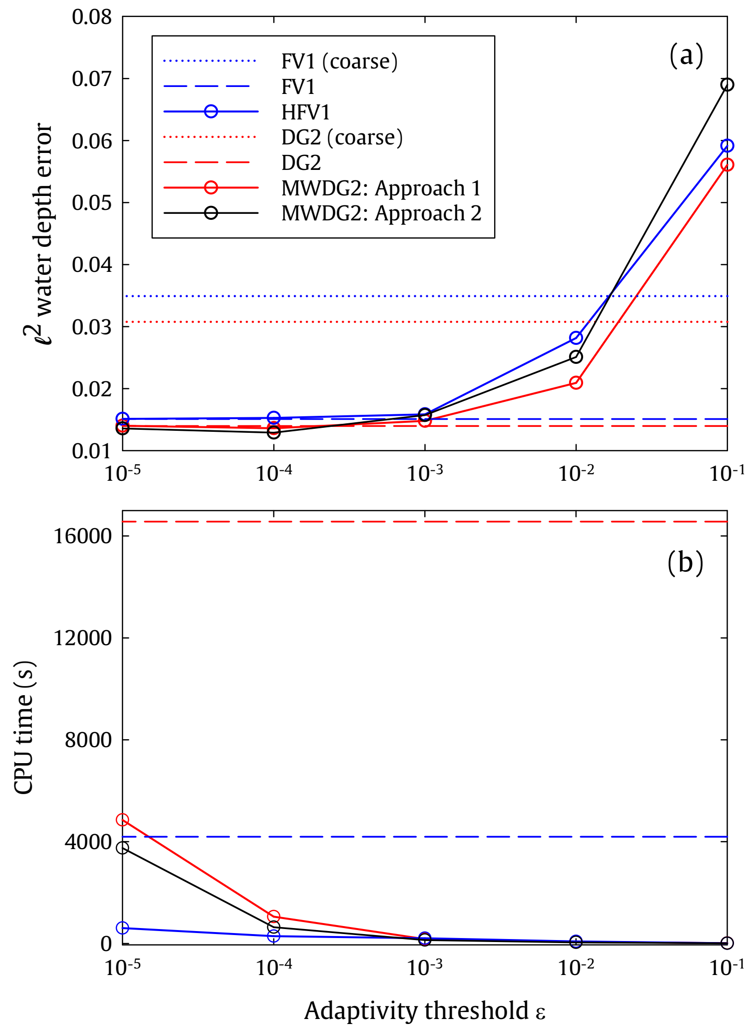

Figure 4. 1D dam-break flow on a wet-domain with shock.

Variation of (a) normalised ℓ^2 water depth error at $t = 2.5$ s and (b) total CPU time required to finish 40 s simulation, for different threshold values ranging between $\varepsilon = 10^{-5}$ and $\varepsilon = 10^{-1}$. Results of HFV1 and MWDG2 (with Approach 1 and Approach 2) are compared to those of FV1 and DG2 on uniform grids with $512 \times 256 = 131,072$ (shown as dashed lines) and $128 \times 64 = 8,192$ elements (shown as dotted lines).

the same range of threshold values $10^{-5} \leq \varepsilon \leq 10^{-1}$. As ε decreases, the CPU time of the adaptive solvers increases (as expected). More strikingly, for almost all ε values, the CPU times of the adaptive solvers do not exceed the CPU time of the uniform FV1 solver, with the exception of MWDG2 with Approach 1 at $\varepsilon = 10^{-5}$, which slightly surpasses it. MWDG2 with Approach 1 is found to be slightly more expensive to run than MWDG2 with Approach 2, which is expected as the former involves an extra detail coefficient during encoding and decoding (Section 2.2.1). Note that the 1D version of these same adaptive solvers dictated $\varepsilon \geq 10^{-3}$ to reach the same efficiency target (Kesserwani et al., 2019), hence suggesting that HFV1/MWDG2 gain better efficiency in the 2D case.

To maximise accuracy and efficiency targets, an error threshold value in the proximity of $\varepsilon = 10^{-3}$ is recommended. In the remaining parts of Section 3.1 all simulations are performed using $\varepsilon = 10^{-3}$. In Section 3.2, in addition to considering $\varepsilon = 10^{-3}$, simulations will also be run considering an upper or a lower limit ε value, i.e. 10^{-2} or 10^{-4}, with preference to the lower limit when a test case involves a DEM at a resolution < 1 m.

Adaptive solution predictability of relevant flow features: The grid prediction ability of the adaptive solvers is analysed by comparing the adaptive grids resulting from the MWDG2 and HFV1 solvers at $t = 2.5$ s. Fig. 5a shows the grid of the adaptive MWDG2 solvers at the initial time $t = 0$ s, which indicates that both Approach 1 and Approach 2 refine the grid similarly: only in the vicinity of the dam where there is an abrupt water depth state. Fig. 5b-d show the adaptive grids at $t = 2.5$ s resulting from HFV1 and MWDG2 with Approach 1 and Approach 2, respectively. Around the shock wave, all the adaptive solvers predict the finest level of resolution. In contrast, much coarser resolution levels are selected along the constant state: MWDG2 with Approach 1 and Approach 2 predict almost similar coarsest resolution level, while that of HFV1 is one level-of-resolution higher as expected, given its lower order-of-accuracy. However, refining at shock or coarsening along a constant state is quite easy for any adaptive grid refinement method; rather, the challenge is to sensibly capture the variability in resolution levels when representing a gradual flow or topographic profile, such as a rarefaction wave (Kesserwani et al., 2019).

In the prediction of the rarefaction wave, substantial differences are observed between the grids predicted by the adaptive HFV1 and MWDG2 solvers. The HFV1 solver inefficiently refines its grid throughout the rarefaction wave by selecting the highest resolution level. MWDG2 with Approach 1 only selects the highest resolution level at the top and toe of the rarefaction wave, while coarsening in the middle part by at least one resolution level. MWDG2 with Approach 2 delivers the same coarsening in the middle part but shows wider extent of refinement around the rarefaction's top and no refinement at all along its toe. These results indicate that Approach 1 is likely more suitable to devise an adaptive MWDG2 solver for hydraulic modelling, where there is a need to sensibly capture appropriate levels of resolution variability in track with transitional flows.

Size of the coarsest baseline grid *vs* maximum refinement level: Conventional adaptive mesh refinement methods are reported to require a baseline grid that is fine enough to reliably trigger refinement at key initial flow features (Kesserwani and Liang, 2012a; George, 2011; Popinet, 2011; Acuña and Aoki, 2018). In contrast, wavelet-based adaptive solvers can unconditionally and reliably operate down to a baseline grid with one or two elements, as demonstrated for the 1D case (Caviedes-Voullième and Kesserwani, 2015; Kesserwani et al., 2019). To further validate this aspect in the 2D case, the adaptive solvers are run by varying the setting of the baseline grid size $M \times N$ *vs* maximum resolution level L: introducing a doubling in $M \times N$ while systematically reducing L such that to preserve the same maximum allowable number of elements, i.e. $512 \times 256 = 131,072$. Hence, $\{M, N, L\}$ varied as $\{2, 1, 8\}$, $\{4, 2, 7\}$, $\{8, 4, 6\}$, $\{16, 8, 5\}$, $\{32, 16, 4\}$, $\{64, 32, 3\}$, $\{128, 64, 2\}$ and $\{256, 128, 1\}$. The same accuracy and efficiency analyses are performed for the adaptive HFV1 and MWDG2 solvers with $\varepsilon = 10^{-3}$: outputs at $t = 2.5$ s to compare water depth errors and outputs at $t = 40$ s to compare CPU times. In terms of accuracy, the adaptive solvers with all the settings retrieved the same normalised ℓ^2 water depth errors and the same qualitative results as those shown in Fig. 5. This implies that coarsening the size of the baseline grid with the (multi)wavelet-based adaptive solvers does not have consequential effects on accuracy.

In terms of efficiency, Fig. 6 shows the CPU time costs required by the adaptive solvers for all the settings. Up to the

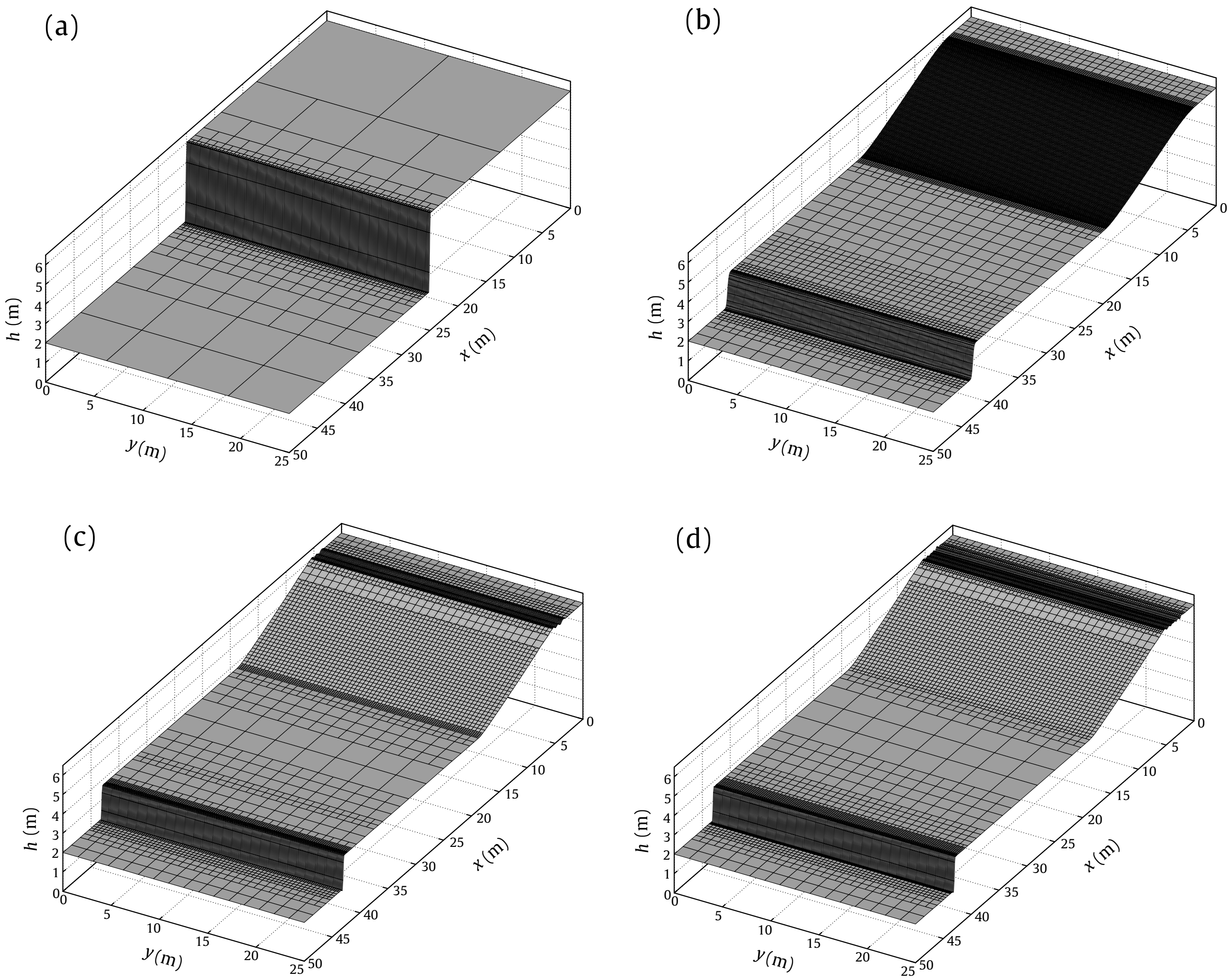

Figure 5. 1D dam-break flow on a wet-domain with shock.

The profiles of the 1D dam-break flow resulted from (a) the MWDG2 solvers based on Approach 1 and Approach 2 at $t = 0$ s, (b) the HFV1 solver at $t = 2.5$ s, (c) the MWDG2 solver based on Approach 1 at $t = 2.5$ s and (d) the MWDG2 solver based on Approach 2 at $t = 2.5$ s.

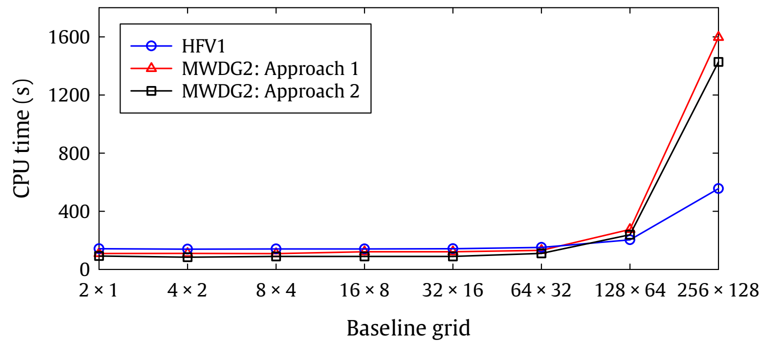

Figure 6. 1D dam-break flow on a wet-domain with shock.

CPU times required by the HFV1 and MWDG2 solvers based on Approach 1 and Approach 2 to complete 40 s long simulation for different settings with the baseline grid of $M \times N$ elements and the maximum resolution level of L. The baseline grid and the maximum resolution level are changed together in a way that the resulting adaptive grid is comprised of the same maximum of elements equal to $512 \times 256 = 131,072$.

baseline grid of size 64×32, the computational cost of all adaptive solvers remains almost the same, showing also that the adaptive MWDG2 solvers are less costly to run than the adaptive HFV1 solver. On finer baseline grids, the CPU times of the MWDG2 solvers start to significantly increase, becoming much bigger than the CPU time of the adaptive HFV1 solver dominated by the cost of the DG2 operators. MWDG2 with Approach 1 required slightly higher CPU runtimes than MWDG2 with Approach 2, implying that Approach 1 has slightly higher overhead costs to achieve MW adaptivity. For this test, choosing a baseline grid with a number of elements not exceeding 64×32 may be necessary to keep the cost of the adaptive MWDG2 solvers unhindered by the expenses of the DG2 solver, and thus make them more efficient than the adaptive HFV1 solver.

Grid coarsening ability and time-step size: The efficiency of the adaptive solvers is further explored by analysing the histories of element counts and time-step sizes over the 40 s simulation. Fig. 7a shows the number of elements consumed by MWDG2 with Approach 1, MWDG2 with Approach 2 and HFV1. At $t \leq 10$ s, the dam-break flow is still energetic in the domain due to flow depression at high velocity, initially stimulated by the shock wave ($t \leq 3.5$ s) and the rarefaction wave. During this period, HFV1 required more elements than both MWDG2 solvers, i.e. at least 3 times more after the shock left the domain during $3.5\,\text{s} \leq t \leq 10$ s, which is expected (recall Fig. 5). After $t = 10$ s, the flow continues to exit the domain at a much lower speed to become almost flood-like. During this period, the number of elements consumed by the adaptive solvers significantly decreases, as illustrated in the zoom-in portion within Fig. 7a. Now, MWDG2 with Approach 1 consumes far fewer elements compared to MWDG2 with Approach 2 and HFV1. The number of elements required by MWDG2 with Approach 2 fluctuated to be on average almost close to the number of elements consumed by HFV1. This finding points to recommending Approach 1 to devise an adaptive MWDG2 simulator for handling flows that are relatively slow and admit gradual transitions, like flooding. Favoring Approach 1 can also be supported by further analysing the recorded time-step size (Fig. 7b), in particular when $t \geq 10$ s: MWDG2 with Approach 1 allowed much larger time-steps than MWDG2 with Approach 2 and HFV1 until around $t = 25$ s, where the time-step of MWDG2 with Approach 2 remained even lower than that of HFV1.

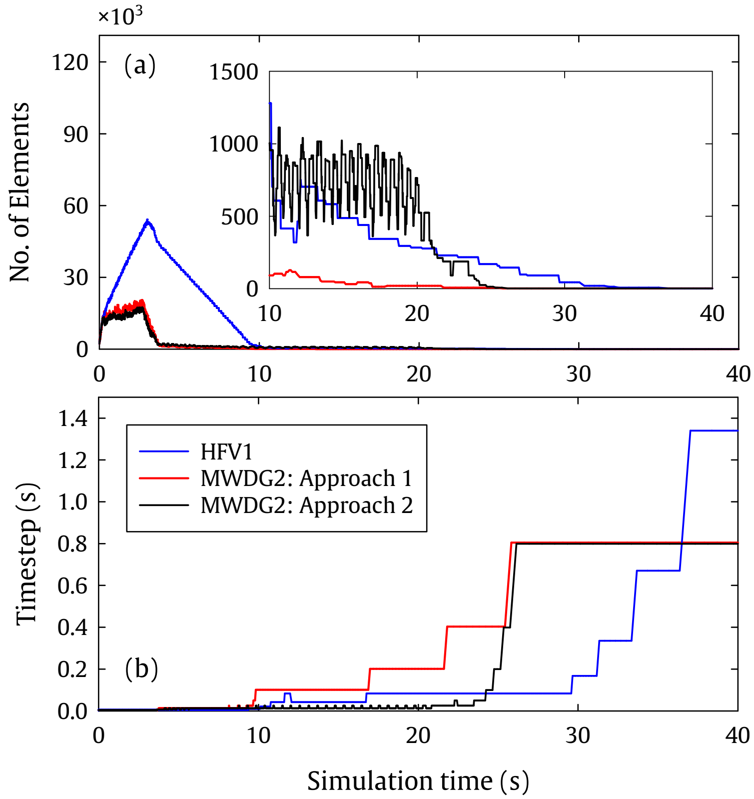

Figure 7. 1D dam-break flow on a wet-domain with shock.

Variation of (a) number of elements and (b) time-step size used by the MWDG2 solver based on Approach 1 and Approach 2 and the HFV1 solver, over the 40 s long simulation. The zoomed plot shows the final 30 s of the simulation when the shock and rarefaction waves have left the domain.

Overall, Fig. 7a and 7b indicate that both Approach 1 and Approach 2 are equally economical on resolution demands when applied to simulate highly dynamic flow cases, for which HFV1 would be inefficiently over-refining. MWDG2 with Approach 1 is found to be the most economical option to more efficiently simulate slow-to-gradual shallow water flows with fewer elements and bigger time-steps. For extremely slow flows, HFV1 is found to coarsen resolution as good as MWDG2 but with a bigger time-step, which suggests that it could still be the most efficient choice for such types of hydraulic simulations.

(Multi)wavelet-adaptivity overhead costs and analysis of CPU runtimes: To study the computational overhead cost due to (multi)wavelet adaptivity, the cumulative CPU times of HFV1 and MWDG2 with Approach 1 and Approach 2 are calculated and compared in Fig. 8. The figure also contains the runtimes of the uniform FV1 and DG2 solvers to accordingly meaure relative efficiency speed-ups for the 2D case. The runtimes of all the adaptive solvers remained lower than the runtime of the FV1 solver (thus also lower than the runtime of the uniform DG2 solver). This indicates that overhead costs due to (multi)wavelet adaptivity is unlikely to make the adaptive solvers more expensive than the uniform FV1 solver for the 2D case. This is in contrast to what the cumulative CPU times identified for the 1D case over this same test case, where HFV1 became more costly to run than the uniform FV1 at certain times even at $\varepsilon = 10^{-3}$.

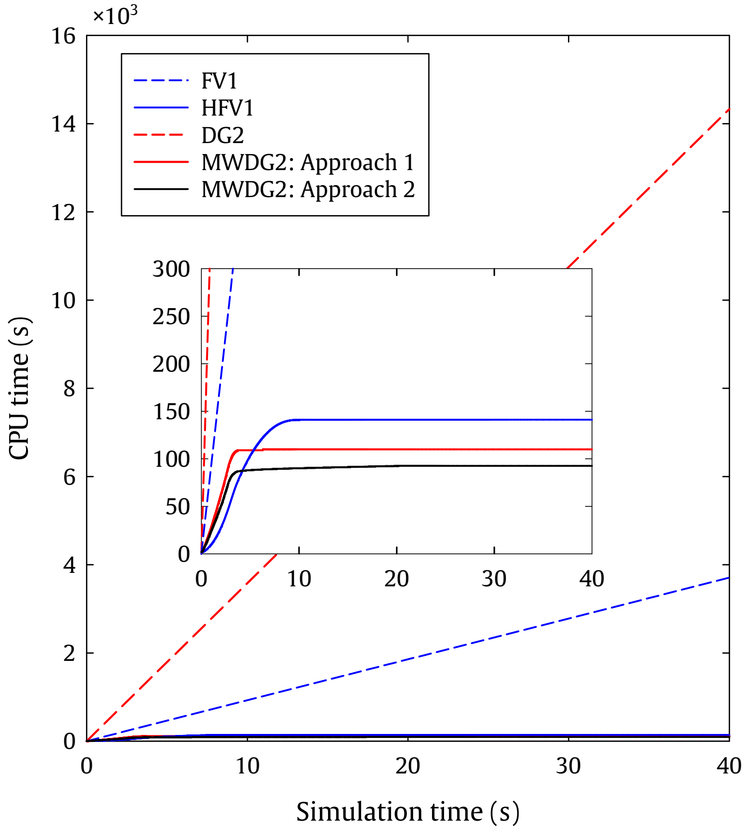

Figure 8. 1D dam-break flow on a wet-domain with shock. Cumulative CPU times over 40 s long simulation generated for the MWDG2 solver with Approach 1 and Approach 2, and for the HFV1 solver along with those of the uniform FV1 and DG2 solvers.

In the zoom-in portion within Fig. 8, the overhead cost of (multi)wavelet adaptivity in the HFV1 and MWDG2 solvers can be identified. At $t < 3.5$ s, when both shock and rarefaction waves were present in the domain, the MWDG2 solvers entail greater costs compared to those of the HFV1 solver. After $t \geq 3.5$ s once the shock has exited the domain, the CPU time of both MWDG2 solvers becomes considerably (and consistently) lower than the CPU time of the HFV1 solver that has to process more elements (see Fig. 7). With MWDG2, its CPU time with Approach 1 is found to be slightly higher than its CPU time with Approach 2. This extra overhead cost with Approach 1 may be expected, due to its extra detail coefficient for decoding and encoding, though it pays off by triggering more aggressive coarsening with MWDG2 (recall Fig. 7).

Finally, CPU times at the end of the simulation are analysed in terms of runtime ratios of the adaptive solvers to those of the uniform solvers to quantify relative 2D runtime efficiency gains. The adaptive MWDG2 solver with both Approach 1 and Approach 2 is found to be about 140 times faster to complete than the uniform DG2 solver, thereby offering 7 times more efficiency speedup compared to the 1D case (Kesserwani et al., 2019). The adaptive HFV1 solver is found to be 26 times faster to complete than the uniform FV1 solver in the 2D case, offering 13 times more efficiency speedup compared to the 1D case (Kesserwani et al., 2019). Most notably, the MWDG2 solvers are found to be around 40 times faster to complete than the uniform FV1 solver for this test.

3.1.2. Circular 2D dam-break flow

The circular dam-break flow test (Toro, 2001) is applied to further verify the implementation of MWDG2 with Approach 1 and Approach 2 when modelling entirely 2D flow hydrodynamics. In this test, a cylindrical imaginary dam is assumed to be located in the middle of a 20 m × 20 m domain, with a circular base of 2.5 m radius. Initially, the water depth inside the cylindrical dam is 2.5 m, separating it from a water depth of 0.5 m elsewhere. The domain is assumed to be enclosed by solid walls and to have a frictionless and flat bed. The adaptive MWDG2 solver is applied with Approach 1 and Approach 2 set up with $\varepsilon = 10^{-3}$, a baseline grid of 2 × 2 elements and $L = 7$. Following Kesserwani et al. (2018), simulations are run up to 4.7 s and model outputs are analysed at $t = 0.4$, 0.7, 1.4, 3.5 and 4.7 s along the radial direction $r = \sqrt{x^2 + y^2}$. A reference solution was produced by numerically solving the 1D radial SWE with a dimensionality source term using a well-balanced second-order finite volume solver on a grid with 1001 elements (Toro, 2001).

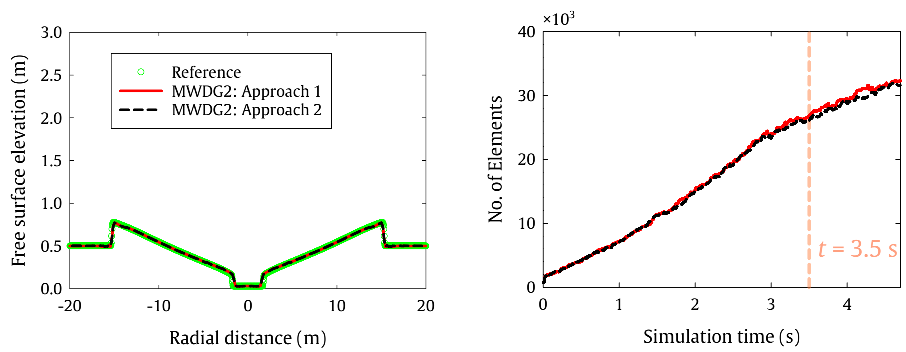

Figure 9. Circular 2D dam-break flow. Adaptive MWDG2 solver predictions with Approach 1 and Approach 2: the left side shows the cross-sectional profile of the free-surface elevations along the radial distance at $t = 3.5$ s; whereas, the right side contains the time history of the number of element consumed by Approach 1 and Approach 2. The dashed line flags the point in time where the adaptive grids produced by MWDG2 are illustrated (see Fig. 10).

Fig. 9 (left) shows the cross-section of the computed free-surface water elevations by the MWDG2 solvers at $t = 3.5$ s and the reference solution. Both Approach 1 and Approach 2 are observed to reproduce the reference solution with indistinguishable (visual) difference. This same observation was also noted for the plots of the MWDG2 solver outputs at $t = 0.4$, 0.7, 1.4 and 4.7 s (not shown). Fig. 9 (right) also includes the time history of the number of elements consumed by using Approach 1 *vs* Approach 2 with MWDG2 to explore their grid prediction ability. Both approaches predict almost the same number of elements up to 4.7 s, when flow is still highly dynamic. This again reinforces that Approach 1 and Approach 2 particularly yield close grid predictions when modelling highly dynamic flow cases. This can be further seen in Fig. 10, which shows the 2D grid predicted at $t = 3.5$ s: both Approach 1 and Approach 2 lead to symmetrical grid resolution patterns in track with the symmetrical flow features of the dam-break wave propagation at $t = 3.5$ s (Fig. 9, left) with minor differences. Hence, both Approach 1 and Approach 2 are equally valid for use in the context of devising an adaptive MWDG2 solver for modelling of highly dynamic 2D shallow water flow.

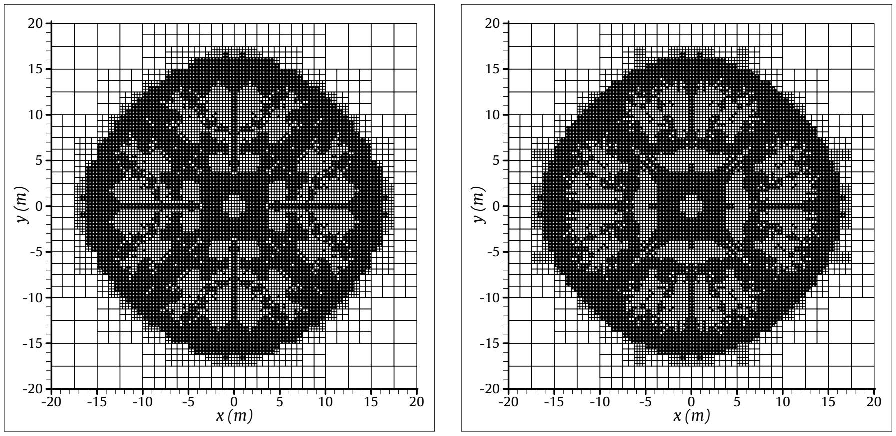

Figure 10. Circular 2D dam-break flow.

Adaptive grids predicted by the MWDG2 solver with Approach 1 (left) and Approach 2 (right) at $t = 3.5$ s.

3.1.3. Lake-at-rest over terrain blocks with wet-dry zones and fronts

Initially proposed in Kesserwani et al. (2018), this test case was used to explore and verify the full extent of well-balancedness for the (unscaled) slope-decoupled uniform DG2 solver, i.e. at the level of both average and slope coefficients. It is re-investigated to further assess this aspect for the adaptive solvers, mainly for MWDG2 with Approach 1 and Approach 2, and their grid generation ability. The test considers a quiescent flow enclosed in a 75 m × 30 m domain which has a frictionless floodplain with three aligned blocks of an identical shape. However, each of the blocks has a different height to ensure this test covers all scenarios of wet-dry zones and fronts. Two geometrical types for the blocks are introduced to represent differentiable and non-differentiable topography shapes: conical and rectangular with initial free-surface water elevations of 1.78 m and 1.95 m, respectively. The adaptive solvers are run to complete a 100 s simulation on a baseline grid of 2×1 elements with $L = 7$ and $\varepsilon = 10^{-3}$. The MWDG2 solvers required 7,000 time-steps to complete the simulation whereas the HFV1 solvers completed it after 3,500 time-steps. Throughout the simulations, all the adaptive solvers have kept the initial free-surface water elevation intact and the initial flow unperturbed. This can be confirmed by analysing the time histories of the maximum error of the discharge coefficients spanning the assembled DG2 solution, i.e. q_x^0, q_y^0, q_x^{1x}, q_y^{1x}, q_x^{1y} and q_y^{1y}, with $q_x = hu$ and $q_y = hv$. These errors are plotted in Fig. 11.

With both differentiable (Fig. 11a) and non-differentiable (Fig. 11b) topography cases, the discharge errors generated by MWDG2 with Approach 1 (left side of Fig. 11a) and Approach 2 (right side of Fig. 11a) remain below 10^{-11} and 10^{-14}, respectively. The slightly larger error variations with Approach 1 are likely to arise from the extra operations on truncation, encoding and decoding due to its extra detail coefficient. However, with both Approach 1 and Approach 2, the errors remain bounded with no sign of growth. Note that the errors associated with the discharge coefficients of the adaptive HFV1 solver (not shown) were found to be close to those of Approach 2 with MWDG2. These findings confirm that the adaptive MWDG2 and HFV1 solvers achieve 2D well-balanced predictions despite the presence of wet-dry fronts across steep bed-slopes with continuous and discontinuous terrain geometries.

As the initial free-surface elevation and zero discharges do not vary in time, the adaptive grids are solely selected

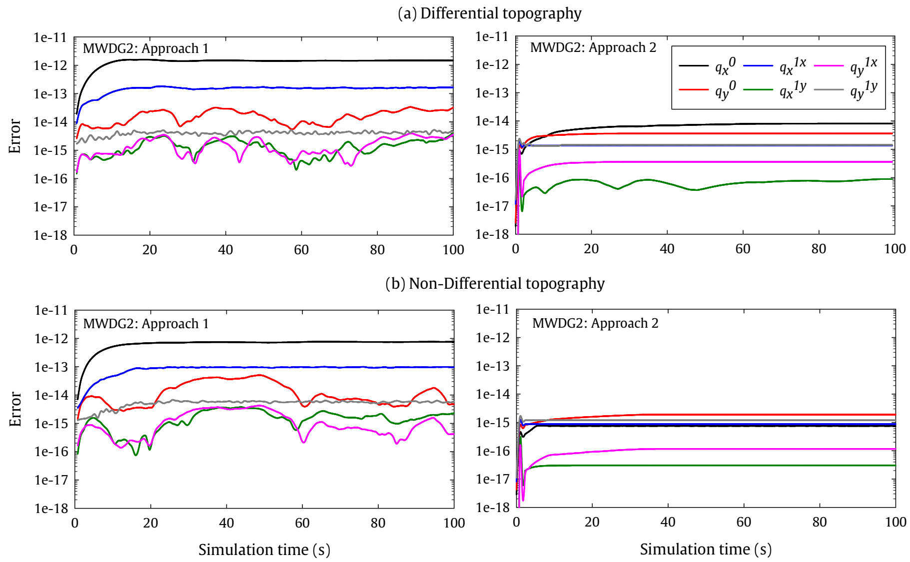

Figure 11. Lake-at-rest over terrain blocks with wet-dry zones and fronts. Time histories of the maximum errors for the discharge coefficients simulated by the MWDG2 solvers. Panels (a) and (b) contain the error histories for the case with differentiable (conical) and non-differentiable (rectangular) topography types, whereas the left and right panels contain the predictions made with Approach 1 and Approach 2, respectively.

according to the features of the terrain blocks. This allows to also investigate the potential of the adaptive well-balanced solvers for use as multiresolution grid generators (Liang et al., 2015; Hou et al., 2018). Fig. 12 shows the 2D multiresolution grids predicted by the adaptive solvers after completing the 100 s simulation together with the 2D contour maps of the topography. For the differentiable case (Figs. 12a and 12b), MWDG2 and HFV1 predicted the highest resolution allowable along the circular kinks at the base of the conical blocks. However, HFV1 comparatively predicted a wider extent of fine resolution, which could be expected given its piecewise-constant representation. The over-refinement tendency of HFV1 becomes clearer for the portions of the grid inside the conical blocks where the topography is non-flat and has a linear slope. There, HFV1 predicted a uniform grid that is only one-level coarser in resolution, whereas MWDG2 allowed more coarsening up to two levels, due to its piecewise-planar representation. Outside the conical blocks, where the topography is flat, grid portions were more aggressively coarsened by both solvers, though MWDG2 allowed an additional level of coarsening, likely due to its higher order-of-accuracy. For the non-differentiable case (Figs. 12c and 12d), MWDG2 and HFV1 are found to exhibit the same grid prediction patterns, as those observed in the differentiable case (Figs. 12a and 12b), around the portions of the grid where the topography is flat (i.e. outside and inside the rectangular blocks) and along the rectangular discontinuity. These results imply that both MWDG2 and HFV1 are usable to generate multiresolution grids and that the use of MWDG2 would allow more effective coarsening for curved and linear topography shapes, such as for a mountainous catchment (Özgen Xian et al., 2020).

3.2. Application to reproduce more realistic tests

The testing in Section 3.1 showed that choosing ε around 10^{-3} is sufficient for the HFV1 and both MWDG2 solvers to preserve the accuracy and robustness of their uniform counterparts, and for making them more efficient to run than the uniform FV1 solver. The testing also suggests favouring Approach 1 for the adaptive MWDG2 solver given its ability

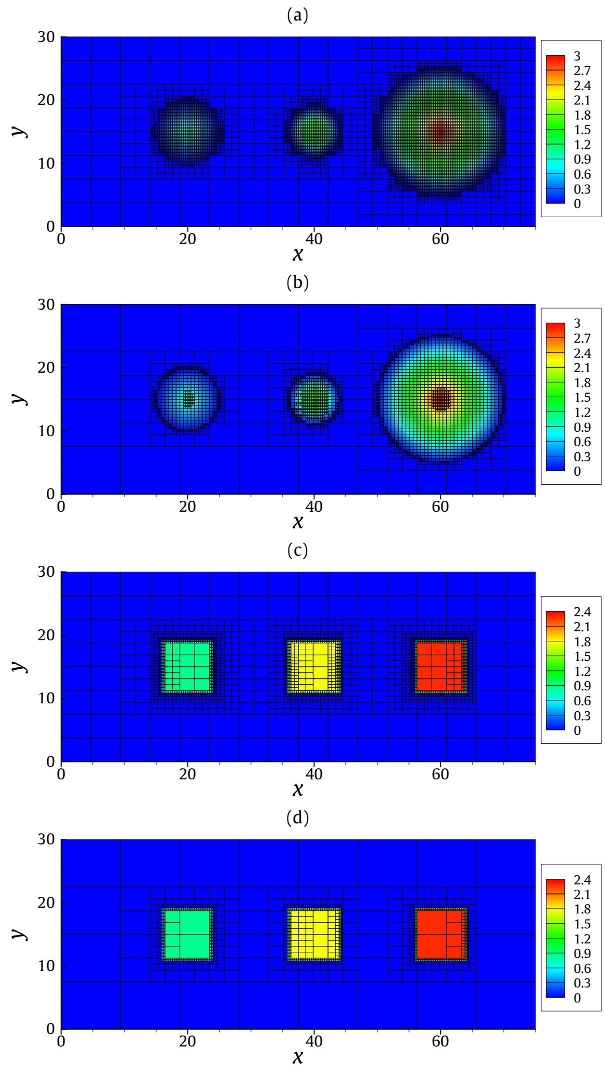

Figure 12. Lake-at-rest over terrain blocks with wet-dry zones and fronts. Multiresolution grids predicted by the adaptive HFV1 and MWDG2 solvers after completing the 100 s simulation (MWDG2 outcomes were identical with Approach 1 and Approach 2): Panels (a) and (b) show the grids for the case with the differentiable topography (conical); and, panels (c) and (d) show the grids for the case with the non-differentiable topography (rectangular).

to coarsen more aggressively than with Approach 2 for smooth and flood-like flows. Hence, the numerical tests in this section will only consider MWDG2 with Approach 1 with an ε value of 10^{-3}, considered alongside the upper or lower limit. The first test replicates a symmetrical flow interaction with a conical island (Section 3.2.1). The second test simulates a laboratory-basin replica of Okushiri tsunami in 1993 (Section 3.2.2), including non-symmetrical wetting and drying processes and wave run-up inundating a coastal area. The third test reproduces the 1959 Malpasset dam-break (Section 3.2.3) that propagated over an initially dry and rugged valley before inundating an urban area. In all tests, the adaptive HFV1 and MWDG2 solvers are studied alongside their uniform FV1 and DG2 counterparts to compare their predictive accuracy, grid-coarsening capability and runtime efficiency.

3.2.1. Symmetrical flow around a conical island

This test case involves run up of a solitary wave over a conical island (Briggs et al., 1995), and has inspired several comparisons in computational hydraulics literature (Liu et al., 1995; Valiani and Begnudelli, 2006; Vater et al., 2019; Ma et al., 2019). It is employed here to further study the performance of the adaptive HFV1 and MWDG2 solvers when simulating a symmetrical 2D flow over topography with wetting and drying. The 2D domain includes a 26 m × 27.6 m basin in which the conical island is located at its centre (12.96 m, 13.80 m). The solitary wave is generated at the western inlet boundary with an amplitude of 0.014 m. It propagates over a mean water depth of 0.32 m. The details of the numerical wave generation can be found in (Liu et al., 1995). HFV1 and MWDG2 simulations are run on a baseline coarsest grid made of a single element, for two choices of $\varepsilon \in \{10^{-4}, 10^{-3}\}$ and with a maximum resolution level $L = 9$. Under this setting, HFV1/MWDG2 allow up to 262, 144 elements on the grid with the maximum resolution level, i.e. $\Delta x^{(L)} = \Delta y^{(L)} \approx 0.05$ m. On this grid, uniform FV1 and DG2 simulations are run.

Experimental time series for the free-surface water elevation are available at several gauge points that surround the island (Briggs et al., 1995). Here, computed time series for the free-surface water elevation are compared with the experimental ones at four gauge points: G1 (9.36 m, 13.80 m), G2 (10.36 m, 13.80 m), G3 (12.96 m, 11.22 m) and G4 (15.56 m, 13.80 m). Point G1 is located at the kink of the island from the side where the wave first arrives (i.e. towards the western side, see Fig. 13). Points G2, G3 and G4 are located at the middle of island from the western, southern and eastern sides, respectively (see Fig. 13). Since the wave is generated (and travels from) the west, more discrepancies among the solver predictions would be expected at the gauge point located downstream, towards the east. Note that, for this test, a discrepancy between the numerical and the experimental wave patterns at point G2 is commonly reported, but is attributed to the lower accuracy in the experimental wave capturing at the western gauge point on the island (Briggs et al., 1995; Tonelli and Petti, 2010; Lannes and Marche, 2015). The simulations are run up to 30 s without friction effects, following other investigators (e.g. Liu et al., 1995; Valiani and Begnudelli, 2006).

Fig. 13 includes the numerical time series of the free-surface water elevation computed by HFV1 and MWDG2 at $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-3}$, those of their FV1 and DG2 counterparts and the experimental ones. The water levels produced by DG2 and MWDG2 at $\varepsilon = 10^{-4}$ are found be very close to each other, and are in a very good agreement with the experimental water levels and those reported for an alternative uniform DG2 solver (Vater et al., 2019). The water levels produced by FV1 and HFV1 at $\varepsilon = 10^{-4}$ are found to fairly trail the experimental water level profiles, except at point G4 where they completely fail in capturing the last phase of wave drop and rise, i.e. after $t = 17.5$ s (Fig. 13). The performance of the HFV1 solver deteriorates significantly at $\varepsilon = 10^{-3}$, leading to water levels that clearly deviate from its first-order model counterparts. In contrast, MWDG2 at $\varepsilon = 10^{-3}$ reproduces water level profiles that are much more aligned with those predicted by its second-order model counterparts, and fairly[2] captures the presence of the last phase at G4.

In terms of multiresolution grid prediction, Fig. 14 shows the grids predicted by HFV1 (left) and MWDG2 (left) at 12 s, considering both threshold error values $\varepsilon = 10^{-3}$ (Fig. 14a) and $\varepsilon = 10^{-4}$ (Fig. 14b). At $\varepsilon = 10^{-3}$, the coarse resolution

[2]Within the scope of shallow flow modelling. Capturing the flow curvature of a long wave such as a solitary wave seems to require $\varepsilon \geq 10^{-4}$ (Sharifian et al., 2019).

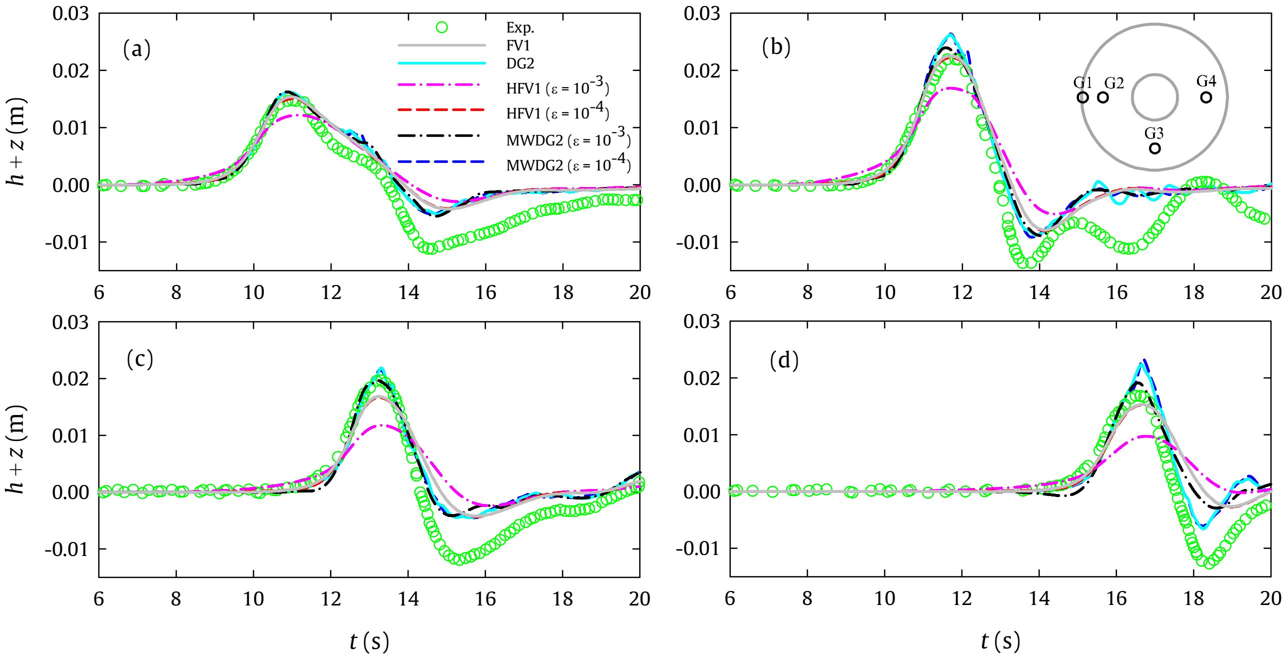

Figure 13. Symmetrical flow around a conical island. Computed and experimental time histories of the free-surface water elevation at gauges. Panel (a) to (d) contain the plots associated with points G1 to G4, where their locations are included in a subfigure within panel (b).

portions in HFV1's grid are at least one level-of-resolution finer compared to those in MWDG2' grid, which is to be expected given the higher-order accuracy of MWDG2. In terms of resolution refinement, HFV1 is noted to uniformly deploy the finest resolution along the kinks of the solitary wave and within the surrounding of the island (i.e. the area including the gauge points). In contrast, MWDG2 keeps at least twice-coarser resolution for the grid for these same portions, and only deploys the finest resolution along the circular extents of the reflected wave and the wet-dry frontline. At $\varepsilon = 10^{-4}$, HFV1's grid becomes excessively finer as it over-uses the finest resolution within the portions surrounding the island and along with the solitary wave. In contrast, MWDG2 at $\varepsilon = 10^{-4}$ still keeps using the finest resolution only along the circular extents of the reflected wave and the wet-dry front. But now, it incorporates two more levels of resolution in track with the solitary wave and within the area bounded by the circular extents of the wet-dry frontline and the reflected wave (i.e. where water level information is sampled at the gauge points: this could be a reason why the water level histories predicted by MWDG2 at $\varepsilon = 10^{-3}$ are slightly different from those of MWDG2 at $\varepsilon = 10^{-4}$).

In terms of runtime speed-up (see Table 1), DG2 is found to be around 9 times slower to run than FV1, and is thus the most expensive choice. At $\varepsilon = 10^{-4}$, HFV1 is found to be 2 times faster to run than FV1, but MWDG2 is 11 times faster to run than DG2. Supported also by the results in Fig. 13, running MWDG2 at $\varepsilon = 10^{-4}$ seems to be more beneficial for such a highly dynamic flows to better preserve the predictive quality of the expensive DG2 (Sharifian et al., 2019), while remaining cheaper to run than uniform FV1. At $\varepsilon = 10^{-3}$, HFV1 is 13.5 times faster to run than FV1 but MWDG2 becomes 131 times faster to run than DG2. The faster rate of efficiency gain with MWDG2 is attributed to its more aggressive coarsening gained by the smoothness of its piecewise-planar solution (Kesserwani et al., 2019). Most strikingly, at $\varepsilon = 10^{-3}$, MWDG2 is able to complete the simulation while being 15 times faster than the uniform FV1 solver.

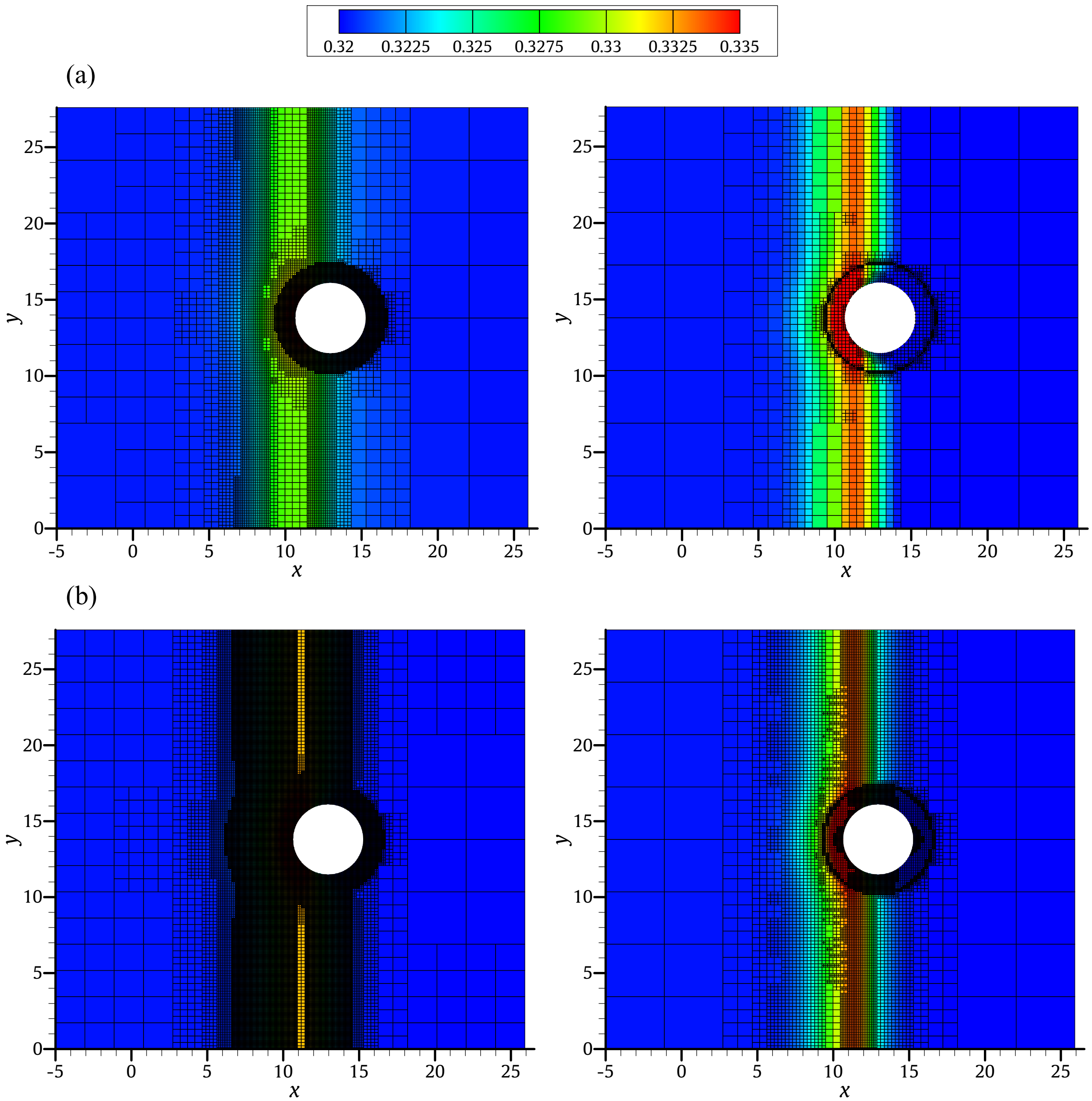

Figure 14. Symmetrical flow around a conical island. Water depth contours and multiresolution grids predicted at t = 12 s by HFV1 (left) and MWDG2 (right): panel (a) shows the results for $\varepsilon = 10^{-4}$; panel (b) shows the results for $\varepsilon = 10^{-3}$.

Overall, this test makes a strong case for preferring MWDG2 to simulate highly dynamic shallow water flows. HFV1 can at best deliver the outcomes of FV1; it does not gain enough efficiency at $\varepsilon = 10^{-4}$ (hindered by wavelet-adaptivity overheads due to over-refinement), and delivers the least accurate predictions at $\varepsilon = 10^{-3}$. MWDG2 at $\varepsilon = 10^{-3}$ delivers very close outputs to those of the expensive DG2 model counterpart, but gains significant efficiency up to becoming 15 times faster than the less-accurate uniform FV1 model.

Table 1. Symmetrical flow around a conical island. Runtime ratios for the MWDG2, HFV1, DG2 and FV1 solvers to complete a 30 s long simulation.

Runtime ratios	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$
DG2/FV1	8.72X	8.72X
FV1/HFV1	13.5X	2.1X
DG2/MWDG2	131.0X	10.75X
FV1/MWDG2	15.0X	1.23X

3.2.2. Tsunami run-up over a complex 3D beach

The performance of the adaptive MWDG2 and HFV1 solvers is explored in reproducing a 1:400 scaled experimental replica of the 1993 Okushiri tsunami (Matsuyama and Tanaka, 2001). The 5.488 m × 3.402 m computational domain has a uniform resolution of 0.014 m × 0.014 m on a Digital Elevation Model (DEM) made of 163,840 elements (Fig 15a). The tsunami-generated flood wave enters the domain from the western boundary over an initial water depth of 0.135 m and lasts up to 22 s (Fig. 15b). Experimental histories of the water surface elevation are available at three gauge points: G1 (4.521 m, 1.196 m), G2 (4.521 m, 1.696 m) and G3 (4.521 m, 2.196 m), located upstream of the coastal area hit by the tsunami-generated flooding (Fig. 15a). The adaptive MWDG2 and HFV1 solvers are run on a baseline grid made of 8 × 5 elements and at $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-3}$. With $L = 6$, the adaptive HFV1/MWDG2 solvers allow up to 163,840 elements on the finest resolution level with $\Delta x^{(L)} = \Delta y^{(L)} = 0.014$ m. The uniform FV1/DG2 solvers are run at the same resolution of the DEM. Simulations are run up to 25 s with a Manning coefficient of 0.01 $\mathrm{s}^{-1}\mathrm{m}^{1/3}$ and using reflective boundary conditions for the northern and southern boundaries.

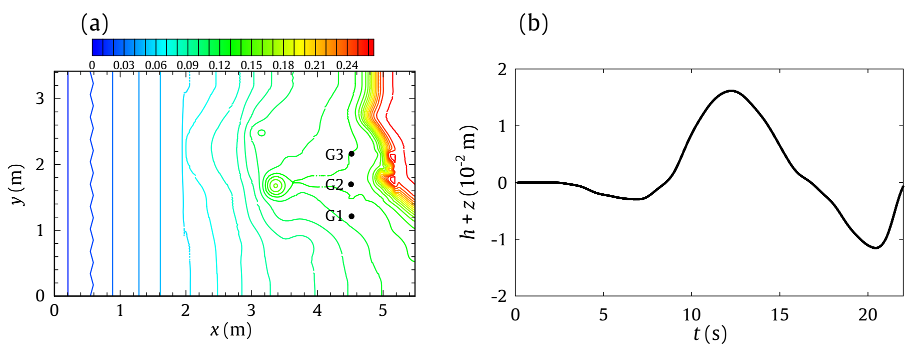

Figure 15. Tsunami run-up over a complex 3D beach: (a) DEM contours including the three gauge points G1, G2 and G3 where experimental time histories of the free-surface water elevation are available, and (b) tsunami-generated inflow hydrograph for the free-surface elevation entering the western boundary during 22 s.

The numerical time histories produced by HFV1 and MWDG2, at $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-4}$, and those of FV1 and DG2 were compared with the experiments at gauge G1, G2 and G3. Fig. 16 shows these time histories at point G2 because it is the closest to the shoreline that was flooded, and has higher and more abrupt variations as compared to the histories at point G1 and G3 (not shown). As expected for a tsunami wave propagation: a gradual drop in water level is predicted during 10 s $< t <$ 15 s, indicative of an initial wave retraction, and then an abrupt increase in water level during $t \geq 16$ s that reaches its peak at $t = 17$ s, predicting a flood wave propagation towards the coastal area thereafter.

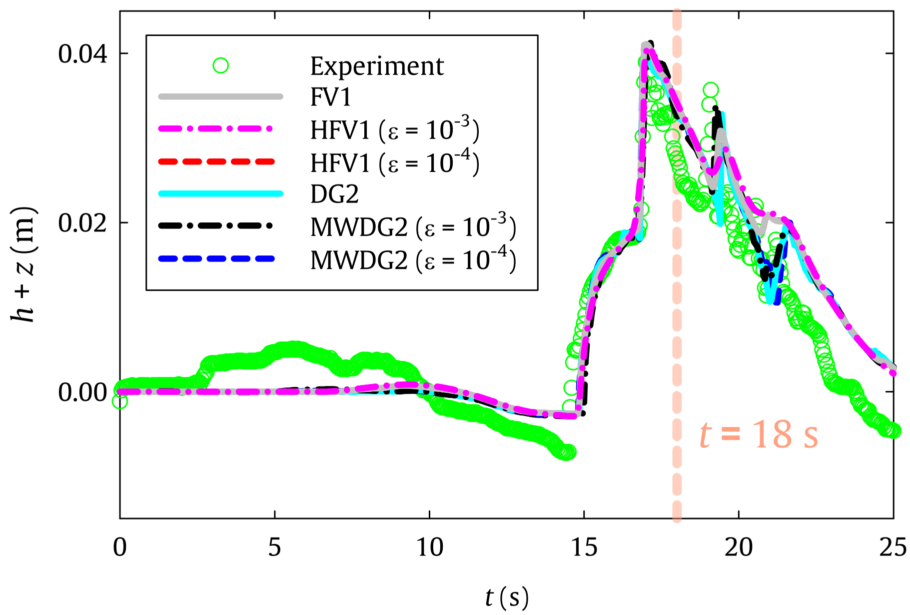

Figure 16. Tsunami run-up over a complex 3D beach. Time histories of the free-surface water elevation at gauge G2 where the numerical results and experimental measurements are compared. The numerical results are produced by the uniform FV1/DG2 solvers and the adaptive HFV1/MWDG2 solvers at $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-4}$.

At $t \leq 18$ s, all adaptive and uniform solvers correctly predict the experimental arrival time and amplitude of the wave. With the occurrence of the weaker wave reflections during $19\,\text{s} \leq t \leq 22\,\text{s}$, the FV1/HFV1 solvers predict results that deviate from the experimental measurements (by $t = 19$ s). These discrepancies become clear after the second wave reflection (by $t = 22$ s) where HFV1/FV1 smear arrival time prediction. In contrast, the uniform DG2 solver and the MWDG2 solver at $\varepsilon = 10^{-4}$ reliably predict the experimental arrival time during $19\,\text{s} \leq t \leq 22\,\text{s}$. Similar behaviour is observed with MWDG2 at $\varepsilon = 10^{-3}$, which tends to deliver predictions closer to its MWDG2/DG2 model counterparts. At $t \geq 22$ s, all the numerical models overestimate the experimental measurements. These results again indicate better predictive capability for the MWDG2/DG2 solvers due to their piecewise-planar (second-order) accuracy as compared to the piecewise-constant (first-order) accuracy of the HFV1/FV1 solvers.

To further assess the grid prediction capability of the MWDG2 and HFV1 solvers, their predicted grids are analysed at $t = 18$ s (see dashed line in Fig. 17), where both solvers provided equally close agreement with the experimental measurements at $\varepsilon = 10^{-3}$. Fig. 17 shows the 2D multiresolution grids alongside the water depth contours predicted by the adaptive HFV1 and MWDG2 solvers. As can be seen in Fig. 17b, the MWDG2 solver only deploys the highest resolution to track two reflected waves and the wavefronts along the flooded coastal area, and aggressively coarsened elsewhere. In contrast, the HFV1 solver over-refines the grid (Fig. 17a): it deploys a much wider extent for the highest resolution along the coastal area and involves much finer resolution throughout the domain. Moreover, MWDG2 predicts wider wetting extent around the island in the middle of the domain and at the critical zone downstream of G2 where the flooding first struck (compare water depth contours between Fig. 17a and Fig. 17b). These results are aligned with the initial finding in

Section 3.1.1 — reinforcing the ability of the MWDG2 solver to more economically coarsen resolution as compared to the HFV1 solver for such a flow type, which is predominantly gradual with mild discontinuous transients (i.e. wave reflection and diffraction).

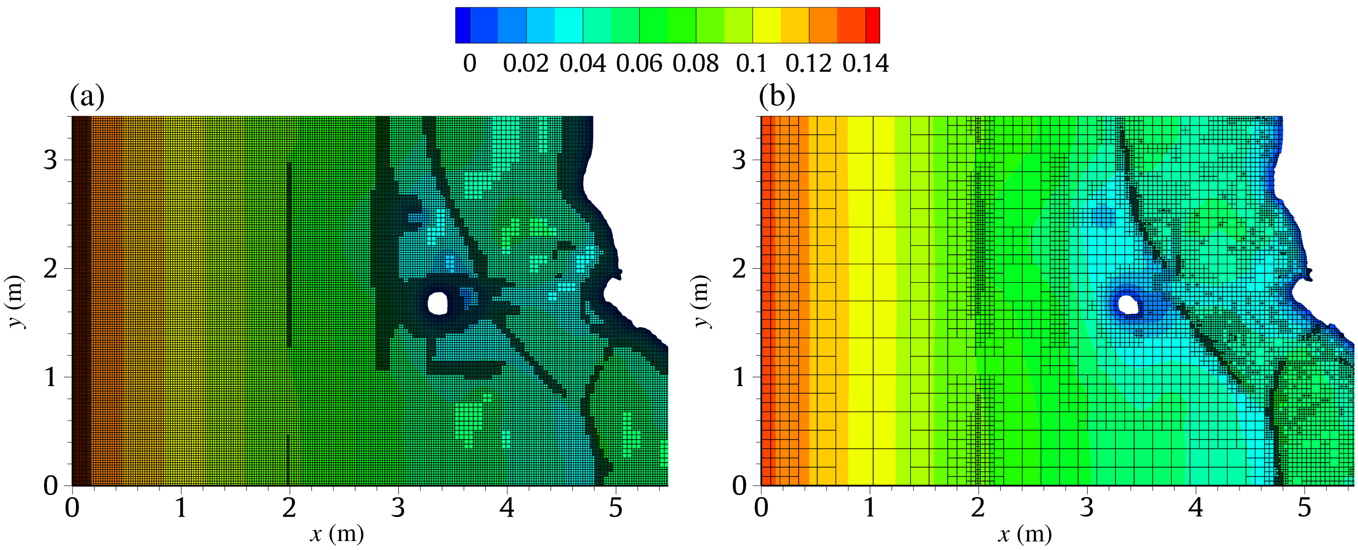

Figure 17. Tsunami run-up over a complex 3D beach. Water depth contours and adaptive grids at $t = 18$ s predicted by the adaptive solvers at $\varepsilon = 10^{-3}$: (a) HFV1 and (b) MWDG2.

To study the efficiency of HFV1/MWDG2 relative to their uniform solver counterparts, their CPU runtimes are analysed as per the ratios listed in Table 2. CPU runtimes were recorded for all the solvers after completing the full 25 s simulation, and using $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-4}$ for the adaptive solvers. As expected, the uniform DG2 solver is 3.8 times more expensive to run than uniform FV1 for this test. At $\varepsilon = 10^{-4}$, the adaptive HFV1 solver is almost as costly to run as the uniform FV1 solver, and is only 3 times more efficient at $\varepsilon = 10^{-3}$ — most likely due to the high cost of wavelet-adaptivity overhead identified for gradual flow types (recall Fig. 8), which led to the overly refined grid shown in Fig. 17. In contrast, MW's adaptivity in MWDG2 made it 7 and 39 times faster than the uniform DG2 solver at $\varepsilon = 10^{-4}$ and $\varepsilon = 10^{-3}$, respectively. Because of this, MWDG2, at $\varepsilon = 10^{-3}$, is found to be around 10 times faster than uniform FV1 for this test.

Table 2. Tsunami run-up over a complex 3D beach. Runtime ratios for the MWDG2, HFV1, DG2 and FV1 solvers to complete a 25 s long simulation.

Runtime ratios	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$
DG2/FV1	3.82X	3.82X
FV1/HFV1	3.03X	1.05X
DG2/MWDG2	38.77X	7.06X
FV1/MWDG2	10.14X	1.84X

All considered, for gradual to energetic flow types, the adaptive MWDG2 solver can produce comparable predictive accuracy to the expensive uniform DG2 solver, while remaining up to 10 times faster than the less accurate uniform FV1 solver.

3.2.3. *Malpasset dam-break*

In this final test, the MWDG2 and HFV1 solvers are studied to replicate the Malpasset dam break. This test case is frequently used as a benchmark test for shallow water numerical models (Hervouet and Petitjean, 1999; Valiani et al., 2002; Brufau et al., 2004; Liang et al., 2007; Shi and Nguyen, 2008; Wang et al., 2011; Delis et al., 2011; Kesserwani and Liang, 2012a; Guermond et al., 2018; Zhao et al., 2019). The Malpasset dam was located in the Reyran river valley in France. It collapsed in 1959 leading to a dam-break wave hitting an urban area located downstream of the valley. After the disaster, the police collected data on maximum free-surface water elevations by surveying watermarks left by the flood. In 1964, Electricité de France (EDF) built an undistorted 1:400 physical model and measured the propagation time and maximum height of the flood wave at gauge points located near the police survey points (Goutal, 1999). The study domain has an area of 17.5 km $\times$ 9 km with a DEM at a resolution of 20 m $\times$ 20 m, representing the finest uniform grid. The free-surface water elevation upstream of the dam is 100 m and the rest of the domain is initially dry. The MWDG2 and HFV1 solvers deployed a baseline grid with 7×4 elements and $L = 7$, thus allowing up to a maximum of 458,752 elements at the finest resolution level L with $\Delta x^{(L)} = \Delta y^{(L)} = 20$ m. On this fine uniform grid, FV1 and DG2 simulations are executed. The simulation time is $t = 40$ min with a uniform Manning coefficient of $0.033\ \mathrm{s}^{-1}\mathrm{m}^{1/3}$. As the size of the finest resolution level for this test is relatively coarse and the choice of ε is reported to be proportional to the finest feature accessible (Gerhard et al., 2015a), $\varepsilon \in \{10^{-3}, 10^{-2}\}$ is explored with the adaptive solvers for this test.

Numerical outputs of the HFV1/FV1 and MWDG2/DG2 solvers were compared for reproducing the maximum water levels at both the police survey points and at the physical measurement points, where all the solvers showed very close predictions to the measurement data (not shown). Hence, only the outputs in terms of arrival times at the measurement points are analysed, where clearer discrepancies appear among solver predictions. These arrival times are shown in Fig. 18 at gauges 1-9 located along the Reyran river valley up to the start of the urban area that was inundated 40 minutes after the dam collapsed. At gauges 1-5, the arrival times predicted by MWDG2 at $\varepsilon = 10^{-3}$ and DG2 are very similar, both showing a very good agreement with the measured arrival times. The arrival times predicted by HFV1 for both ε values and by FV1 are also very similar, but overestimate the measured arrival times. At $\varepsilon = 10^{-2}$, the arrival times predicted by MWDG2 are slightly higher than those of DG2 and MWDG2 at $\varepsilon = 10^{-3}$, but are noted to be closer to the arrival times predicted by HFV1/FV1. However, MWDG2's arrival times at $\varepsilon = 10^{-2}$ reproduce a closer trend to those of its second-order DG2/MWDG2 counterparts, despite being deviated. Such deviations are likely to be attributed to the more aggressive thresholding of slope coefficients with MWDG2 at $\varepsilon = 10^{-2}$ —- as opposed to HFV1 that only thresholds average coefficients. This could have led to more aggressive thresholding to the modelled information along the propagating wave front, which could be the reason why MWDG2 arrival times at $\varepsilon = 10^{-2}$ are somewhat closer to those of HFV1/FV1. Nonetheless, MWDG2 predictions at $\varepsilon = 10^{-2}$ can be deemed acceptable: beside the sources of uncertainty in this test (sensitivity to the choice of the Manning's coefficient and errors in experimental data measurement), its predictions are found to be in a better agreement with measured data compared with other predictions made by a second-order finite volume solver (FV2) on a triangular mesh type (Zhao et al., 2019), and by a third-order MWDG (MWDG3) solver designed with different treatments to achieve well-balancedness with wetting and drying (Caviedes-Voullième et al., 2020).

At gauges 6-9 (Fig. 18), HFV1/FV1 clearly overestimate the measured arrival times, whereas MWDG2/DG2 (predomi-

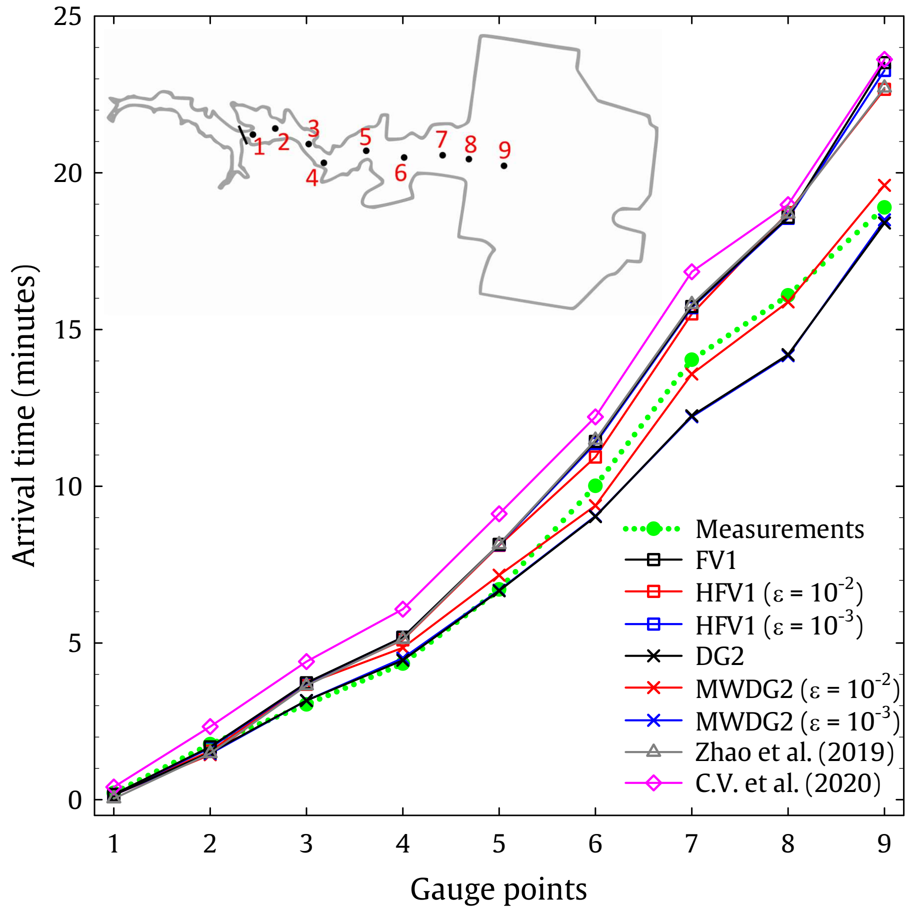

Figure 18. Malpasset dam-break. Arrival times predicted by the uniform DG2/FV1 and the adaptive MWDG2/HFV1 solvers at $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-2}$ compared to measured arrival times at gauge points 1 to 9 shown by dots within the subfigure. The predictions are also compared with numerical results of Zhao et al. (2019) using a uniform FV2 solver, and Caviedes-Voullième et al. (2020) (denoted as C.V. et al., 2020) using an MWDG3 solver.

nantly) exhibits an underestimating trend with relatively lesser deviations. These results imply that the first-order solvers predict slower arrival times (i.e. maximum discrepancy of about 7 min), whereas the second-order solvers predict faster but more reliable arrival times (i.e. maximum discrepancy of about 2 min). The better performance with MWDG2/DG2 can be associated with their piecewise-planar accuracy (Kesserwani and Wang, 2014), that allows much smoother tracking of the wetting front of the dam-break wave as compared to the piecewise-constant accuracy of HFV1/FV1 that handles wetting and drying more crudely. At gauge 9, when the dam-break wave starts to spread into the urban area by t = 19 min (Fig. 19): MWDG2/DG2 predict arrival times that are close to 19 min, whereas HFV1/FV1's arrival times are much slower, being about 23.5 min. Hence, MWDG2/DG2 predict faster and more reliable arrival times than HFV1/FV1, leading to faster filling patterns (Figs. 19 and 20).

The differences between the adaptive HFV1 and MWDG2 solvers are further analysed by looking at their predicted 2D water depth contours and grids at t = 20 min, i.e. up to the time when the dam-break wave starts to spread into the urban area. Fig. 19 shows the predictions at $\varepsilon = 10^{-2}$: compared to HFV1, MWDG2 clearly predicts a wider wetting extent by the start of the urban area along with signs of narrower drying extents, such as at the downslope tributary branch located in the

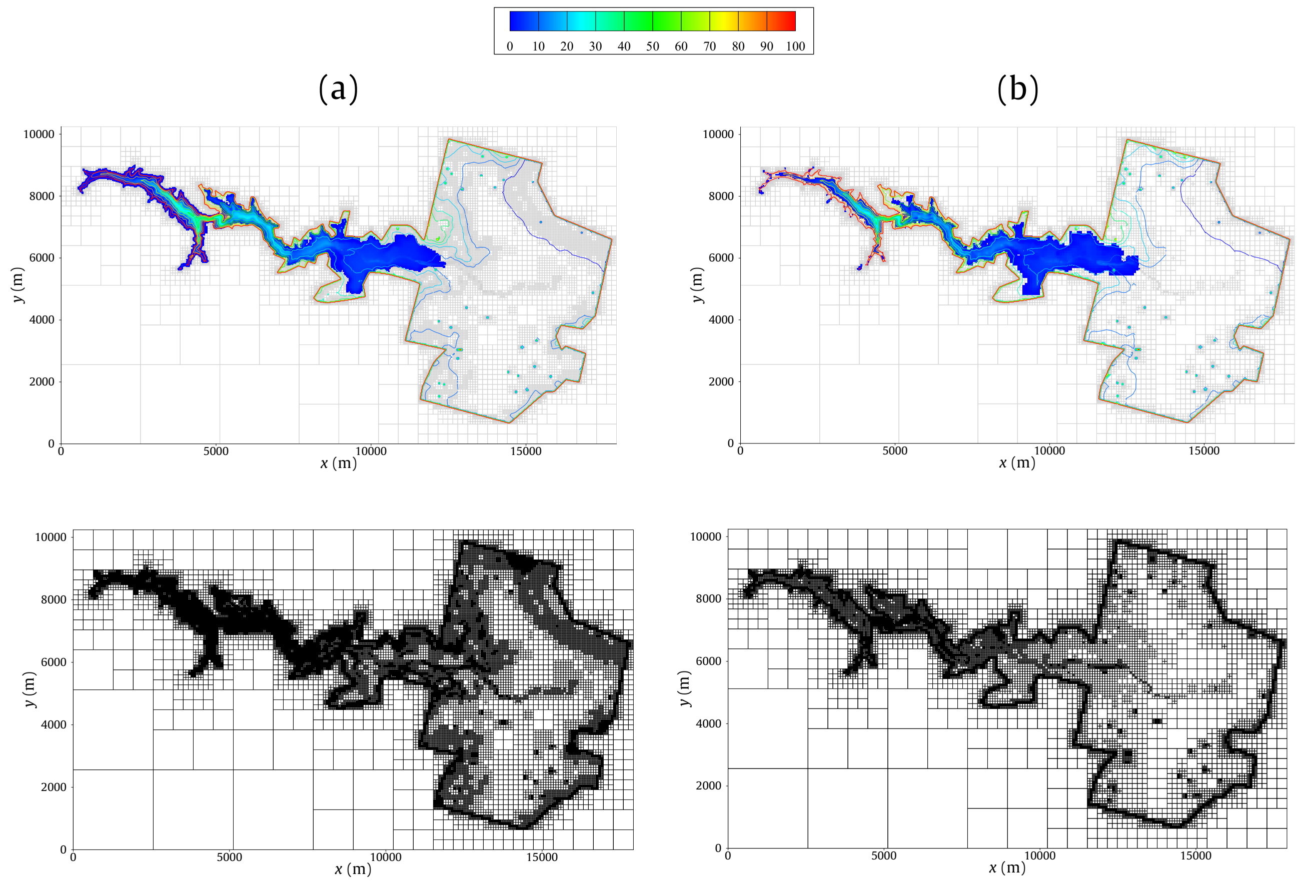

Figure 19. Malpasset dam-break. Flood map contours over topography (top) and adaptive grids (bottom) predicted by the adaptive solvers at $\varepsilon = 10^{-2}$ and for $t = 20\,\mathrm{min}$: (a) HFV1 and (b) MWDG2.

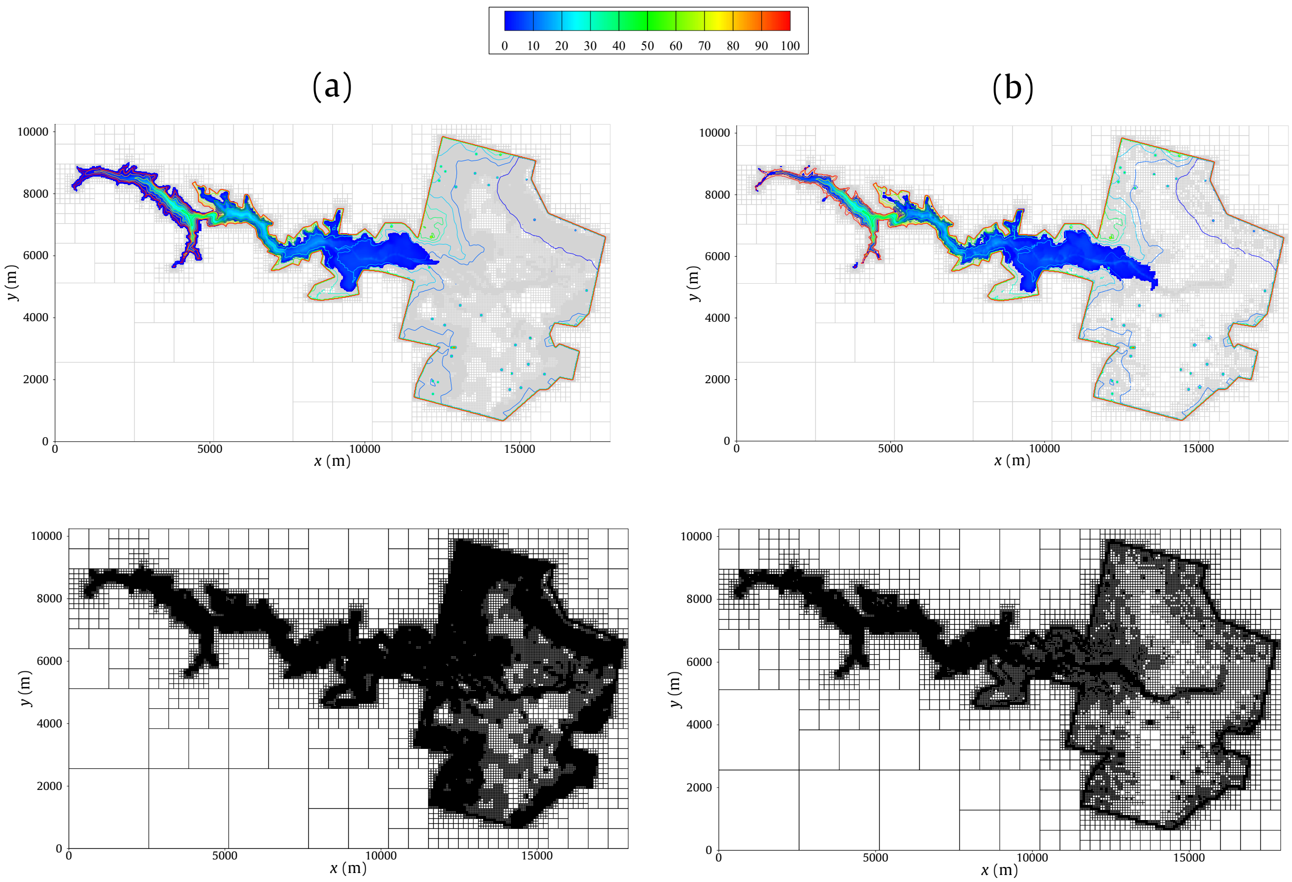

Figure 20. Malpasset dam-break. Flood map contours over topography (top) and adaptive grids (bottom) predicted by the adaptive solvers at $\varepsilon = 10^{-3}$ and for $t = 20\,\text{min}$: (a) HFV1 and (b) MWDG2.

middle of the valley (compare contour maps at the top of Fig. 19a and Fig. 19b). In terms of grid predictions, HFV1 with its piecewise-constant accuracy required fine resolution levels (bottom of Fig. 19a) along the valley to track the dynamics of the dam-break wave, and to capture the topographic features within the (dry) urban area. The piecewise-planar accuracy of MWDG2 allows it to capture these same features at a coarser resolution (bottom of Fig. 19b). At $\varepsilon = 10^{-3}$ (Fig. 20), the propagation of the dam-break wave along the river valley is predicted by HFV1 with almost similar accuracy as with HFV1 at $\varepsilon = 10^{-2}$ (compare the contour maps at the top of Fig. 19a and Fig. 20a). However, HFV1 induces a significant increase in refinement levels (compare the grids at the bottom of Fig. 19a and Fig. 20a). Similar behaviour is observed with MWDG2 (compare the grids at the bottom of Fig. 19b and Fig. 20b), which also leads to more refinement in the grid at $\varepsilon = 10^{-3}$. However, MWDG2 at $\varepsilon = 10^{-3}$ enables a much coarser resolution overall, in particular for the urban area while also being able to capture the topographic connectivities linking it to the valley at a fine resolution, thereby providing more detailed channeling of flow pathways on the floodplain. Because of this, MWDG2 at $\varepsilon = 10^{-3}$ predicted a more prolonged wetting extent than at $\varepsilon = 10^{-2}$. Overall, the piecewise-planar accuracy of MWDG2 enables more reliable predictions of arrival times and flooding extents than the piecewise-constant accuracy of HFV1, but at a much coarser resolution for the computational grid.

Table 3. Malpasset dam-break. Runtime ratios between the MWDG2, HFV1, DG2 and FV1 solvers analysed after completing a 40 min simulation at $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$ for the adaptive solvers.

Runtime ratios	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-3}$
DG2/FV1	11.62X	11.62X
FV1/HFV1	7.05X	3.34X
DG2/MWDG2	11.90X	7.08X
FV1/MWDG2	1.02X	0.60X

To explore the efficiency gain associated with the more aggressive coarsening ability of MWDG2, CPU runtimes to complete a 40 min simulation were calculated, together with those of the adaptive HFV1 solver and the uniform FV1 and DG2 solvers. Table 3 lists the runtime ratios of adaptive solvers at $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$ with respect to the uniform solvers. For this test, DG2 is found to be 11.6 times slower to run than FV1. Compared to the previous test (Section 2.3.1), where DG2 is 3.84 times slower to run than FV1, this test involves about 3 times the number of elements, which explains why DG2's runtime is much higher than FV1 for this test.

Consistent with the findings of the previous test (Table 2), HFV1 at $\varepsilon = 10^{-3}$ is found to be around 3 times more efficient than FV1, but experiences an increase in relative speedup to become 7 times faster at $\varepsilon = 10^{-2}$. In contrast, MWDG2 at $\varepsilon = 10^{-3}$ is here found to be only 7 times more efficient to run than the uniform DG2 solver (i.e. 30 times less relative to the previous test, Table 2), but experiences an increase in relative speedup to become 12 times faster at $\varepsilon = 10^{-2}$. The main reason for this relative loss in runtime efficiency can be attributed to the highly dynamic flow being analysed over the 40-min duration for this simulation, where the overhead cost of the MW-adaptivity in MWDG2 exceeds that of the wavelet-adaptivity in HFV1 (recall Fig. 8). Still, MWDG2 only demanded similar-to-double the runtime of the uniform FV1 solver (Table 3) to deliver similar predictive accuracy as the more expensive uniform DG2, making MWDG2 a feasible alternative to improve both accuracy and efficiency of Godunov-type hydrodynamic models.

4. Summary and conclusions

A second-order multiwavelet discontinuous Galerkin (MWDG2) solver was devised to robustly solve the two-dimensional (2D) depth-averaged shallow water equations (SWE) with bed-slope and friction source terms. The MWDG2 solver was founded upon a scaled version of the slope-decoupled DG2 solver (Section 2.1), which has been designed as an appropriate reference scheme that robustly harnesses the prowess of the multiresolution adaptivity of multiwavelets. This reference DG2 solver: (i) expands piecewise-planar solutions by 3 scale coefficients, or DG2 modes, representing an average and two directionally independent slopes; and, (ii) preserves well-balancedness in all 3 scale coefficients for flows over uneven terrain with wet-dry zones and fronts (Kesserwani et al., 2018).

Two alternative multiwavelet bases were proposed to form adaptive MWDG2 solvers that maintain compatibility with the slope-decoupled DG2 basis and readily preserve well-balancedness (Section 2.2). Approach 1 (Section 2.2.1) adopted the 2D tensor product of a 1D piecewise-linear basis and multiwavelet basis (Kesserwani et al., 2019). It relied on zeroing the 4th scale coefficient—associated with the nonlinear cross-dimensional slope—in the scaling basis (Eq. (24)), while retaining all 4 coefficients in the multiwavelet basis (Eq. (25)). Thereby, Approach 1 allows the reuse of simple 1D filter expressions for encoding and decoding 2D piecewise-planar solutions (Eqs. (26)-(27) and Appendix A). Approach 2 (Section 2.2.2) relied on constructing a 2D multiwavelet basis from the basis of the reference DG2 solver, both involving 3 coefficients (Eqs. (28)-(29)). Thereby, Approach 2 offered an alternative for encoding and decoding 2D piecewise-planar solutions (Eqs. (30)-(31) and Appendix A), but yielded more complicated 2D filter expressions.

With either Approach 1 or Approach 2, the adaptive multiresolution DG2 solution (Section 2.3) followed a similar procedure as the 1D case (Kesserwani et al., 2019): piecewise-planar flow and topography data were compressed into a single dataset of details, then significant details were identified by comparing their magnitude relative to a single error threshold value ε. Signficant details were then added to the coarsest piecewise-planar solution to assemble a multiresolution DG2 solution on a non-uniform grid, where local DG2 modes were evolved by applying the reference DG2 solver, harnessing the locally-planar solution and multiwavelet transformations to ensure flux conservation across faces between arbitrarily coarse- and fine-scale elements. A first-order variant was formed by adopting Haar wavelets with the piecewise-constant basis of a finite volume solver (HFV1), by reducing the complexity of the MWDG2 solver (Section 2.4).

The adaptive HFV1 and MWDG2 solvers were assessed across six well-known hydrodynamic tests to measure their performance relative to the uniform first-order finite volume (FV1) and DG2 solvers. Adaptive solver runs started from the coarsest grid with $M \times N$ baseline elements at a maximum refinement level L, allowing a maximum of 2^{2L} sub-elements per baseline element. The uniform solvers were run at the finest resolution accessible by their adaptive counterparts, on a grid of $M \times N \times 2^{2L}$ uniform elements. Simulations were conducted using the maximum stable Cr number: 0.5 for FV1 and HFV1, and 0.3 for DG2 and MWDG2. The accuracy, efficiency and robustness of the adaptive solvers were verified for three diagnostic tests (Section 3.1), and for three laboratory-scale tests (Section 3.2).

The results of the three diagnostic tests reinforce that an error threshold around $\varepsilon = 10^{-3}$ allows the HFV1 and MWDG2 solvers to deliver the accuracy of their uniform FV1 and DG2 solvers (Kesserwani et al., 2019), while being 26 and 140 times more efficient, respectively. Compared to their 1D counterparts, HFV1 and MWDG2 are found to be more efficient in 2D for the same test by around 13 and 7 times, respectively, even at $\varepsilon = 10^{-5}$ (Section 3.1.1). The results offer compelling evidence that the MWDG2 solver with both Approach 1 and Approach 2 robustly preserves the well-balancedness of its full 2D piecewise-planar solutions without any indication of momentum error through wet-dry fronts intersecting with steeply-sloping terrain (Section 3.1.3). Both Approach 1 and Approach 2 are found to be valid for use with the adaptive MWDG2 solver for the simulation of highly dynamic flow transients (Section 3.1.2). Nonetheless, Approach 1 is shown to benefit from larger time-steps and more aggressive grid coarsening for smooth and gradual flows typical of flood inundation (Section 3.1.1). As a result of its simplicity and flexibility, MWDG2 with Approach 1 was favoured for subsequent modelling of three laboratory-scale tests.

The laboratory-scale test results revealed that HFV1, with its piecewise-constant representation, can exhibit excessive grid refinement, particularly in cases with fast flow transitions where the runtime cost of HFV1 cost can be be dominated by wavelet-adaptivity overhead. However, HFV1 demonstrates potential benefits as an efficient alternative to FV1 for very slowly propagating flows over large spatial domains with predominantely dry initial conditions. In contrast, MWDG2's piecewise-planar handling of model data enables two advantages: (1) relevant features—such as flood wave arrival times, details of flow pathways, and wetting and drying extents—are captured at a similar level of accuracy to that of the expensive DG2 solver; and, (2) MWDG2 coarsens the grid more aggressively, making MWDG2 a competitive alternative to the less-accurate FV1 solver: MWDG2 is found to be 0.6 to 15 times faster than FV1 depending on the test case and choice of ε.

Overall, this work offers strong evidence that the proposed MWDG2 scheme with Approach 1 produces an adaptive Godunov-type hydrodynamic model that preserves the robustness and accuracy of the expensive uniform DG2 solver, for a substantially reduced runtime cost that is competitive with the commonly-used uniform FV1 solver.

5. Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

6. CRediT authorship contribution statement

Georges Kesserwani: Conceptualization, Investigation, Writing - original draft, review & editing, Supervision, Project administration, Funding acquisition. **Mohammad Kazem Sharifian**: Conceptualization, Methodology, Software, Investigation, Writing - review & editing.

Acknowledgments

This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/R007349/1. It is part of the SEAMLESS-WAVE (SoftwarE infrAstructure for Multi-purpose fLood modElling at variouS scaleS based on WAVElets) project, `https://www.seamlesswave.com`, where information on software availability can be found.

We wish to thank James Shaw (University of Sheffield), Jennifer Ryan (Colorado School of Mines), and Chris Keylock (Loughborough University) for their feedback on the methodology and approaches; and, Paul Bates (University of Bristol) and Brett Sanders (University of California, Irvine) for their feedback on the results and discussions.

Appendix A. Low-pass and high-pass 2D filters

In this section, the explicit formulae of the filters used in Sections 2.2.1 and 2.2.2 are presented. Considering the bases in Eqs. (16) and (19), the following relationships across two subsequent resolution levels (n) and $(n+1)$ can be obtained (Gerhard and Müller, 2016):

$$\hat{P}_{i,j}^{K,(n)} = \sum_{i',j'=0}^{2^n-1} \sum_{K'=0}^{p} \hat{P}_{i',j'}^{K',(n+1)} \left\langle \hat{P}_{i',j'}^{K',(n+1)}, \hat{P}_{i,j}^{K,(n)} \right\rangle \tag{A.1}$$

$$\psi_{i,j}^{K,(n)} = \sum_{i',j'=0}^{2^n-1} \sum_{K'=0}^{p} \hat{P}_{i',j'}^{K',(n+1)} \left\langle \hat{P}_{i',j'}^{K',(n+1)}, \psi_{i,j}^{K,(n)} \right\rangle \tag{A.2}$$

$$\hat{P}_{i',j'}^{K,(n+1)} = \sum_{K'=0}^{p} \hat{P}_{i,j}^{K',(n)} \left\langle \hat{P}_{i,j}^{K',(n)}, \hat{P}_{i',j'}^{K,(n+1)} \right\rangle + \sum_{K'=0}^{p} \psi_{i,j}^{K',(n)} \left\langle \psi_{i,j}^{K',(n)}, \hat{P}_{i',j'}^{K,(n+1)} \right\rangle \tag{A.3}$$

in which $\hat{P}$ and ψ are components of the 2D scaling and multiwavelet bases $\hat{\mathbf{P}}$ and $\mathbf{\Psi}$ used in either Approach 1 (Section 2.2.1) or (Section 2.2.2).

In Approach 1, $\hat{\mathbf{P}}$ and $\mathbf{\Psi}$ are 4-component bases obtained by 2D tensor products of 1D scaling and multiwavelet bases (i.e. similar to Eqs. (34)-(35) and (40)-(41) in Kesserwani et al., 2019). This leads to the following 1D low-pass filters $\mathbf{H}^0$ and $\mathbf{H}^1$ and high-pass filters $\mathbf{G}^0$ and $\mathbf{G}^1$:

$$\mathbf{H}^0 = \left\langle \hat{P}_{2i}^{K',(n+1)}, \hat{P}_{i}^{K,(n)} \right\rangle = \begin{bmatrix} 1/\sqrt{2} & 0 \\ -\sqrt{6}/4 & \sqrt{2}/4 \end{bmatrix} \quad \mathbf{H}^1 = \left\langle \hat{P}_{2i+1}^{K',(n+1)}, \hat{P}_{i}^{K,(n)} \right\rangle = \begin{bmatrix} 1/\sqrt{2} & 0 \\ \sqrt{6}/4 & \sqrt{2}/4 \end{bmatrix} \tag{A.4}$$

$$\mathbf{G}^0 = \left\langle \hat{P}_{2i}^{K',(n+1)}, \psi_{i}^{K,(n)} \right\rangle = \begin{bmatrix} 0 & -1/\sqrt{2} \\ \sqrt{2}/4 & \sqrt{6}/4 \end{bmatrix} \quad \mathbf{G}^1 = \left\langle \hat{P}_{2i+1}^{K',(n+1)}, \psi_{i}^{K,(n)} \right\rangle = \begin{bmatrix} 0 & 1/\sqrt{2} \\ -\sqrt{2}/4 & \sqrt{6}/4 \end{bmatrix} \tag{A.5}$$

By replacing the inner product terms in Eqs. (A.1) and (A.2) by the 1D filter matrices of Eqs. (A.4) and (A.5), and then applying the inner product $\mathbf{s}_{i,j}^{(n)} = \left\langle s, \hat{\mathbf{P}}_{i,j}^{(n)} \right\rangle$ operator on s_h of Eq. (20), Eq. (26) for encoding is obtained. Similarly, the decoding formula of Eq. (27) can be obtained by replacing the inner product terms in Eqs. (A.3) by the 1D filter matrices of Eqs. (A.4) and (A.5) and then applying the inner product $\mathbf{d}_{i,j}^{(n)} = \left\langle s, \mathbf{\Psi}_{i,j}^{(n)} \right\rangle$ operator on s_h.

In Approach 2, the multiwavelet bases, $\mathbf{\Psi}$, are made of 3 components constructed by applying the Gram-Schmidt process to the 3-component basis $\hat{\mathbf{P}}$. Following the process explained in Gerhard and Müller (2016), the final form of the low-pass filters $\mathbf{HH}^0$, $\mathbf{HH}^1$, $\mathbf{HH}^2$ and $\mathbf{HH}^3$ are:

$$\begin{aligned}
&\mathbf{HH}^0 = \begin{bmatrix} 1/2 & 0 & 0 \\ -\sqrt{3}/4 & 1/4 & 0 \\ -\sqrt{3}/4 & 0 & 1/4 \end{bmatrix} \quad \mathbf{HH}^1 = \begin{bmatrix} 1/2 & 0 & 0 \\ -\sqrt{3}/4 & 1/4 & 0 \\ \sqrt{3}/4 & 0 & 1/4 \end{bmatrix} \\
&\mathbf{HH}^2 = \begin{bmatrix} 1/2 & 0 & 0 \\ \sqrt{3}/4 & 1/4 & 0 \\ -\sqrt{3}/4 & 0 & 1/4 \end{bmatrix} \quad \mathbf{HH}^3 = \begin{bmatrix} 1/2 & 0 & 0 \\ \sqrt{3}/4 & 1/4 & 0 \\ \sqrt{3}/4 & 0 & 1/4 \end{bmatrix}
\end{aligned} \tag{A.6}$$

while the high-pass filters relative to the horizontal direction, $\mathbf{GA}^0$, $\mathbf{GA}^1$, $\mathbf{GA}^2$ and $\mathbf{GA}^3$ take the form:

$$\begin{aligned}
\mathbf{GA}^0 &= \begin{bmatrix} -\sqrt{14}/14 & -\sqrt{42}/14 & -\sqrt{42}/14 \\ 0 & 0 & -\sqrt{2}/2 \\ 0 & -\sqrt{2}/2 & 0 \end{bmatrix} \\
\mathbf{GA}^1 &= 0 \\
\mathbf{GA}^2 &= 0 \\
\mathbf{GA}^3 &= \begin{bmatrix} \sqrt{14}/14 & -\sqrt{42}/14 & -\sqrt{42}/14 \\ 0 & 0 & \sqrt{2}/2 \\ 0 & \sqrt{2}/2 & 0 \end{bmatrix}
\end{aligned} \tag{A.7}$$

For the vertical direction, the high-pass filters $\mathbf{GB}^0$, $\mathbf{GB}^1$, $\mathbf{GB}^2$ and $\mathbf{GB}^3$ read as:

$$\begin{aligned}
\mathbf{GB}^0 &= 0 \\
\mathbf{GB}^1 &= \begin{bmatrix} -\sqrt{14}/14 & -\sqrt{42}/14 & \sqrt{42}/14 \\ 0 & 0 & -\sqrt{2}/2 \\ 0 & -\sqrt{2}/2 & 0 \end{bmatrix} \\
\mathbf{GB}^2 &= \begin{bmatrix} \sqrt{14}/14 & -\sqrt{42}/14 & \sqrt{42}/14 \\ 0 & 0 & \sqrt{2}/2 \\ 0 & \sqrt{2}/2 & 0 \end{bmatrix} \\
\mathbf{GB}^3 &= 0
\end{aligned} \tag{A.8}$$

and finally, for the diagonal direction $\mathbf{GC}^0$, $\mathbf{GC}^1$, $\mathbf{GC}^2$ and $\mathbf{GC}^3$ would be formed as:

$$\mathbf{GC}^0 = \begin{bmatrix} 1/2 & 0 & 0 \\ -\sqrt{21}/28 & -3\sqrt{7}/28 & 2\sqrt{7}/14 \\ -\sqrt{21}/28 & 2\sqrt{7}/14 & -3\sqrt{7}/28 \end{bmatrix}$$

$$\mathbf{GC}^1 = \begin{bmatrix} -1/2 & 0 & 0 \\ -\sqrt{21}/28 & -3\sqrt{7}/28 & -2\sqrt{7}/14 \\ \sqrt{21}/28 & -2\sqrt{7}/14 & -3\sqrt{7}/28 \end{bmatrix}$$

$$\mathbf{GC}^2 = \begin{bmatrix} -1/2 & 0 & 0 \\ \sqrt{21}/28 & -3\sqrt{7}/28 & -2\sqrt{7}/14 \\ -\sqrt{21}/28 & -2\sqrt{7}/14 & -3\sqrt{7}/28 \end{bmatrix} \tag{A.9}$$

$$\mathbf{GC}^3 = \begin{bmatrix} 1/2 & 0 & 0 \\ \sqrt{21}/28 & -3\sqrt{7}/28 & 2\sqrt{7}/14 \\ \sqrt{21}/28 & 2\sqrt{7}/14 & -3\sqrt{7}/28 \end{bmatrix}$$

which are the expressions of the 2D filters used in Eqs. (30) and (31) for encoding and decoding, respectively.

Appendix B. Instructions for running the FV1, DG2, HFV1 and MWDG2 solvers

The MWDG2 and HFV1 codes are available to download from Zenodo (Sharifian and Kesserwani, 2020). They are implemented in Fortran 2003 and can be compiled using Intel Fortran Compiler in both Windows and Linux. Other Fortran compilers have not been tested. Windows users can simply add the source files to the project created for Microsoft Visual Studio or any other IDE. Linux users can use the included makefile to compile the codes. The user can configure the simulations by modifying `config.dat` input file. The codes are preconfigured for the three test cases of Section 3.1, and the test cases of Section 3.2 can be configured manually within the same input file and by loading their DEM datasets that are also provided.

References

Acuña MA, Aoki T. Tree-based mesh-refinement GPU-accelerated tsunami simulator for real-time operation. Natural Hazards and Earth System Sciences 2018;18(9):2561–602. doi:10.5194/nhess-18-2561-2018.

Alpert BK. A Class of Bases in L^2 for the Sparse Representation of Integral Operators. SIAM Journal on Mathematical Analysis 1993;doi:10.1137/0524016.

Archibald R, Fann G, Shelton W. Adaptive discontinuous Galerkin methods in multiwavelets bases. Applied Numerical Mathematics 2011;61(7):879–90. URL: http://linkinghub.elsevier.com/retrieve/pii/

S0168927411000389https://linkinghub.elsevier.com/retrieve/pii/S0168927411000389. doi:10.1016/j.apnum.2011.02.005.

Begnudelli L, Sanders BF, Bradford SF. Adaptive Godunov-Based Model for Flood Simulation. Journal of Hydraulic Engineering 2008;134(6):714–25. URL: http://ascelibrary.org/doi/10.1061/{%}28ASCE{%}290733-9429{%}282008{%}29134{%}3A6{%}28714{%}29. doi:10.1061/(ASCE)0733-9429(2008)134:6(714).

Briggs MJ, Synolakis CE, Harkins GS, Green DR. Laboratory experiments of tsunami runup on a circular island. Pure and applied geophysics 1995;144(3-4):569–93.

Brufau P, García-Navarro P, Vázquez-Cendón ME. Zero mass error using unsteady wetting-drying conditions in shallow flows over dry irregular topography. International Journal for Numerical Methods in Fluids 2004;45(10):1047–82. doi:10.1002/fld.729.

Buttinger-Kreuzhuber A, Horváth Z, Noelle S, Blöschl G, Waser J. A fast second-order shallow water scheme on two-dimensional structured grids over abrupt topography. Advances in Water Resources 2019;127:89 – 108. URL: http://www.sciencedirect.com/science/article/pii/S0309170818305335. doi:https://doi.org/10.1016/j.advwatres.2019.03.010.

Caviedes-Voullième D, Kesserwani G. Benchmarking a multiresolution discontinuous Galerkin shallow water model: Implications for computational hydraulics. Advances in Water Resources 2015;86:14–31. URL: http://linkinghub.elsevier.com/retrieve/pii/S0309170815002237. doi:10.1016/j.advwatres.2015.09.016.

Caviedes-Voullième D, Gerhard N, Sikstel A, Müller S. Multiwavelet-based mesh adaptivity with Discontinuous Galerkin schemes: Exploring 2D shallow water problems. Advances in Water Resources 2020;138:103559. URL: http://www.sciencedirect.com/science/article/pii/S0309170819309121. doi:https://doi.org/10.1016/j.advwatres.2020.103559.

Cockburn B, Shu CW. Runge-Kutta Discontinuous Galerkin methods for convection-dominated problems. Journal of Scientific Computing 2001;16(3):173–261. URL: https://www.scopus.com/inward/record.uri?eid=2-s2.0-0000076512&partnerID=40&md5=df96bdc9e32a788a36e1eea7155b362c; cited By 1098.

Cohen A, Kaber SM, Müller S, Postel M. Fully adaptive multiresolution finite volume schemes for conservation laws. Mathematics of Computation 2001;72(241):183–226. URL: http://www.ams.org/journal-getitem?pii=S0025-5718-01-01391-6. doi:10.1090/S0025-5718-01-01391-6.

Dahmen W. Wavelet and multiscale methods for operator equations. Acta Numerica 1997;6(2):55–228. URL: https://www.cambridge.org/core/product/identifier/S0962492900002713/type/journal{_}article. doi:10.1017/S0962492900002713.

Dahmen W, Hovhannisyan N, Müller S. Adaptive Multiscale Methods for Flow Problems: Recent Developments. In: Summary of Flow Modulation and Fluid-Structure Interaction Findings. Springer, Berlin, Heidelberg; 2010. p. 77–103. URL: http://link.springer.com/10.1007/978-3-642-04088-7{_}4. doi:10.1007/978-3-642-04088-7_4.

Daubechies I. Ten Lectures on Wavelets. Society for Industrial and Applied Mathematics, 1992. URL: https://epubs.siam.org/doi/abs/10.1137/1.9781611970104. doi:10.1137/1.9781611970104. arXiv:https://epubs.siam.org/doi/pdf/10.1137/1.9781611970104.

Dawson C, Trahan CJ, Kubatko EJ, Westerink JJ. A parallel local timestepping runge-kutta discontinuous Galerkin method with applications to coastal ocean modeling. Computer Methods in Applied Mechanics and Engineering 2013;259:154–65. URL: http://dx.doi.org/10.1016/j.cma.2013.03.015. doi:10.1016/j.cma.2013.03.015.

Delis AI, Nikolos IK, Kazolea M. Performance and Comparison of Cell-Centered and Node-Centered Unstructured Finite Volume Discretizations for Shallow Water Free Surface Flows. Archives of Computational Methods in Engineering 2011;18(1):57–118. doi:10.1007/s11831-011-9057-6.

Díaz Calle JL, Devloo PRB, Gomes SM. Wavelets and adaptive grids for the discontinuous Galerkin method. Numerical Algorithms 2005;39(1):143–54. URL: https://doi.org/10.1007/s11075-004-3626-9. doi:10.1007/s11075-004-3626-9.

Echeverribar I, Morales-Hernández M, Brufau P, García-Navarro P. 2D numerical simulation of unsteady flows for large scale floods prediction in real time. Advances in Water Resources 2019;134(June). doi:10.1016/j.advwatres.2019.103444.

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). International Journal for Numerical Methods in Fluids 2011;66(8):1000–18. URL: http://doi.wiley.com/10.1002/fld.229http://doi.wiley.com/10.1002/fld.2298. doi:10.1002/fld.2298.

Gerhard N, Caviedes-Voullième D, Müller S, Kesserwani G. Multiwavelet-based grid adaptation with discontinuous Galerkin schemes for shallow water equations. Journal of Computational Physics 2015a;301:265 –88. URL: http://www.sciencedirect.com/science/article/pii/S0021999115005574. doi:https://doi.org/10.1016/j.jcp.2015.08.030.

Gerhard N, Iacono F, May G, Müller S, Schäfer R. A High-Order Discontinuous Galerkin Discretization with Multiwavelet-Based Grid Adaptation for Compressible Flows. Journal of Scientific Computing 2015b;62(1):25–52. URL: http://link.springer.com/10.1007/s10915-014-9846-9. doi:10.1007/s10915-014-9846-9.

Gerhard N, Müller S. Adaptive multiresolution discontinuous Galerkin schemes for conservation laws: multi-dimensional case. Computational and Applied Mathematics 2016;35(2):321–49.

Glenis V, Kutija V, Kilsby CG. A fully hydrodynamic urban flood modelling system representing buildings, green space and interventions. Environmental Modelling and Software 2018;109(September 2017):272–92. URL: https://doi.org/10.1016/j.envsoft.2018.07.018. doi:10.1016/j.envsoft.2018.07.018.

Gourgue O, Comblen R, Lambrechts J, Kärnä T, Legat V, Deleersnijder E. A flux-limiting wetting-drying method for finite-element shallow-water models, with application to the Scheldt Estuary. Advances in Water Resources 2009;32(12):1726–39. URL: http://dx.doi.org/10.1016/j.advwatres.2009.09.005. doi:10.1016/j.advwatres.2009.09.005.

Goutal N. The Malpasset dam failure. An overview and test case definition. In: Proceedings of the Fourth CADAM Meeting. 1999. p. 1–8.

Guermond JL, de Luna MQ, Popov B, Kees CE, Farthing MW. Well-Balanced Second-Order Finite Element Approximation of the Shallow Water Equations with Friction. SIAM Journal on Scientific Computing 2018;40(6):A3873–901. URL: https://epubs.siam.org/doi/10.1137/17M1156162. doi:10.1137/17M1156162.

Guinot V, Delenne C, Rousseau A, Boutron O. Flux closures and source term models for shallow water models with depth-dependent integral porosity. Advances in Water Resources 2018;122(January):1–26. URL: https://doi.org/10.1016/j.advwatres.2018.09.014. doi:10.1016/j.advwatres.2018.09.014.

Guo W, Cheng Y. A Sparse Grid Discontinuous Galerkin Method for High-Dimensional Transport Equations and Its Application to Kinetic Simulations. SIAM Journal on Scientific Computing 2016;38(6):A3381–409. URL: http://epubs.siam.org/doi/10.1137/16M1060017. doi:10.1137/16M1060017.

Haleem DA, Kesserwani G, Caviedes-Voullième D. Haar wavelet-based adaptive finite volume shallow water solver. Journal of Hydroinformatics 2015;17(6):857–73. URL: https://iwaponline.com/jh/article/17/6/857-873/3355. doi:10.2166/hydro.2015.039.

Hervouet JM, Petitjean A. Malpasset dam-break revisited with two-dimensional computations. Journal of Hydraulic Research 1999;37(6):777–88. doi:10.1080/00221689909498511.

Hou J, Wang R, Liang Q, Li Z, Huang MS, Hinkelmann R. Efficient surface water flow simulation on static Cartesian grid with local refinement according to key topographic features. Computers and Fluids 2018;176:117–34. URL: https://doi.org/10.1016/j.compfluid.2018.03.024. doi:10.1016/j.compfluid.2018.03.024.

Hovhannisyan N, Müller S, Schäfer R. Adaptive multiresolution discontinuous Galerkin schemes for conservation laws. Mathematics of Computation 2013;83(285):113–51. URL: http://www.ams.org/jourcgi/jour-getitem?pii=S0025-5718-2013-02732-9. doi:10.1090/S0025-5718-2013-02732-9.

Jarczewska K, Glabisz W, Zielichowski-Haber W. Multiwavelets and multiwavelet packets of Legendre functions in the direct method for solving variational problems. Archives of Civil and Mechanical Engineering 2015;15(1):1 – 10. URL: http://www.sciencedirect.com/science/article/pii/S1644966514000739. doi:https://doi.org/10.1016/j.acme.2014.04.008.

Kesserwani G. Topography discretization techniques for Godunov-type shallow water numerical models: a comparative study. Journal of Hydraulic Research 2013;51(4):351–67. URL: https://doi.org/10.1080/00221686.2013.796574. doi:10.1080/00221686.2013.796574. arXiv:https://doi.org/10.1080/00221686.2013.796574.

Kesserwani G, Ayog JL, Bau D. Discontinuous Galerkin formulation for 2D hydrodynamic modelling: Trade-offs between theoretical complexity and practical convenience. Computer Methods in Applied Mechanics and Engineering 2018;342:710 –41. URL: http://www.sciencedirect.com/science/article/pii/S004578251830389X. doi:https://doi.org/10.1016/j.cma.2018.08.003.

Kesserwani G, Caviedes-Voullième D, Gerhard N, Müller S. Multiwavelet discontinuous Galerkin h-adaptive shallow water model. Computer Methods in Applied Mechanics and Engineering 2015;294:56–71. doi:10.1016/j.cma.2015.05.016.

Kesserwani G, Liang Q. A discontinuous Galerkin algorithm for the two-dimensional shallow water equations. Computer Methods in Applied Mechanics and Engineering 2010;199(49):3356 –68. URL: http://www.sciencedirect.com/science/article/pii/S0045782510002197. doi:https://doi.org/10.1016/j.cma.2010.07.007.

Kesserwani G, Liang Q. Dynamically adaptive grid based discontinuous Galerkin shallow water model. Advances in Water Resources 2012a;37:23 – 39. URL: http://www.sciencedirect.com/science/article/pii/S0309170811002181. doi:https://doi.org/10.1016/j.advwatres.2011.11.006.

Kesserwani G, Liang Q. Locally limited and fully conserved RKDG2 shallow water solutions with wetting and drying. Journal of scientific computing 2012b;50(1):120–44.

Kesserwani G, Shaw J, Sharifian MK, Bau D, Keylock CJ, Bates PD, Ryan JK. (Multi)wavelets increase both accuracy and efficiency of standard Godunov-type hydrodynamic models. Advances in Water Resources 2019;129:31 – 55. URL: http://www.sciencedirect.com/science/article/pii/S0309170819301770. doi:https://doi.org/10.1016/j.advwatres.2019.04.019.

Kesserwani G, Wang Y. Discontinuous Galerkin flood model formulation: Luxury or necessity? Water Resources Research 2014;50(8):6522–41. URL: http://doi.wiley.com/10.1002/2013WR014906. doi:10.1002/2013WR014906.

Krivodonova L, Xin J, Remacle JF, Chevaugeon N, Flaherty J. Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. Applied Numerical Mathematics 2004;48(3):323 –38. URL: http://www.sciencedirect.com/science/article/pii/S0168927403001831. doi:https://doi.org/10.1016/j.apnum.2003.11.002; workshop on Innovative Time Integrators for PDEs.

Lamby P, Müller S, Stiriba Y. Solution of shallow water equations using fully adaptive multiscale schemes. International Journal for Numerical Methods in Fluids 2005;49(4):417–37. URL: http://doi.wiley.com/10.1002/fld.1004. doi:10.1002/fld.1004. arXiv:arXiv:1011.1669v3.

Lannes D, Marche F. A new class of fully nonlinear and weakly dispersive Green–Naghdi models for efficient 2D simulations. Journal of Computational Physics 2015;282:238 –68. URL: http://www.sciencedirect.com/science/article/pii/S0021999114007748. doi:https://doi.org/10.1016/j.jcp.2014.11.016.

Lhomme J, Gutierrez-Andres J, Weisgerber A, Davison M, Mulet-Marti J, Cooper A, Gouldby B. Testing a new two-dimensional flood modelling system: analytical tests and application to a flood event. Journal of Flood Risk Management 2010;3(1):33–51. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1753-318X.2009.01053.x. doi:10.1111/j.1753-318X.2009.01053.x. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1753-318X.2009.01053.x.

Li S. Comparison of refinement criteria for structured adaptive mesh refinement. Journal of Computational and Applied Mathematics 2010;233(12):3139 –47. URL: http://www.sciencedirect.com/science/article/pii/S037704270900586X. doi:https://doi.org/10.1016/j.cam.2009.08.104; finite Element Methods in Engineering and Science (FEMTEC 2009).

Liang D, Lin B, Falconer RA. A boundary-fitted numerical model for flood routing with shock-capturing capability. Journal of Hydrology 2007;332(3-4):477–86. doi:10.1016/j.jhydrol.2006.08.002.

Liang Q. A simplified adaptive Cartesian grid system for solving the 2D shallow water equations. International Journal for Numerical Methods in Fluids 2012;69(2):442–58. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.2568. doi:10.1002/fld.2568. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.2568.

Liang Q, Hou J, Xia X. Contradiction between the C-property and mass conservation in adaptive grid based shallow flow models: cause and solution. International Journal for Numerical Methods in Fluids 2015;78(1):17–36. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4005. doi:10.1002/fld.4005. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.4005.

Liang Q, Marche F. Numerical resolution of well-balanced shallow water equations with complex source terms. Advances in Water Resources 2009;32(6):873 –84. URL: http://www.sciencedirect.com/science/article/pii/S0309170809000396. doi:https://doi.org/10.1016/j.advwatres.2009.02.010.

Liu PLF, Cho YS, Briggs MJ, Kanoglu U, Synolakis CE. Runup of solitary waves on a circular Island. Journal of Fluid Mechanics 1995;302:259–285. doi:10.1017/S0022112095004095.

Ma Z, Zhou T, Sun J, Zhai G. Simulation on tsunami-like solitary wave run-up around a conical island using a modified mass source method. Engineering Applications of Computational Fluid Mechanics 2019;13(1):849–59. URL: https://doi.org/10.1080/19942060.2019.1642960. doi:10.1080/19942060.2019.1642960. arXiv:https://doi.org/10.1080/19942060.2019.1642960.

Matsuyama M, Tanaka H. An experimental study of the highest run-up height in the 1993 Hokkaido Nansei-oki earthquake tsunami. In: National Tsunami Hazard Mitigation Program Review and International Tsunami Symposium (ITS). 2001. p. 879–89.

Mignot E, Paquier A, Haider S. Modeling floods in a dense urban area using 2D shallow water equations. Journal of Hydrology 2006;327(1-2):186–99. doi:10.1016/j.jhydrol.2005.11.026.

Minatti L, Cicco PND, Solari L. Second order discontinuous Galerkin scheme for compound natural channels with movable bed. Applications for the computation of rating curves. Advances in Water Resources 2016;93:89 – 104. URL: http://www.sciencedirect.com/science/article/pii/S0309170815001335. doi:https://doi.org/10.1016/j.advwatres.2015.06.007; numerical modelling of river morphodynamics.

Morales-Hernández M, Sharif MB, Gangrade S, Dullo TT, Kao SC, Kalyanapu A, Ghafoor SK, Evans KJ, Madadi-Kandjani E, Hodges BR. High-performance computing in water resources hydrodynamics. Journal of Hydroinformatics 2020;URL: https://doi.org/10.2166/hydro.2020.163. doi:10.2166/hydro.2020.163. arXiv:https://iwaponline.com/jh/article-pdf/doi/10.2166/hydro.2020.163/667135/jh2020163.pdf; jh2020163.

Müller S. Adaptive Multiscale Schemes for Conservation Laws. volume 27 of *Lecture Notes in Computational Science and Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. URL: http://link.springer.com/10.1007/978-3-642-18164-1. doi:10.1007/978-3-642-18164-1.

Néelz S, Pender G. Benchmarking the latest generation of 2D hydraulic modelling packages. Environment Agency, Horison House, Deanery Road, Bristol, BS1 9AH 2013;.

Popinet S. Quadtree-adaptive tsunami modelling. Ocean Dynamics 2011;61(9):1261–85. URL: http://link.springer.com/10.1007/s10236-011-0438-z. doi:10.1007/s10236-011-0438-z.

Qin X, LeVeque RJ, Motley MR. Accelerating an Adaptive Mesh Refinement Code for Depth-Averaged Flows Using GPUs. Journal of Advances in Modeling Earth Systems 2019;11(8):2606–28. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019MS001635. doi:10.1029/2019MS001635. arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019MS001635.

Sanders BF, Schubert JE. PRIMo: Parallel raster inundation model. Advances in Water Resources 2019;126(February):79–95. URL: https://doi.org/10.1016/j.advwatres.2019.02.007. doi:10.1016/j.advwatres.2019.02.007.

Sharifian MK, Hassanzadeh Y, Kesserwani G, Shaw J. Performance study of the multiwavelet discontinuous Galerkin approach for solving the Green-Naghdi equations. International Journal for Numerical Methods in Fluids 2019;90(10):501–21. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4732. doi:10.1002/fld.4732.

Sharifian MK, Kesserwani G. Two-dimensional fortran 2003 finite volume and discontinuous galerkin hydrodynamic models with (multi)wavelet-based adaptivity 2020;doi:10.5281/zenodo.3941723.

Shi Ye, Nguyen KD. A projection method-based model for dam- and dyke-break flows using an unstructured finite-volume technique: Applications to the Malpasset dam break (France) and to the flood diversion in the Red River Basin (Vietnam). International Journal for Numerical Methods in Fluids 2008;56(8):1505–12. URL: http://doi.wiley.com/10.1002/fld.1699. doi:10.1002/fld.1699.

Smith CW, Zang J, Eatock Taylor R. Wavelet-based adaptive grids as applied to hydrodynamics. International Journal for Numerical Methods in Fluids 2008;57(7):877–903. URL: http://doi.wiley.com/10.1002/fld.1657. doi:10.1002/fld.1657.

Tao Z, Guo W, Cheng Y. Sparse grid discontinuous Galerkin methods for the Vlasov-Maxwell system. Journal of Computational Physics: X 2019;3(558704):100022. URL: https://doi.org/10.1016/j.jcpx.2019.100022. doi:10.1016/j.jcpx.2019.100022. arXiv:1805.10747.

Teng J, Jakeman AJ, Vaze J, Croke BF, Dutta D, Kim S. Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. Environmental Modelling and Software 2017;90:201–16. URL: http://dx.doi.org/10.1016/j.envsoft.2017.01.006. doi:10.1016/j.envsoft.2017.01.006.

Tonelli M, Petti M. Finite volume scheme for the solution of 2D extended Boussinesq equations in the surf zone. Ocean Engineering 2010;37(7):567 –82. URL: http://www.sciencedirect.com/science/article/pii/S0029801810000405. doi:https://doi.org/10.1016/j.oceaneng.2010.02.004.

Toro EF. Shock-capturing methods for free-surface shallow flows. Wiley, 2001.

Toro EF, Garcia-Navarro P. Godunov-type methods for free-surface shallow flows: A review. Journal of Hydraulic Research 2007;45(6):736–51. URL: https://doi.org/10.1080/00221686.2007.9521812. doi:10.1080/00221686.2007.9521812. arXiv:https://doi.org/10.1080/00221686.2007.9521812.

Trahan CJ, Dawson C. Local time-stepping in Runge-Kutta discontinuous Galerkin finite element methods applied to the shallow-water equations. Computer Methods in Applied Mechanics and Engineering 2012;217-220:139–52. URL: http://dx.doi.org/10.1016/j.cma.2012.01.002. doi:10.1016/j.cma.2012.01.002.

Valiani A, Begnudelli L. Divergence Form for Bed Slope Source Term in Shallow Water Equations. Journal of Hydraulic Engineering 2006;132(7):652–65. doi:10.1061/(ASCE)0733-9429(2006)132:7(652).

Valiani A, Caleffi V, Zanni A. Case study: Malpasset dam-break simulation using a two-dimensional finite volume method. Journal of Hydraulic Engineering 2002;128(5):460–72. doi:10.1061/(ASCE)0733-9429(2002)128:5(460).

Vater S, Beisiegel N, Behrens J. A limiter-based well-balanced discontinuous Galerkin method for shallow-water flows with wetting and drying: Triangular grids. International Journal for Numerical Methods in Fluids 2019;91(8):395–418. doi:10.1002/fld.4762. arXiv:1811.09505.

Vuik MJ, Ryan JK. Multiwavelet troubled-cell indicator for discontinuity detection of discontinuous Galerkin schemes. Journal of Computational Physics 2014;270:138 –60. URL: http://www.sciencedirect.com/science/article/pii/S0021999114002344. doi:https://doi.org/10.1016/j.jcp.2014.03.047.

Wang Y, Liang Q, Kesserwani G, Hall JW. A 2D shallow flow model for practical dam-break simulations. Journal of Hydraulic Research 2011;49(3):307–16. doi:10.1080/00221686.2011.566248.

Wang Z, Tang Q, Guo W, Cheng Y. Sparse grid discontinuous Galerkin methods for high-dimensional elliptic equations. Journal of Computational Physics 2016;314:244–63. URL: http://dx.doi.org/10.1016/j.jcp.2016.03.005. doi:10.1016/j.jcp.2016.03.005. arXiv:1508.07781.

Xia X, Liang Q. A new efficient implicit scheme for discretising the stiff friction terms in the shallow water equations. Advances in Water Resources 2018;117:87 – 97. URL: http://www.sciencedirect.com/science/article/pii/S0309170818302124. doi:https://doi.org/10.1016/j.advwatres.2018.05.004.

Xia X, Liang Q, Ming X. A full-scale fluvial flood modelling framework based on a high-performance integrated hydrodynamic modelling system (HiPIMS). Advances in Water Resources 2019;132(March). doi:10.1016/j.advwatres.2019.103392.

Özgen Xian I, Kesserwani G, Caviedes-Voullième D, Molins S, Xu Z, Dwivedi D, Moulton JD, Steefel CI. Wavelet-based local mesh refinement for rainfall–runoff simulations. Journal of Hydroinformatics 2020;URL: https://doi.org/10.2166/hydro.2020.198. doi:10.2166/hydro.2020.198. arXiv:https://iwaponline.com/jh/article-pdf/doi/10.2166/hydro.2020.198/681917/jh2020198.pdf; jh2020198.

Zhao J, Özgen-Xian I, Liang D, Wang T, Hinkelmann R. An improved multislope MUSCL scheme for solving shallow water equations on unstructured grids. Computers and Mathematics with Applications 2019;77(2):576–96. URL: https://doi.org/10.1016/j.camwa.2018.09.059. doi:10.1016/j.camwa.2018.09.059.

Zhou F, Chen G, Huang Y, Yang JZ, Feng H. An adaptive moving finite volume scheme for modeling flood inundation over dry and complex topography. Water Resources Research 2013;49(4):1914–28. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/wrcr.20179. doi:10.1002/wrcr.20179. arXiv:https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/wrcr.20179.