Article:

Kesserwani, G., Shaw, J. orcid.org/0000-0002-0928-3604, Sharifian, M. et al. (4 more authors) (2019) (Multi)wavelets increase both accuracy and efficiency of standard Godunov-type hydrodynamic models. Advances in Water Resources, 129. pp. 31-55. ISSN 0309-1708

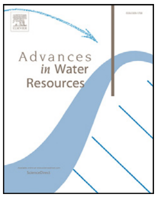
https://doi.org/10.1016/j.advwatres.2019.04.019

Advances in Water Resources 129 (2019) 31–55

Contents lists available at ScienceDirect

Advances in Water Resources

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



(Multi)wavelets increase both accuracy and efficiency of standard Godunov-type hydrodynamic models

Georges Kesserwani[a], James Shaw[a,*], Mohammad K Sharifian[b], Domenico Bau[a], Christopher J Keylock[c], Paul D Bates[d], Jennifer K Ryan[e]

[a] *Department of Civil and Structural Engineering, University of Sheffield, Sheffield S1 3JD, UK*
[b] *Department of Civil Engineering, University of Tabriz, Tabriz, Iran*
[c] *School of Architecture, Building and Civil Engineering, Loughborough University, Loughborough LE11 3TU, UK*
[d] *School of Geographical Science, University of Bristol, Bristol BS8 1BF, UK*
[e] *School of Mathematics, East Anglia University, Norwick NR4 7TJ, UK*

ARTICLE INFO

Keywords:
Adaptive multiresolution schemes
Scaled discontinuous Galerkin and finite volume hydraulic models
(Multi)wavelet data (de)compression
Performance comparisons

ABSTRACT

This paper presents a scaled reformulation of a robust second-order Discontinuous Galerkin (DG2) solver for the Shallow Water Equations (SWE), with guiding principles on how it can be naturally extended to fit into the multiresolution analysis of multiwavelets (MW). Multiresolution analysis applied to the flow and topography data enables the creation of an adaptive MWDG2 solution on a non-uniform grid. The multiresolution analysis also permits control of the adaptive model error by a single user-prescribed parameter. This results in an adaptive MWDG2 solver that can fully exploit the local (de)compression of piecewise-linear modelled data, and from which a first-order finite volume version (FV1) is directly obtainable based on the Haar wavelet (HFV1) for local (de)compression of piecewise-constant modelled data. The behaviour of the adaptive HFV1 and MWDG2 solvers is systematically studied on a number of well-known hydraulic tests that cover all elementary aspects relevant to accurate, efficient and robust modelling. The adaptive solvers are run starting from a baseline mesh with a single element, and their accuracy and efficiency are measured referring to standard FV1 and DG2 simulations on the uniform grid involving the finest resolution accessible by the adaptive solvers. Our findings reveal that the MWDG2 solver can achieve the same accuracy as the DG2 solver but with a greater efficiency than the FV1 solver due to the smoothness of its piecewise-linear basis, which enables more aggressive coarsening than with the piecewise-constant basis in the HFV1 solver. This suggests a great potential for the MWDG2 solver to efficiently handle the depth and breadth in resolution variability, while also being a multiresolution mesh generator. Accompanying model software and simulation data are openly available online.

1. Introduction

Explicit Godunov-type finite volume schemes (Toro and Garcia-Navarro, 2007) have become standard in hydraulic models (Teng et al., 2017). In essence, the Finite Volume (FV) foundation uses a piecewise-constant representation of flow variables over a local mesh element in a first-order accurate framework (FV1). Piecewise-constant data can be evolved element-wise driven by spatial flux exchange through element boundaries, while only needing data from adjacent neighbours to complete Riemann flux calculations. This locality in storage and evolution of piecewise constant data offers practical advantages such as suitability for parallelisation (Lacasta et al., 2013, Sanders et al., 2010) and makes wetting and drying a lot easier to handle (Hou et al., 2013, Medeiros and Hagen, 2013). However, the FV1 approach suffers from excessive numerical diffusion, which can only be alleviated by using fine resolution meshes, often leading to unacceptable computational costs and meshing inflexibilities over large spatial domains. Attempts to incorporate classical adaptive mesh refinement strategies within the FV1 approach are shown to cause adverse effects, such as keeping a coarsest mesh resolution that is fine enough, increasing model sensitivity to tuning many adaptivity parameters, and impacting overall conservativeness (Zhou et al., 2013, Kesserwani and Liang, 2015, Liang et al., 2015, Donat et al., 2014). These adverse effects are not alleviated with higher-order FV methods that involve non-local interpolation of piecewise-constant

[*] Corresponding author.
E-mail addresses: g.kesserwani@shef.ac.uk (G. Kesserwani), js102@zepler.net (J. Shaw), mksharifian@gmail.com (M.K. Sharifian), d.bau@sheffield.ac.uk (D. Bau), c.j.keylock@lboro.ac.uk (C.J. Keylock), paul.bates@bristol.ac.uk (P.D. Bates), jennifer.ryan@uea.ac.uk (J.K. Ryan).

https://doi.org/10.1016/j.advwatres.2019.04.019
Received 27 February 2019; Received in revised form 26 April 2019; Accepted 29 April 2019
Available online 30 April 2019

data (Li, 2010, An and Yu, 2014). A numerical modelling strategy is still desired that can inherently automate and initialise mesh resolution and improve runtime efficiency within the FV1 approach.

The Discontinuous Galerkin (DG) method extends the foundation of the FV1 approach by shaping local piecewise-polynomial solutions from a discrete (element-wise) formulation of the conservative model equation(s). DG methods significantly reduce numerical diffusion even on very coarse meshes (e.g. at a grid resolution exceeding 10 m^2) and have excellent conservation properties (Minatti et al., 2016, Kesserwani, 2013, Kesserwani and Wang, 2014, Bokhove, 2005). Compared to a FV counterpart, the DG method has a much larger cost per mesh element in terms of data storage and computing time, and such cost is proportional to the desired order-of-accuracy. Even with a simplified second-order DG (DG2) method for practical conveniences (Kesserwani et al., 2018), runtime costs on uniform meshes are 7-15 times greater than with first- and second-order accurate FV alternatives (Kesserwani, 2013, Kesserwani and Wang, 2014). Classical adaptive mesh refinement strategies with DG methods do not seem a practical way forward because they still suffer from many of the adverse effects reported for the FV1 method (Kesserwani and Liang, 2015, Kesserwani and Liang, 2012). A *sparse* numerical modelling strategy, which can make DG2 as efficient as FV1, is thus highly desired to increase accuracy and coverage in handling high-resolution modelled data.

Adaptive wavelet-based schemes offer an attractive route to overcome many of the adverse effects observed in classical adaptive mesh refinement methods (Liang et al., 2015, Donat et al., 2014, Li, 2010, An and Yu, 2014, Kesserwani and Liang, 2012, Liang and Borthwick, 2009). When applied to the reformulation of FV1 models, these schemes introduce a multiresolution analysis to (de)compress piecewise-constant modelled data mapped by the Haar wavelet from within the local basis of the FV1 method (Harten, 1995, Cohen et al., 2003, Müller, 2002, Minbashian et al., 2017, Archibald et al., 2011, Wang et al., 2016). We term this Haar-wavelet variant of FV1 the HFV1 method. Haleem et al. (2015) were the first to propose an HFV1 approach for solving the shallow water equations (SWE) with irregular topography and wet-dry fronts, demonstrating that HFV1 directly inherits the robustness properties of the underlying FV1 scheme. However, Haleem et al. (2015) did not fully leverage the local (de)compressibility property of wavelets. Instead, their HFV1 approach retained some of the aforementioned adverse effects, by still relying on an extrinsic gradient sensor alongside its extra user-specified parameter and use of relatively fine initial meshes with very few resolution levels (Haleem et al., 2015).

More recently, adaptive multiwavelet-based schemes have been devised based on a multiresolution analysis implemented using multiwavelets (MW) within the local basis of DG methods (Hovhannisyan et al., 2014, Gerhard et al., 2015, Gerhard et al., 2015, Gerhard and Müller, 2014). Adaptive MWDG schemes have also been proposed for the solution of the SWE in the works of Gerhard et al. (2015), Kesserwani et al. (2015), Caviedes-Voullième and Kesserwani (2015), who have highlighted the ability of these approaches to:

- Achieve resolution refinement and coarsening driven by a single user-prescribed parameter;
- Rigorously transfer and recover data between disparate resolution levels, thereby allowing arbitrarily large resolution gaps and any degree of mesh coarsening; and,
- Readily preserve accuracy, conservation and robustness properties of the underlying DG scheme.

Starting with a robust DG2 hydrodynamic model, MW can be introduced subject to appropriate scaling of the DG2 local basis functions to form an MWDG2 scheme in which piecewise-linear modelled data can be analysed, scaled and assembled into an adaptive solution. Compared to the HFV1 adaptive solver, which relies on piecewise-constant modelled data, MW allow greater compression rates. However, the strength of this property relating to standard FV1 and DG2 models is not yet identified from consistent MWDG2 and HFV1 schemes that fully exploit local (multi)wavelet compression of data.

This paper studies the behaviour of (multi)wavelets integrated within robust FV1 and DG2 solvers, and identifies the extent of their benefits and limitations for hydraulic modelling. In Section 2, a practical implementation of an MWDG2 solver is presented that fully exploits local MW compression of data, and in which an HFV1 solver is obtained by direct simplification from the MWDG2 formulation (Section 2.4). Section 2 includes also the formulation of a scaled DG2 solver (Section 2.1) with guiding principles on how it readily fits into the multiresolution analysis of MW (Section 2.2) to form the so-called adaptive MWDG2 scheme (Section 2.3). In Section 3, the adaptive HFV1 and MWDG2 solvers are systematically tested and compared in the simulation of well-known hydraulic tests that cover elementary aspects relevant to accurate, efficient and robust hydraulic modelling. The adaptive solvers are run starting from an initial mesh with a single element spanning the entire domain, and the accuracy and efficiency of the adaptive solvers are quantified in relation to standard FV1 and DG2 simulations on the uniform grid involving the finest resolution accessible to the adaptive solvers. In Section 4, key findings and conclusions of this work are summarised. Numerical simulation data (Shaw and Kesserwani, 2018) and a Fortran 2003 implementation of the HFV1/MWDG2 shallow flow models (Shaw et al., 2018) are available to download from Zenodo. Instructions for running the models and interpreting the data are provided in Appendix 1.

2. Adaptive MWDG2 scheme

This section outlines the implementation details of an MWDG2 solver for the conservative form of the standard SWE with source terms over a 1D domain Ω, written as:

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

where ∂_t and ∂_x represent partial derivatives with respect to t and x, $\mathbf{U}(x, t)$ is the vector of the state variables at a location x and time t, $\mathbf{F}(\mathbf{U})$ is the spatial flux vector and $\mathbf{S}(\mathbf{U})$ is a vector including bed and friction slope terms. These vectors are given by:

$$\mathbf{U} = \begin{bmatrix} h \\ q \end{bmatrix},\ \mathbf{F} = \begin{bmatrix} q \\ \frac{q^2}{h} + g\frac{h^2}{2} \end{bmatrix} \text{ and } \mathbf{S} = \begin{bmatrix} 0 \\ S_b + S_f \end{bmatrix} \tag{2}$$

where g (m/s^2) is gravity, h (m) is the water height, $q = hv$ (m^2/s) is the flow discharge per unit width with v (m/s) being the velocity, and $z(x)$ is the topography function in the bed slope source term $S_b = -gh\,\partial_x z$. The term $S_f = -C_f v\,|v|$ represents the energy loss due to friction effects with $C_f = g\,n_M^2/h^{1/3}$ in which n_M is the Manning's bed roughness coefficient.

2.1. Scaled DG2 formulation

The 1D domain Ω is divided into a set of M elements $\{I_i\}_{i=1,\ \ldots,\ M}$ by means of $M+1$ interface points $[x_{i-1/2}]_{i=1,\ \ldots,\ M+1}$ such that $I_i = [x_{i-1/2},\ x_{i+1/2}]$ is a segment with $\Omega = \cup_{i=1}^{M} I_i$ and $I_i \bigcap I_{i+1} = \{x_{i+1/2}\}$. An element I_i has the centre $x_i = \frac{1}{2}(x_{i+1/2} + x_{i-1/2})$ and size $\Delta x = x_{i+1/2} - x_{i-1/2}$. I_i can be mapped into a reference element $[-1,\ 1]$ by the following change of variable $\xi(x) = 2(x - x_i)/\Delta x$; therefore $\xi(x)$, such that $x(\xi) = x_i + \xi\ \Delta x/2$, can be used to position I_i onto $[-1,\ 1]$.

2.1.1. Finite element weak form

By multiplying Eq. (1) by a test function $v(x)$, integrating by parts to remove ∂_x on the flux term, and moving the flux terms to the RHS, the following weak form can be obtained (Cockburn and Shu, 2001):

$$\int_\Omega \partial_t \mathbf{U}(x,t)\ v(x)dx = -\left\{ [\mathbf{F}(\mathbf{U}(x,t))\ v(x)]_{\partial\Omega} - \int_\Omega \mathbf{F}(\mathbf{U}(x,t))\ \partial_x v(x)dx \right. \\ \left. - \int_\Omega \mathbf{S}(\mathbf{U}(x,t)) v(x)dx \right\} \tag{3}$$

It is worth noting that, in Eq. (3), the incorporation of appropriate local bases functions (orthonormal, compactly-supported and discontinuous) as choices for the test function $v(x)$ and for expanding an approximate solution $\mathbf{U}_h = [h_h\ q_h]^{\mathrm{T}}$ to $\mathbf{U}$ are key ingredients to designing an adaptive MWDG scheme (Hovhannisyan et al., 2014, Gerhard et al., 2015). These choices are needed in order to:

(i) Embed local resolution variability into the basis functions shaping the DG spatial operators via a *dual basis*;
(ii) Expand a local DG approximate solution that is compatible with multi-scale decomposition offered by MW via a *primal basis*; and,
(iii) Get the identity matrix as the only multiplier of the time derivative term $\partial_t \mathbf{U}$ in the LHS of Eq. (3) via deploying *bi-orthonormal primal and dual bases*.

The key concepts relevant to these basis functions are introduced next as appropriate.

2.1.2. Choice of bi-orthonormal bases

The starting point is to consider the Legendre basis of polynomials up to first-order within the scope of designing a DG2 scheme (Cockburn and Shu, 2001). This basis is denoted by $\mathbf{P} = [P^0\ P^1]^{\mathrm{T}}$ with $P^0(\xi) = 1$ and $P^1(\xi) = \xi$. As such, it is compactly-supported on $[-1,\ 1]$, inherently discontinuous at $\xi = \pm 1$, and orthogonal for the L^2-norm defined by the following inner product:

$$\langle f, g \rangle = \int_{\Omega} f(\xi) g(\xi) d\xi \tag{4}$$

The basis $\mathbf{P}$ is normalised for the L^2-norm to produce the L^2-orthonormal basis $\hat{\mathbf{P}} = [\hat{P}^0\ \hat{P}^1]^{\mathrm{T}}$, such that $\langle \hat{P}^K, \hat{P}^{K'} \rangle = \delta_{KK'}$ where $\delta_{KK'} = 1$ for $K = K'$ and $\delta_{KK'} = 0$ otherwise. The components of the orthonormal basis $\hat{\mathbf{P}}$ are (Alpert et al., 2002):

$$\hat{P}^K(\xi) = \sqrt{\frac{2K+1}{2}}\, P^K(\xi) \quad (K = 0, 1 \text{ and } \xi \in [-1, 1]) \tag{5}$$

From the orthonormal basis components $\hat{P}^0$ and $\hat{P}^1$, the local primal and dual bases can be defined over I_i, which are denoted as $\boldsymbol{\phi}_i = [\varphi_i^0(x)\ \varphi_i^1(x)]^{\mathrm{T}}$ and $\tilde{\boldsymbol{\phi}}_i = [\tilde{\varphi}_i^0(x)\ \tilde{\varphi}_i^1(x)]^{\mathrm{T}}$ with:

$$\varphi_i^K(x) = \sqrt{2}\ \hat{P}^K(\xi(x)) \quad \left(K = 0, 1 \text{ and } x \in I_i\right) \tag{6}$$

$$\tilde{\varphi}_i^K(x) = \frac{\varphi_i^K(x)}{\Delta x} \quad \left(K = 0, 1 \text{ and } x \in I_i\right) \tag{7}$$

Each of the primal and the dual bases is compactly-supported, orthogonal and discontinuous at the interfaces $x_{i\pm 1/2}$ of the element I_i. These bases are bi-orthonormal since the following relationship holds:

$$\left\langle \varphi_i^K, \tilde{\varphi}_{i'}^{K'} \right\rangle = \delta_{ii'} \delta_{KK'} \tag{8}$$

2.1.3. DG2 operators

By choosing the test function $v(x)$ as the components of the dual basis $\tilde{\varphi}_i^K(x)$ in Eq. (7) and exploiting their orthogonality and compact-support properties, the weak form in Eq. (3) becomes:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \partial_t \mathbf{U}\ \tilde{\varphi}_i^K(x)\, dx = \mathbf{L}_i^K(\mathbf{U}) \quad (K = 0, 1) \tag{9}$$

where $\mathbf{L}_i^K(\mathbf{U})$ are operators involving spatial evaluations of flux and source terms, given by:

$$\begin{aligned}\mathbf{L}_i^K = -\Bigg\{ & \left[\mathbf{F}(\mathbf{U})\ \tilde{\varphi}_i^K(x)\right]_{x_{i-1/2}}^{x_{i+1/2}} - \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{F}(\mathbf{U})\ \partial_x \tilde{\varphi}_i^K(x) dx \\ & - \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{S}(\mathbf{U}) \tilde{\varphi}_i^K(x) dx \Bigg\}\end{aligned} \tag{10}$$

$\mathbf{U}$ is replaced by an approximate solution $\mathbf{U}_h$ expressed in terms of the primal basis as:

$$\mathbf{U}_h(x,t)|_{Ii} = \sum_{K=0}^{1} \mathbf{U}_i^K(t) \varphi_i^K(x) = \mathbf{U}_i^0(t) + \sqrt{3}\ \xi(x)\ \mathbf{U}_i^1(t) \tag{11}$$

in which $\mathbf{U}_i^0(t)$ and $\mathbf{U}_i^1(t)$ are expansion coefficients, or modes, representing an average and a slope characterising the local linear approximation of $\mathbf{U}_h$ over I_i. The initial state of the coefficients at the RHS of Eq. (11), $\mathbf{U}_i^K(0)$, is obtained by projecting a given initial condition $\mathbf{U}_0(x) = \mathbf{U}(x, 0)$ onto the dual basis as follows:

$$\mathbf{U}_i^K(0) = \left\langle \mathbf{U}_0, \tilde{\varphi}_i^K \right\rangle = \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}_0(x)\ \tilde{\varphi}_i^K(x)\ dx \tag{12}$$

which, once mapped into the reference element $[-1,\ 1]$ for applying $(K+1)$ Gauss–Legendre quadrature rules and then manipulated to involve interface evaluations (Kesserwani et al., 2010), yield the following expressions for initialising the initial average and slope coefficients:

$$\mathbf{U}_i^0(0) \approx \frac{1}{2}\left[\mathbf{U}_0\left(x_{i+1/2}\right) + \mathbf{U}_0\left(x_{i-1/2}\right)\right] \tag{13}$$

$$\mathbf{U}_i^1(0) \approx \frac{1}{2\sqrt{3}}\left[\mathbf{U}_0\left(x_{i+1/2}\right) - \mathbf{U}_0\left(x_{i-1/2}\right)\right] \tag{14}$$

Now, considering Eqs. (9)–(10) with $\mathbf{U}_h$ instead of $\mathbf{U}$, and exploiting the bi-orthonormality property, via Eq. (8), the system of PDEs is locally decoupled to solve for two independent ODEs over I_i:

$$\partial_t \mathbf{U}_i^K(t) = \mathbf{L}_i^K\left(\mathbf{U}_h\right) \quad (K = 0, 1) \tag{15}$$

The time derivative in Eq. (15) is solved using an explicit two-stage Runge–Kutta (RK2) time-stepping scheme (e.g. as described in Kesserwani et al., 2010), which requires evaluation of the spatial DG2 operators $\mathbf{L}_i^K(\mathbf{U}_h)$ to evolve $\mathbf{U}_i^K(t)$ over I_i over each RK2 stage. For simplicity, the local DG2 operators $\mathbf{L}_i^K(\mathbf{U}_h)$ is denoted hereafter by $\mathbf{L}_i^K$, which can be expressed as:

$$\begin{aligned}\mathbf{L}_i^K = -\Bigg\{ & \mathbf{F}\left(\mathbf{U}_h\left(x_{i+1/2}, t\right)\right)\ \tilde{\varphi}_i^K\left(x_{i+1/2}\right) \\ & - \mathbf{F}\left(\mathbf{U}_h\left(x_{i-1/2}, t\right)\right)\ \tilde{\varphi}_i^K\left(x_{i-1/2}\right) - \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{F}\left(\mathbf{U}_h(x,t)\right)\ \partial_x \tilde{\varphi}_i^K(x) dx \\ & - \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{S}\left(\mathbf{U}_h(x,t)\right) \tilde{\varphi}_i^K(x) dx \Bigg\}\end{aligned} \tag{16}$$

Adopting discontinuous basis functions allows $\mathbf{U}_h$ to be discontinuous at the element interfaces $x_{i\pm 1/2}$. To incorporate both limits, $\mathbf{U}_h^-(x_{i\pm 1/2}, t)$ and $\mathbf{U}_h^+(x_{i\pm 1/2}, t)$ in the flux evaluation therein, a numerical flux function $\tilde{\mathbf{F}}\ (\cdot,\cdot)$ is introduced as is usually done in Godunov-type finite volume methods (Toro and Garcia-Navarro, 2007, Toro, 2001). By further mapping $\mathbf{L}_i^K$ onto the reference element where $(K + 1)$ Gauss–Legendre quadrature rules can be applied to approximate volume integral terms of the flux and source terms, and by considering only the bed slope source term $\mathbf{S}_b = [0\ S_b]^{\mathrm{T}}$, Eq. (16) becomes:

$$\mathbf{L}_i^0 = -\frac{1}{\Delta x}\left\{\tilde{\mathbf{F}}_{i+1/2}\ -\ \tilde{\mathbf{F}}_{i-1/2} - \Delta x\ \mathbf{S}_b\left(\mathbf{U}_i^0, \partial_x z_h\right)\right\} \tag{17}$$

$$\mathbf{L}_i^1 = -\frac{\sqrt{3}}{\Delta x}\left\{\begin{array}{l}\tilde{\mathbf{F}}_{i+1/2} +\ \tilde{\mathbf{F}}_{i-1/2} - \left[\mathbf{F}\left(\mathbf{U}_i^0 + \mathbf{U}_i^1\right) + \mathbf{F}\left(\mathbf{U}_i^0 - \mathbf{U}_i^1\right)\right] \\ -\frac{\Delta x}{2\sqrt{3}}\left[\mathbf{S}_b\left(\mathbf{U}_i^0 + \mathbf{U}_i^1, \partial_x z_h\right) - \mathbf{S}_b\left(\mathbf{U}_i^0 - \mathbf{U}_i^1, \partial_x z_h\right)\right]\end{array}\right\} \tag{18}$$

In Eq. (18), $\tilde{\mathbf{F}}_{i+1/2} = \tilde{\mathbf{F}}\ (\mathbf{U}_{i+1/2}^-, \mathbf{U}_{i+1/2}^+)$ represents a flux evaluation at $x_{i+1/2}$ via a two-argument numerical flux function $\tilde{\mathbf{F}}$ based on the Harten, Lax and van Leer approximate Riemann solver (Toro, 2001). $\mathbf{U}_{i+1/2}^- = \mathbf{U}_h(x_{i+1/2}, t)|_{I_i}$ and $\mathbf{U}_{i+1/2}^+ = \mathbf{U}_h(x_{i+1/2}, t)|_{I_{i+1}}$ denote the limits of $\mathbf{U}_h$ at both sides from $x_{i+1/2}$, which are known as *Riemann states,* at which wetting and drying considerations occur (as outlined later in Section 2.3.3). These limits can obtained from Eq. (11) as follows:

$$\mathbf{U}_{i+1/2}^- = \mathbf{U}_i^0(t) + \sqrt{3}\ \mathbf{U}_i^1(t) \text{ and } \mathbf{U}_{i+1/2}^+ = \mathbf{U}_{i+1}^0(t) - \sqrt{3}\ \mathbf{U}_{i+1}^1(t) \tag{19}$$

The bed slope discretisation in $\mathbf{S}_b$ is performed by expanding z_h locally over I_i onto the primal basis, consistently with the shaping of the local approximate solution (Eqs. (11)–(14)):

$$z_h(x)|_{I_i} = z_i^0 + \sqrt{3}\,\xi(x)\,z_i^1 \tag{20}$$

with z_i^0 and z_i^1 being time-independent modes for the topography term approximation, which can be initialised as in Eqs. (13)–(14), by:

$$z_i^0 \approx \frac{1}{2}\left[z\left(x_{i+1/2}\right) + z\left(x_{i-1/2}\right)\right] \tag{21}$$

$$z_i^1 \approx \frac{1}{2\sqrt{3}}\left[z\left(x_{i+1/2}\right) - z\left(x_{i-1/2}\right)\right] \tag{22}$$

The discretisation is then completed by extracting an approximate partial derivative while mapping from the reference element:

$$\partial_x z_h|_{I_i} = \frac{2\sqrt{3}}{\Delta x}\,z_i^1 \tag{23}$$

Therefore, the expressions of the bed slope source terms involved in Eqs. (17) and (18) become:

$$\mathbf{S}_b\left(\mathbf{U}_h(x,t), \partial_x z_h\right) = \frac{2\sqrt{3}}{\Delta x_i}\begin{bmatrix} 0 \\ -g\ h_h(x,t)\ z_i^1 \end{bmatrix} \tag{24}$$

Substituting Eq. (24) into Eqs. (17) and (18), the DG2 operators can be further simplified to:

$$\mathbf{L}_i^0 = -\frac{1}{\Delta x}\left\{\tilde{\mathbf{F}}_{i+1/2} - \tilde{\mathbf{F}}_{i-1/2} + \begin{bmatrix} 0 \\ 2g\sqrt{3}h_i^0\ z_i^1 \end{bmatrix}\right\} \tag{25}$$

$$\mathbf{L}_i^1 = -\frac{\sqrt{3}}{\Delta x}\left\{\tilde{\mathbf{F}}_{i+1/2} + \tilde{\mathbf{F}}_{i-1/2} - \mathbf{F}\left(\mathbf{U}_i^0 + \mathbf{U}_i^1\right) - \mathbf{F}\left(\mathbf{U}_i^0 - \mathbf{U}_i^1\right) + \begin{bmatrix} 0 \\ 2gh_i^1\,z_i^1 \end{bmatrix}\right\} \tag{26}$$

2.1.4. Extension to multiresolution bases

From the *same* L^2-orthonormal basis $\hat{\mathbf{P}}$, a series of child bases $\{\hat{\mathbf{P}}^{(n)}\}_n$ can be defined given its property of being a *refinable function* (Alpert et al., 2002, Keinert, 2004, Alpert, 1993) – where n is a positive integer indicating the refinement level, which will hereafter be used as a bracketed superscript 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 refinement level (n) can be associated with a grid $g^{(n)}$ based on n dyadic subdivisions of the support $[-1, 1]$ of $\hat{\mathbf{P}}$. Hence, $g^{(n)}$ spans $[-1, 1]$ such that $g^{(n)} = \cup_{j=0}^{2^n-1} I_j^{(n)}$, where $\{I_j^{(n)}\}_{j=0,1,\ldots,2^n-1}$ is a set of non-overlapping sub-divisions of $[-1, 1]$. Moreover, a sub-division $I_j^{(n)}$ can be regarded as a sub-element of $[-1, 1]$, taking the following form:

$$I_j^{(n)} = \left[\chi_{j-1/2}, \chi_{j+1/2}\right] \tag{27}$$

with $\chi_{j-1/2} = -1 + \frac{2}{2^n}\,j$ are interface points forming sub-elements $\{I_j^{(n)}\}_{j=0,1,\ldots,2^n-1}$, and the index $j = 0, 1, \ldots, 2^n - 1$ representing the position of $I_j^{(n)}$ in $g^{(n)}$, on which the components $\hat{\mathbf{P}}_j^{(n)}$ of the basis $\hat{\mathbf{P}}^{(n)} = [\hat{\mathbf{P}}_j^{(n)}]_j$ can be obtained by translation and dilatation of $\hat{\mathbf{P}}$, as follows:

$$\hat{\mathbf{P}}_j^{(n)}(\chi) = \left(\sqrt{2}\right)^n\,\hat{\mathbf{P}}(2^n(\chi+1) - 2j - 1)\ \left(\chi \in I_j^{(n)}\right) \tag{28}$$

From the compact-support and L^2-orthonormality properties of $\{\hat{\mathbf{P}}^{(n)}\}_n$, the grids $\{g^{(n)}\}_n$ form a hierarchy spanning $[-1, 1]$, i.e. $\cup_n g^{(n)} = [-1, 1]$, and are globally nested across all refinement levels while having local and non-overlapping support at each level (n).

Similarly, on a mesh element $I_i = [x_{i-1/2}, x_{i+1/2}]$ a hierarchy of nested grids $\{g_i^{(n)}\}_n$ can be defined such that $g_i^{(n)} = \cup_{j=0}^{2^n-1} I_{j,i}^{(n)}$ with $\{I_{j,i}^{(n)}\}_{j=0,1,\ldots,2^n-1}$ now denoting sub-divisions of I_i, with $I_{j,i}^{(n)}$ representing a sub-element of I_i at a position j relative to refinement level (n), namely:

$$I_{j,i}^{(n)} = \left[x_{j-1/2,i}^{(n)}, x_{j+1/2,i}^{(n)}\right] \tag{29}$$

In Eq. (29), $x_{j-1/2,i}^{(n)} = x_{i-1/2} + \Delta x^{(n)} j$ are interface points forming sub-elements $\{I_{j,i}^{(n)}\}_{j=0,1,\ldots,2^n-1}$ and $\Delta x^{(n)} = \Delta x/2^n$ is the grid spacing relative to grid $g_i^{(n)}$ with positions j such that $j = 0, 1, \ldots, 2^n - 1$. For convenience of presentation, sub-elements $I_{j,i}^{(n)}$ will hereafter be denoted by $I_e^{(n)}$ where index "*e*" is shorthand for "*j, i*" to position sub-elements in I_i. Thereby, sub-elements $I_e^{(n)}$ can be linked to $I_j^{(n)}$ by translation into $[-1, 1]$. This also makes it easy to keep consistent with the notation associated with the DG2 method presented previously (Sections 2.1.1–2.1.3) for application at sub-elements $I_e^{(n)}$, which take the following form:

$$I_e^{(n)} = \left[x_{e-1/2}^{(n)}, x_{e+1/2}^{(n)}\right] \tag{30}$$

with $x_e^{(n)}$ and $\Delta x_e^{(n)}$ being the centre position and the size of a sub-element $I_e^{(n)}$, respectively. On $I_e^{(n)} \in g_i^{(n)}$ bi-orthonormal dual and primal bases, denoted by $\boldsymbol{\phi}_e$ and $\tilde{\boldsymbol{\phi}}_e$, can be defined via the refined bases $[\hat{\mathbf{P}}_j^{(n)}]_j$ by analogy (recall Eqs. (6)–(7), and take the form:

$$\boldsymbol{\phi}_e^{(n)}(x) = \sqrt{2}\,\hat{\mathbf{P}}_j^{(n)}(\chi) \quad \left(x \in I_e^{(n)} \subset I_i\right) \tag{31}$$

$$\tilde{\boldsymbol{\phi}}_e^{(n)}(x) = \frac{\boldsymbol{\phi}_e^{(n)}(x)}{\Delta x^{(n)}} \quad \left(x \in I_e^{(n)} \subset I_i\right) \tag{32}$$

where $\chi(x) = 2(x - x_e^{(n)})/\Delta x_e^{(n)}$ is a change of variable used to map the position $x \in I_e^{(n)}$ into $I_j^{(n)}$. Adopting the local basis functions in Eqs. (31)–(32), and reworking the steps in Section 2.1.3, yield similar DG2 operators for any sub-element $I_e^{(n)} \in g_i^{(n)}$, which are similar to Eqs. (25)–(26) but with index e instead of i and the grid spacing $\Delta x^{(n)}$ of $g_i^{(n)}$ instead of Δx. Such DG2 operators can be applied to evolve DG2 modes $\mathbf{U}_e^0(t)$ and $\mathbf{U}_e^1(t)$, spanning local *flow* solutions $\mathbf{U}_h(x,t)|_{I_e^{(n)}}$ over any sub-element $I_e^{(n)} \in \{g_i^{(n)}\}_n$, starting from initial *flow* modes as described in Eqs. (13) and (14) with index e instead of i. Similarly, topography modes, z_e^0 and z_e^1 on $I_e^{(n)}$, can be initialised as in Eqs. (21) and (22) for use in the DG2 operators on $I_e^{(n)}$.

To ease the presentation in the following sections, DG2 flow and topography modes ($\mathbf{U}_e^0(t)$, $\mathbf{U}_e^1(t)$, z_e^0 and z_e^1) will be considered component-wise, and the scalar variable $u \in \{h, q, z\}$ will be used to represent any physical quantities in $\mathbf{U} = [h\ q]^{\mathrm{T}}$ and z. Since each u has DG2 modes, which are actually its spectral components in terms of average and slope coefficients, DG2 modes of any physical quantity u on sub-elements $I_e^{(n)} \in \{g_i^{(n)}\}_n$ will be denoted as $\boldsymbol{u}_e^{(n)} = [u_e^{0,(n)}\ u_e^{1,(n)}]$.

2.2. Multiresolution analysis

From the *same* L^2-orthonormal basis $\hat{\mathbf{P}}$, child bases $\{\hat{\mathbf{P}}^{(n)}\}_n$ and multiwavelet bases $\{\boldsymbol{\Psi}^{(n)}\}_n$ can be defined. This allows multiresolution analysis to be performed, which is summarised in this section with a view to presenting how it is directly applicable to analysing the behaviour of the DG2 modes on multiresolution bases.

2.2.1. Relationship between the scaling bases $\{\hat{\mathbf{P}}^{(n)}\}_n$

From the properties of the scaling bases $\{\hat{\mathbf{P}}^{(n)}\}_n$ defined on the hierarchy of grids $\{g^{(n)}\}_n$, it is possible to produce a recurrence relationship for binary merging of two adjacent components of the bases belonging to $g^{(n+1)}$ to form the components of the bases in $g^{(n)}$. Without loss of generality, it suffices to outline the relationship linking an elementary father basis $[\hat{\mathbf{P}}_j^{(n)}]$ and its child bases $[\hat{\mathbf{P}}_{2j}^{(n+1)}\ \hat{\mathbf{P}}_{2j+1}^{(n+1)}]$, in particular for the case between $g^{(0)}$ and $g^{(1)}$ where $n = j = 0$. This relationship between the scaling bases can be achieved by involving the so-called *low-pass filter matrices* $\mathbf{H}^0$ and $\mathbf{H}^1$ (Alpert et al., 2002, Alpert, 1993), which allow $\hat{\mathbf{P}}_j^{(n)}$ to be expressed as linear combination of $\hat{\mathbf{P}}_{2j}^{(n+1)}$ and $\hat{\mathbf{P}}_{2j+1}^{(n+1)}$:

$$\hat{\mathbf{P}}_j^{(n)} = \mathbf{H}^0\,\hat{\mathbf{P}}_{2j}^{(n+1)} + \mathbf{H}^1\,\hat{\mathbf{P}}_{2j+1}^{(n+1)} \tag{33}$$

$$\mathbf{H}^0 = \left[\left\langle \hat{\mathbf{P}}_j^{(n)}, \hat{\mathbf{P}}_{2j}^{(n+1)} \right\rangle\right] = \begin{bmatrix} 1/\sqrt{2} & 0 \\ -\sqrt{6}/4 & \sqrt{2}/4 \end{bmatrix} \tag{34}$$

$$\mathbf{H}^1 = \left[\left\langle \hat{\mathbf{P}}_j^{(n)}, \hat{\mathbf{P}}_{2j+1}^{(n+1)} \right\rangle\right] = \begin{bmatrix} 1/\sqrt{2} & 0 \\ \sqrt{6}/4 & \sqrt{2}/4 \end{bmatrix} \tag{35}$$

2.2.2. Multiwavelet bases and their relationship to the scaling bases

Now reconsidering the father basis $\hat{\mathbf{P}}$, a *mother basis* of wavelets $\mathbf{\Psi}$, or *multiwavelets* (Alpert et al., 2002), can be defined on $g^{(0)} = [-1, 1]$, which represents the encoded (L^2-orthonormal) difference between $\hat{\mathbf{P}} = \hat{\mathbf{P}}_0^{(0)}$ and the components of its two child bases $[\hat{\mathbf{P}}_0^{(1)}\ \hat{\mathbf{P}}_1^{(1)}]$ supported on $g^{(1)} = [-1, 0] \cup [0, 1]$. In essence, $\mathbf{\Psi}$ represents the (L^2-orthonormal) complement of $\hat{\mathbf{P}}_0^{(0)} = \hat{\mathbf{P}}$ in $g^{(1)}$. Therefore, $\mathbf{\Psi}$ is one refinement level higher than $\hat{\mathbf{P}}_0^{(0)}$ and spans $g^{(0)} \cap g^{(1)}$, taking the form (Alpert et al., 2002):

$$\mathbf{\Psi}(\chi) = \left[\psi_0^{(0)}(\chi)\ \psi_1^{(0)}(\chi)\right] \tag{36}$$

$$\psi_0^{(0)}(\chi) = \begin{cases} -\sqrt{\frac{3}{2}}(2\chi+1)\ \chi \in I_0^{(1)} \\ +\sqrt{\frac{3}{2}}(2\chi-1)\ \chi \in I_1^{(1)} \end{cases} \text{and}\, \psi_1^{(0)}(\chi) = \begin{cases} \sqrt{\frac{1}{2}}(3\chi+2)\ \chi \in I_0^{(1)} \\ \sqrt{\frac{1}{2}}(3\chi-2)\ \chi \in I_1^{(1)} \end{cases} \tag{37}$$

with $I_0^{(1)} = [-1, 0]$ and $I_1^{(1)} = [0, 1]$ denoting the two shifts forming $g^{(1)}$, for generality relating to Eq. (27). Note that $\mathbf{\Psi}$ admits a discontinuity at $\chi = 0$, which offers an advantage for the analysis of signals with discontinuities. Moreover, $\mathbf{\Psi}$ and $\hat{\mathbf{P}}$ are bi-orthonormal with the former inheriting the properties of the latter. Hence, a series of child multiwavelets $\{\mathbf{\Psi}^{(n)}\}_n$ can be defined on the hierarchy of grids $\{g^{(n)}\}_n$ by translation and dilatation of $\mathbf{\Psi}$, such that on a grid $g^{(n)} = \cup_{j=0}^{2^n-1} I_j^{(n)}$, $\mathbf{\Psi}^{(n)} = [\mathbf{\Psi}_j^{(n)}]_{j=0,1,\ldots,2^n-1}$ where each $\mathbf{\Psi}_j^{(n)}$ takes the following form:

$$\mathbf{\Psi}_j^{(n)}(\chi) = \left(\sqrt{2}\right)^n \mathbf{\Psi}(2^n(\chi+1) - 2j - 1)\ \left(\chi \in I_j^{(n)}\right) \tag{38}$$

From the scaling bases, binary merging of two adjacent components belonging to $g^{(n+1)}$ can be achieved to produce the components of the multiwavelet bases in $g^{(n)}$. Again, it suffices to outline the relationship linking an elementary multiwavelet basis $[\mathbf{\Psi}_j^{(n)}]$ in $g^{(n)}$ to the scaling bases $[\hat{\mathbf{P}}_{2j}^{(n+1)}\ \hat{\mathbf{P}}_{2j+1}^{(n+1)}]$ in $g^{(n+1)}$ for $n = j = 0$. This relationship can be expressed by using the so-called *high-pass filter* matrices $\mathbf{G}^0$ and $\mathbf{G}^1$, which allow $\mathbf{\Psi}_j^{(n)}$ to be derived as linear combination of $\hat{\mathbf{P}}_{2j}^{(n+1)}$ and $\hat{\mathbf{P}}_{2j+1}^{(n+1)}$:

$$\mathbf{\Psi}_j^{(n)} = \mathbf{G}^0\ \hat{\mathbf{P}}_{2j}^{(n+1)} + \mathbf{G}^1\ \hat{\mathbf{P}}_{2j+1}^{(n+1)} \tag{39}$$

$$\mathbf{G}^0 = \left[\left\langle \mathbf{\Psi}_j^{(n)}, \hat{\mathbf{P}}_{2j}^{(n+1)} \right\rangle\right] = \begin{bmatrix} 0 & -1/\sqrt{2} \\ \sqrt{2}/4 & \sqrt{6}/4 \end{bmatrix} \tag{40}$$

$$\mathbf{G}^1 = \left[\left\langle \mathbf{\Psi}_j^{(n)}, \hat{\mathbf{P}}_{2j+1}^{(n+1)} \right\rangle\right] = \begin{bmatrix} 0 & 1/\sqrt{2} \\ -\sqrt{2}/4 & \sqrt{6}/4 \end{bmatrix} \tag{41}$$

2.2.3. Single-scale vs. multi-scale expansions

The definition of scaling and multiwavelet bases on the hierarchy of grids $\{g^{(n)}\}_n$ allows for two interchangeable ways to approximate a given scalar signal $s(\xi)$ defined on $[-1, 1]$. Given a grid $g^{(n)} = \cup_{j=0}^{2^n-1} I_j^{(n)}$ associated with the scaling bases $\hat{\mathbf{P}}^{(n)} = [\hat{\mathbf{P}}_j^{(n)}]_{j=0,1,\ldots,2^n-1}$, an approximation $s_h(\xi)$ of the signal $s(\xi)$ can be obtained by expanding it onto the bases $\hat{\mathbf{P}}^{(n)}$ as follows (Alpert et al., 2002):

$$s_h(\xi) = \sum_{j=0}^{2^n-1} s_h(\chi)|_{I_j^{(n)}} \tag{42}$$

in which $s_h(\chi)|_{I_j^{(n)}}$ is a piecewise-linear expansions onto each basis $\hat{\mathbf{P}}_j^{(n)}$ that is compactly-supported on the sub-element $I_j^{(n)}$. The signal approximation can therefore be expressed as:

$$s_h|_{I_j^{(n)}} = \left\langle \mathbf{s}_j^{(n)},\ \hat{\mathbf{P}}_j^{(n)} \right\rangle = s_j^{0,(n)}\ \hat{P}_j^{0,(n)} + s_j^{1,(n)}\ \hat{P}_j^{1,(n)} \tag{43}$$

where $\mathbf{s}_j^{(n)} = [s_j^{0,(n)}\ s_j^{1,(n)}]$ denotes local scale coefficients expanding $s_h(\chi)|_{I_j^{(n)}}$ onto the basis $\hat{\mathbf{P}}_j^{(n)}$, which can be initialised as $s_j^{K,(n)} = \langle s, \hat{P}_j^{K,(n)} \rangle$ with $K = 0, 1$. This type of description, i.e. in Eqs. (42) and (43), is called *single-scale expansion* as it only involves scale coefficicents from the grid $g^{(n)}$, at a single-scale refinement level (n).

Another way to expand $s_h(\xi)$ is to involve the multiwavelet bases. By doing so, the single-scale description of in Eqs. (42)–(43) can be recursively decomposed to produce a so-called *multi-scale expansion*. This form of description sums up the features of $s_h(\xi)$, via wavelet coefficients, throughout grids $g^{(0)}, \ldots, g^{(n-1)}$ to its background information at its coarsest level (i.e. the scale coefficients on $g^{(0)}$). Hence, the multi-scale expansion takes the form (Alpert et al., 2002):

$$s_h(\xi) = s_h(\xi)|_{I_0^{(0)}} + \sum_{l=0}^{n-1}\left(\sum_{j=0}^{2^l-1} \left\langle \boldsymbol{d}_j^{(l)}(\chi),\ \boldsymbol{\psi}_j^{(l)}(\chi) \right\rangle\right) \tag{44}$$

$$\left\langle \boldsymbol{d}_j^{(l)},\ \boldsymbol{\psi}_j^{(l)} \right\rangle = d_j^{0,(n)}\ \psi_j^{0,(l)} + d_j^{1,(n)}\ \psi_j^{1,(l)} \tag{45}$$

with $\boldsymbol{d}_j^{(l)} = [d_j^{0,(l)}\ d_j^{1,(l)}]$ denoting the local *details* also known as *detail coefficients* or *wavelet coefficients*. They can be initialised as $d_j^{K,(l)} = \langle s, \psi_j^{K,(l)} \rangle$ with $K = 0, 1$. The multi-scale expansion in Eqs. (44) and (45) clearly distinguishes the details of $s_h(\xi)$ between successively higher resolution, which become increasingly significant with increasing levels of non-smoothness in $s_h(\xi)$ while remaining negligible where $s_h(\xi)$ is smooth. Therefore, it provides a mechanism to analyse, decompose and reconstruct the approximate signal $s_h(\xi)$ across the grids in the hierarchy $\{g^{(n)}\}_n$.

2.2.4. Two-scale transformations between coefficients

From the link between the high- and low-pass filter matrices Alpert et al., 2002) outlined previously in Eqs. (34)–(35) and (40)–(41), relationships for scaling up or down (recurrently) relevant coefficients between subsequent resolution levels (n) and $(n+1)$ can be produced, namely:

$$\begin{cases} \boldsymbol{s}_j^{(n)} = \mathbf{H}^0\ \boldsymbol{s}_{2j}^{(n+1)} + \mathbf{H}^1\ \boldsymbol{s}_{2j+1}^{(n+1)} \\ \boldsymbol{d}_j^{(n)} = \mathbf{G}^0\ \boldsymbol{s}_{2j}^{(n+1)} + \mathbf{G}^1\ \mathbf{s}_{2j+1}^{(n+1)} \end{cases} \tag{46}$$

$$\begin{cases} \boldsymbol{s}_{2j}^{(n+1)} = \left[\mathbf{H}^0\right]^{\mathrm{T}}\ \boldsymbol{s}_j^{(n)} + \left[\mathbf{G}^0\right]^{\mathrm{T}}\ \boldsymbol{d}_j^{(n)} \\ \boldsymbol{s}_{2j+1}^{(n+1)} = \left[\mathbf{H}^1\right]^{\mathrm{T}}\ \boldsymbol{s}_j^{(n)} + \left[\mathbf{G}^1\right]^{\mathrm{T}}\ \boldsymbol{d}_j^{(n)} \end{cases} \tag{47}$$

Eq. (46) is useful to *encode* (or extract) the scale and detail coefficients $s_j^{(n)}$ and $d_j^{(n)}$ at a sub-element $I_j^{(n)} \in g^{(n)}$ from the scale coefficients $s_{2j}^{(n+1)}$ and $s_{2j+1}^{(n+1)}$ of its two child sub-elements $\{I_{2j}^{(n+1)}, I_{2j+1}^{(n+1)}\} \in g^{(n+1)}$. It applies in a descending order across refinement levels starting from sub-elements on the finest grid $g^{(L)}$ with (L) being a maximum refinement level prescribed by a user. This results in a multi-scale expansion, as in Eq. (44), compressing the details across the *whole* hierarchy $\{g^{(n)}\}_{n=0,\,1,\,\ldots,\,L}$. Eq. (47) is used in the opposite sense to *decode* (or combine) scale and wavelet coefficients at any $I_j^{(n)}$ $(n = L-1, \ldots, 1, 0)$ to generate their scale coefficients located one resolution higher, i.e. the scale coefficients on the two sub-elements $I_{2j}^{(n+1)}$ and $I_{2j+1}^{(n+1)}$. Given a multi-scale expansion, Eq. (47) can successively be applied in an ascending order, starting from the information available at the coarsest grid $g^{(0)}$, to retrieve a single-scale expansion, as in Eq. (42), up to any refinement level (n), $0 \le n \le L$.

2.2.5. Extension of the analysis for the DG2 modes on multiresolution bases

To extend the validlity of the analysis in Sections 2.2.1–2.2.4 from bases $\{\hat{\mathbf{P}}^{(n)}\}_n$, spanning $[-1, 1]$, to the multiresolution bases $\{\boldsymbol{\phi}_e^{(n)}\}_n$, spanning I_i, it suffices to consider Eq. (31) and the notation adopted in Section 2.1.4. Now, Eqs. (42)–(45) can be reused for any physical component u, with $u_h|_{I_i}$ being its expansion on I_i by coefficients $[\boldsymbol{u}_e^{(n)}]_{n,e}$, as in Eq. (42). Each $\boldsymbol{u}_e^{(n)}$ contains the expansion coefficients of a local linear DG2 solution on sub-elements $I_e^{(n)} \subset I_i$, as in Eq. (43), or DG2 modes as $\boldsymbol{u}_e^{(n)} = [u_e^{0,(n)}\ u_e^{1,(n)}]$.

Over a *selected* grid $g_i^{(n)} = \cup_{e=0}^{2^n-1} I_e^{(n)}$ of the hierarchy of grids $\{g_i^{(n)}\}_n$, DG2 modes $[\boldsymbol{u}_e^{(n)}]_{e=0,1,\ldots,2^n-1}$ can be initialised for the single-scale expansion $u_h|_{I_i}$, which actually represents an *assembled DG2 solution* on grid $g_i^{(n)}$. Alternatively, a multi-scale expansion is also possible as in Eqs. (44) and (45), which is actually a *compressed MWDG2 solution* allowing to access the details $[\boldsymbol{d}_e^{(l)}]_{l,e}$, with $\boldsymbol{d}_e^{(l)} = [d_e^{0,(l)}\ d_e^{1,(l)}]$, living on lower resolution grids $\{g_i^{(l)}\}_{l=n-1,\ldots,1,0}$. These details can be initialised from the DG2 modes on $g_i^{(n)}$ for the physical components $u \in \{h+z,\ q,\ z\}$ as explained later in Section 2.3.1. With this change of bases and variable, the two-scale transformation formulae in Eqs. (46)–(47) should be re-scaled by $\sqrt{2}$ to make them relevant to the DG2 modes and their associated details, leading to modified formulae:

$$\begin{cases} \boldsymbol{u}_e^{(n)} = \frac{1}{\sqrt{2}}\left(\mathbf{H}^0\ \boldsymbol{u}_{2e}^{(n+1)} + \mathbf{H}^1\ \boldsymbol{u}_{2e+1}^{(n+1)}\right) \\ \boldsymbol{d}_e^{(n)} = \frac{1}{\sqrt{2}}\left(\mathbf{G}^0\ \boldsymbol{u}_{2e}^{(n+1)} + \mathbf{G}^1\ u_{2e+1}^{(n+1)}\right) \end{cases} \tag{48}$$

$$\begin{cases} \boldsymbol{u}_{2e}^{(n+1)} = \sqrt{2}\left(\left[\mathbf{H}^0\right]^{\mathrm{T}} \boldsymbol{u}_e^{(n)} + \left[\mathbf{G}^0\right]^{\mathrm{T}} \boldsymbol{d}_e^{(n)}\right) \\ \boldsymbol{u}_{2e+1}^{(n+1)} = \sqrt{2}\left(\left[\mathbf{H}^1\right]^{\mathrm{T}} \boldsymbol{u}_e^{(n)} + \left[\mathbf{G}^1\right]^{\mathrm{T}} \boldsymbol{d}_e^{(n)}\right) \end{cases} \tag{49}$$

As detailed later in Section 2.3, Eqs. (48)–(49) can be *directly* deployed within the scaled DG2 method, as needed, to *encode* information via Eq. (48), i.e. binary merging of DG2 modes on $g_i^{(n+1)}$ to generate coarser modes and/or their details on $g_i^{(n)}$, or decode information via Eq. (49), i.e. adding up the details and modes on $g_i^{(n)}$ to generate the DG2 modes on $g_i^{(n+1)}$. *Encoding* is key to produce, scan and distinguish the details across successive refinement levels from within the *compressed MWDG2 solution*, whereas *decoding* is key to generate an *assembled DG2 solution* from a set of carefully-selected DG2 modes relative to sub-elements with non-uniform size $\Delta x^{(n)}$.

2.3. Multiresolution scaled DG2 adaptive solution

This section describes how multiresolution analysis (Section 2.2) can be used directly within the scaled DG2 formulation (Section 2.1) to produce the so-called adaptive MWDG2 numerical solution. The starting point is to set a desired maximum refinement level (L) and thereby refine the coarsest discretisation of the domain $\Omega = \cup_{i=1}^{M} I_i$to be at the finest uniform resolution allowable (a uniform mesh with $2^L M$ sub-elements). Now, each element I_i has 2^L sub-elements $\{I_e^{(L)}\}_{e=0,1,\ldots,2^L-1}$ such that $I_i = g_i^{(L)} = \cup_{e=1}^{2^L-1} I_e^{(L)}$. Given that the combined MWDG2 functioning can be applied element-wise, we hereafter assume that the coarsest grid spanning Ω is made by a single element, hence we take $M = 1$ without loss of generality. Now $g_i^{(L)}$ represents the finest uniform discretisation for Ω, which is made of sub-elements $\{I_e^{(L)}\}_{e=0,1,\ldots,2^L-1}$. On each sub-element $I_e^{(L)}$, DG2 modes, $\boldsymbol{u}_e^{(L)} = [u_e^{0,(L)}\ u_e^{1,(L)}]$ with $u \in \{h,\ q,\ z\}$ can be initialised in terms of flow and topography data (Section 2.1.4), forming an assembled DG2 solution on the finest grid $g_i^{(L)}$ for initial pre-processing (Section 2.3.1).

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

Initially, DG2 modes $[\boldsymbol{u}_e^{(L)}]_{e=0,1,\ldots,2^L-1}$ of the flow and topography are only available on $g_i^{(L)}$. From these modes, details $[\boldsymbol{d}_e^{(n)}]_{n,e}$ living on the lower resolution grids $\{g_i^{(n)}\}_{n=L-1,\ldots,1,0}$ can be encoded. This is achieved by successive application of Eq. (48) in a descending order, starting from refinement level $(L-1)$ until reaching the coarsest level (0) where both the coarsest modes $\boldsymbol{u}_0^{(0)}$ and details $\boldsymbol{d}_0^{(0)}$ become available. Moreover, details representing the water height h were encoded based on the DG2 modes representing the free-surface elevation $h+z$, which was found necessary to avoid producing misinformative details for h when the topography is very steep. In what follows, the details $[\boldsymbol{d}_e^{(n)}]_{n,e}$ will be actually associated with components $u \in \{h+z,\ q,\ z\}$.

From the details $[\boldsymbol{d}_e^{(n)}]_{n,e}$, an alternative set of *normalised detail magnitudes*, denoted by $[\check{d}_e^{(n)}]_{n,e}$, can be generated. This set is needed to enable measuring the *significance* of *all detail coefficents combined*, regardless of which physical quantity u they represent. Namely, a normalised detail magnitude $\check{d}_e^{(n)}$ is a scalar evaluated from from its detail $\boldsymbol{d}_e^{(n)}$ as (Gerhard et al., 2015):

$$\check{d}_e^{(n)} = \frac{max\left(\left|d_e^{0,(n)}\right|, \left|d_e^{1,(n)}\right|\right)}{max\left(1, \left|max\left(\left[u_e^{0,(L)}\right]_e\right)\right|\right)} \tag{50}$$

where $max([u_e^{0,(L)}]_e)$ is the maximum of the average coefficients of the DG2 modes on $g_i^{(L)}$ – also across the hierarchy $\{g_i^{(n)}\}_{n=L,\ldots,1,0}$ due to variational boundness across refinement levels.

Note that, at the starting time, all details $[\boldsymbol{d}_e^{(n)}]_{n,e}$ for all variables $u \in \{h+z,\ q,\ z\}$ are fully accessible on $\{g_i^{(n)}\}_{n=L-1,\ldots,1,0}$. They can be ascendingly summed upon the coarsest DG2 modes, $\boldsymbol{u}_0^{(0)}$, on I_i to form a *compressed MWDG2 solution on* $\{g_i^{(n)}\}_{n=L,\ldots,1,0}$, which is as accurate as the *assembled DG2 solution* on $g_i^{(L)}$. Later, when $t > 0$, details $[\boldsymbol{d}_e^{(n)}]_{n,e}$ of the flow variables $u \in \{h+z,\ q\}$ are subjected to constant change given the time-dependent nature of $h+z$ and q (Section 2.3.4), while the details of 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 in the hierarchy $\{g_i^{(n)}\}_{n=L,\ldots,1,0}$, an adaptive grid at a present time t, denoted by $g_i^A(t)$, can be formed by selecting *certain* sub-elements:

$$g_i^A(t) \subset \left\{ I_e^{(n)} \in \left\{ g_i^{(n)} \right\}_n,\ 0 \leq n \leq L,\ 0 \leq e \leq 2^L - 1 \text{ and } \Omega = \bigcup_{n,e} I_e^{(n)} \right\} \tag{51}$$

The act of measuring normalised detail magnitudes is here refered to as *prediction* and involves four subsequent steps for *deciding* the sub-elements forming $g_i^A(t)$.

Firstly, an *error threshold* ε needs to be prescribed such that $0 < \varepsilon < 1$, which is a parameter chosen by the user to decide which details can be ignored. While there is no unique choice for ε, an optimal range of choices exists to keep the accuracy of assembled DG2 solution on $g_i^A(t)$ at the same level as the finest resolution accessible on $g_i^{(L)}$ at time t – via the compressed MWDG2 solution (Hovhannisyan et al., 2014). An optimal choice for ε is expected to be somewhere between 10^{-4} and 10^{-2}. Arguably, the choice of ε is rather heuristic, context-specific and seemigly dependent on the order-of-accuracy of the DG scheme (Harten, 1995, Cohen et al., 2003, Gerhard et al., 2015). An analysis on the choice of ε for the adaptive HFV1 and MWDG2 solvers used in the present work is carried out later in Section 3.1.1.

Secondly, normalised details $[\check{d}_e^{(n)}]_{n,e}$ living on $\{g_i^{(n)}\}_{n=L-1,\ldots,1,0}$ are compared to ε for indentifying the *significant details*. In doing so, their magnitudes are scanned, level-wise (in an ascending order $n = 0$, 1, …, $L-1$), and compared to *level-depedent error thresholds* $\varepsilon^{(n)}$ such that $\varepsilon^{(n)} = 2^{n-L}\varepsilon$. Within this process, a detail $\boldsymbol{d}_e^{(n)}$ is classified as *significant* if:

$$\check{d}_e^{(n)} > \varepsilon^{(n)} \tag{52}$$

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

Thirdly, re-flagging of active sub-elements $I_e^{(n)}$ is needed for *regularisation*, to ensure that significant details *can be re-accessed within a tree structure*. In fact, across $g_i^{(L-1)}, \ldots,\ g_i^{(1)}$ and $g_i^{(0)}$, whenever *any* child details $\boldsymbol{d}_{2e}^{(n)}$ *or* $\boldsymbol{d}_{2e+1}^{(n)}$ is significant on $g_i^{(n)}$ its parent detail $\boldsymbol{d}_e^{(n-1)}$ on $g_i^{(n-1)}$ can only be significant and should be made accessible for possible use – later in the generation of an *assembled DG2 solution* on $g_i^A(t)$. Thus, regularisation is the act of ensuring that such sub-elements $I_e^{(n-1)}$ are also flagged as active. When many mother elements are used ($M > 1$), regularisation should also consider activating those sub-elements located at the boundaries across the elements, which is necessary to ensure that the modelling information can propagate across different elements.

Fourthly, all significant details $\boldsymbol{d}_e^{(n)}$, at a *present* time *t*, are revisited to also *predict whether their significance is likely to remain or increase at time* $t + \Delta t$, with Δt denoting the simulation time-step. Such a detail is here referred to as *extra-significant* and can be identified by:

$$\check{d}_e^{(n)} \geq 2^{\bar{m}+1}\ \varepsilon^{(n)} \tag{53}$$

In Eq. (53), $\bar{m}$ is the order-of-accuracy of the prediction operator (Cohen et al., 2003), which is chosen such that $K_{max} \leq \bar{m} \leq K_{max}+1$, with K_{max} being the polynomial-order of the DG solution. In this work, $\bar{m}$ is taken equal to 1.5, though it may be useful to note that any other choice within this range was found appropriate. When a detail $\boldsymbol{d}_e^{(n)}$ is extra-significant, the set of active sub-elements is enlarged to include, in addition to $I_e^{(n)}$, its child sub-elements $I_{2e}^{(n+1)}$ and $I_{2e+1}^{(n+1)}$. This step is necessary to ensure that no significant features in the adaptive *flow solution,* $u \in \{h+z,\ q\}$, on $g_i^A(t)$ are overlooked on $g_i^A(t\ +\ \Delta t)$ when generating future details (Section 2.3.4).

Finally, a DG2 solution on $g_i^A(t)$ can be decided by ascendingly inspecting the tree of details, starting from the coarsest details $\boldsymbol{d}_0^{(0)}$ and DG2 modes $\boldsymbol{u}_0^{(0)}$, while decoding. That is, while climbing the details tree ($n = 0, 1, 2, \ldots$ and $n \leq L-1$), Eq. (49) is successively applied to decode local DG2 modes $\boldsymbol{u}_e^{(n)}$ on active sub-elements $I_e^{(n)}$. Inspection of details is aborted under two circumstances:

(i) When a detail $\boldsymbol{d}_e^{(n)}$ switches status to becoming insignificant for the first time, with its local DG2 modes $\boldsymbol{u}_e^{(n)}$ selected for generating the *assembled DG2 solution* on $g_i^A(t)$, or otherwise

(ii) Inspection and decoding reached $g_i^{(L-1)}$ with certain details $\boldsymbol{d}_e^{(L-1)}$ remaining significant, and their local DG2 modes $\boldsymbol{u}_e^{(L-1)}$ are already decoded. Then, a last round of decoding is applied to yield the child modes $\boldsymbol{u}_{2e}^{(L)}$ and $\boldsymbol{u}_{2e+1}^{(L)}$ on $g_i^{(L)}$ for inclusion while generating the *assembled DG2 solution* on $g_i^A(t)$.

The adaptive DG2 solution can now be viewed as a series of carefully-selected DG2 modes forming an *assembled DG2 solution* on the non-uniform grid $g_i^A(t)$. Each local DG2 mode should then be updated by applying the scaled DG2 formulation as described in Section 2.3.3. Prior to this, the DG2 modes representing the water height *h* should be restored, by subtracting the modes representing the topography *z* from those of the free-surface elevation $h+z$. Then, the scaled DG2 formulation can be applied to update the DG2 modes of the main flow data $u \in \{h,\ q\}$ as previously described (Sections 2.1.3 and 2.1.4).

2.3.3. RK2-DG2 update: elevating the modes of the assembled DG2 solution to time $t+\Delta t$

By applying the scaled DG2 formulation described in Section. 2.1.4, each local mode in $\boldsymbol{u}_e^{(n)}$, relevant to the main flow data $u \in \{h,\ q\}$, is updated within a standard RK2 time stepping. While doing so, key treatments are incorporated in the RK2-DG2 update to ensure stability around sharp solution gradients, together with conservative incorporation of source terms with wetting and drying. These treatments are well-reported for the unscaled RK2-DG2 method (Kesserwani and Liang, 2012). Herein, they are re-applied with few modifications to accommodate the scaling introduced to the present DG2 method and the changes related to using the standard SWE model instead of the pre-balanced model (Liang and Borthwick, 2009), and to further exploit the details ensuring the generation of a robust (assembled) DG2 solution. These treatments are summarised in the rest of this section.

Double localisation and slope limiting: Local slope limiting is needed for *certain* slope coefficients $u_e^{1,(L)}$ of the flow variables $u \in \{h,\ q\}$. Slope limiting is a necessary process prior to each RK stage to prevent development of Gibbs phenomena around sharp solution gradients. It should only be triggered at such portions in the solution, otherwise it can degrade the conservative character of DG2 modes in any other portions of the DG2 solution, or even affect robustness (e.g. see examples within Kesserwani and Liang, 2012; Kesserwani and Liang, 2012). Therefore, double localisation is applied to cautiously restrict the application of the slope limiter to the portions of the assembled DG2 solution at which sharp gradients are about to form. The first localisation step consists of *only considering the active slope coefficients at the maximum refinement level* (L), $u_e^{1,(L)}$, for *possible* limiting. In fact, DG2 modes, $\boldsymbol{u}_e^{(L)}$, at refinement level (L) can only be *active* whenever sustained by a tree of *significant details,* as previously described in Section 2.3.2 and also proved in Vuik and Ryan (2014). When this happens, $u_e^{(L)}$ should be representative of a local feature occurring in the assembled DG2 solution. Such a local feature can either be a sharp discontinuity, i.e. a shock wave, or *shockless* representing a solution kink (e.g. a front of a rarefaction wave) or a rapidly changing state (e.g. due to a wetting and/or a drying process). Therefore, a second localisation step is needed to avoid slope limiting around any shockless feature within the assembled DG2 solution. This can be achieved by further subjecting those active slope coefficients $u_e^{1,(L)}$ to *Krivodonova's shock detector* (Krivodonova et al., 2004), which is here used with a detection threshold ≥ 9, instead of 1 (Krivodonova et al., 2004), to ensure it only detects slope coefficients associated with the presence of a sharp solution discontinuity. After double localisation, the relevant slope coefficients can then be limited by a *slope limiter function* such as the *Generalised minmod* (i.e. Eq. 2.9 in Cockburn and Shu, 2001),which is here used. Moreover, shock detection and limiting is applied component-wise on $u \in \{h+z,\ q\}$, with the component $h+z$ used instead of *h* to ensure that the presence of sharp terrain gradients will not mistakenly trigger any slope limiting on the slope coefficients representing the water height *h*. After double localisation and limiting, limited slope coefficients for *h* can be deduced from the limited slope coefficents of $h+z$, by subtracting the slope coefficients of *z*.

It may be useful to note that without *double localisation* the quality of the assembled DG2 solution – compared to the DG2 solution on a uniform grid – might undergo more significant deterioration as a result of unnecessary calls of the *Generalised minmod limiter*. In effect, the limiter tends to either zero or unnecessarily substitute the true DG2 slope coefficients. In any case, this leads to false slope coefficients being used during encoding (Eq. (48)) resulting in false details in the compressed MWDG2 solution, which would manifest themselves in a deteriorated assembled DG2 solution after decoding (Eq. (49)).

Well-balanced and depth-positivity-preserving DG2 modes: The selected DG2 modes forming the assembled DG2 solution on $g_i^A(t)$ are revised based on the wetting and drying condition described in Kesserwani and Liang (2012), which is applied here with the following changes. Firstly, Eq. (19) is used to generate the original Riemann states for the components $u \in \{h+z,\ h,\ q\}$, instead of Eq. (12) in Kesserwani and Liang (2012). Secondly, revised states for the components $u \in \{z,\ h+z,\ q\}$ are reconstructed from original states under conditions ensuring both depth-positivity and well-balancedness (i.e. using Eqs. (14)–(16) in Kesserwani and Liang, 2012). These revised states should be used to calculate Riemann fluxes across the sub-elements forming $g_i^A(t)$. Thirdly, Eqs. (13)–(14) and (21)–(22) are reused to reconstruct DG2 modes based on the revised Riemann states. Fourthly, revised DG2 modes of the *h* variable are deduced from those of the $h+z$ variable by subtracting the revised DG2 modes of the *z* variable. Finally, revised DG2 modes of $u \in \{z,\ h,\ q\}$ and Riemann fluxes become availabe to evaluate the DG2 operators (Eqs. (25)–(26)).

When applying the present wetting and drying condition, it may be useful to note two key aspects. The first is about the continuity property of the DG2 topography projection in Eq. (20). Although Eqs. (21)–(22) ensure that the continuity of the DG2 topography projection holds on a static uniform grid (Kesserwani, 2013), this property does not necessarily hold for the assembled DG2 topography projection on $g_i^A(t)$. In fact, this topography projection is subject to constant decoding (Eq. (49)) from the compressed MWDG2 solution based on coefficients (Eqs. (34)–(35) and (40)–(41) associated with decompositions from essentially discontinuous functions (Eqs. (31)–(32) and (36)–(37)). Hence, involving the free-surface elevation $h+z$ as an intermediate variable (as in Kesserwani and Liang, 2012) is found necessary to achieve wetting and drying without relying on the continuity property for the assembled DG2 topography projection on $g_i^A(t)$.

The second aspect is about a specific time-step restriction criterion to ensure depth-positivity for the average coefficients with time evolution. By denoting $(h_e^0)^t$ and $(h_e^0)^{t+\Delta t}$ the average coefficients of the water height variable at times t and $t+\Delta t$, respectively, the following formula can be obtained (using a similar reasoning as in Kesserwani and Liang, 2012):

$$\left(h_e^0\right)^{t+\Delta t} \geq [1-2\,Cr]\,\left(h_e^0\right)^t \tag{54}$$

In Eq. (54), *Cr* stands for the *Courant number* relative to the Courant–Friedrichs–Lewy condition, which restricts the time-step size Δt within explicit time integration schemes. From Eq. (54), it is clear that, whenever $(h_e^0)^t \geq 0$, Cr must be ≤ 0.5 to also ensure that $(h_e^0)^{t+\Delta t} \geq 0$. While condition (54) may be irrelevant for the RK2-DG2 method for which $Cr \leq 0.3$ (Cockburn and Shu, 2001), it is found critical to preserve the stability of its first-order finite volume variant for which $Cr \leq 1$, as described later (Section 2.4).

Scaled implicit friction term discretisation: Prior to each RK2 time step, the DG2 modes of the discharge are modified to add friction contribution as done for the unscaled DG2 formulation (i.e. see Section 2.5 within Kesserwani and Liang, 2012). The same approach is applied for the scaled DG2 method used in this work, leading to similar expressions as in Kesserwani and Liang (2012) (i.e. Eq. (36) in Section 2.5 of Kesserwani and Liang, 2012) for adding friction into the discharge slope coefficients, but without having any of the $\sqrt{3}$s due to to the use of rescaled basis functions.

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

To create new details, the updated DG2 modes, which form the assembled DG2 solution on $g_i^A(t)$, should be used to reform a compressed MWDG2 solution on $\{g_i^{(n)}\}_{n=L-1,\ldots,1,0}$. DG2 flow modes for the components $u \in \{h,\, q\}$ are only defined for the sub-elements in $\{g_i^{(n)}\}_{n=L-1,\ldots,1,0}$ that spanned $g_i^A(t)$. The other sub-elements remained inactive, hence have non-existent DG2 flow modes. In this work, *truncation* is the process of initialising zero details throughout $\{g_i^{(n)}\}_{n=L-1,\ldots,1,0}$, in particular at the inactive sub-elements to keep them subject to potential activation in the next round (i.e. while redoing the process described in Section 2.3.2). Over the active sub-elements, belonging also to $g_i^A(t)$, encoding is done by successively applying Eq. (48), level-wise in decending order. This generates new flow details from the updated DG2 modes and thereby addresses any irrelevant zeroing introduced previously by truncation. As in the pre-processing step (Section 2.3.1), encoding should be applied on the components $u \in \{h+z,\, q\}$. After truncation and encoding, a full set of new details $[\boldsymbol{d}_e^{(n)}]_{n,e}$ is available, for which an alternative set of normalised details $[\check{d}_e^{(n)}]_{n,e}$ can be produced via Eq. (50) (see Section 2.3.1). With new sets of details in place, the process (Sections. 2.3.2–2.3.4) can be repeated to evolve the adaptive solution up to a specific simulation time.

2.4. First-order variant: adaptive Haar Finite Volume (HFV1) scheme

The HFV1 adaptive solution is effectively an MWDG1 method formulated upon the same scaling and wavelet basis described in Sections 2.1–2.3, but only considering the the zeroth component of the Legendre basis, i.e. $P^0(\xi) = 1$, hence neglecting the slope coefficents. Now the local approximate solution $\mathbf{U}_h$ in Eq. (11) becomes piecewise-constant, which can be initialised by Eq. (13) and updated by the operator (17). The filter matrices are thus made of a single scalar, given by:

$$\mathbf{H}^0 = \mathbf{H}^1 = \mathbf{G}^0 = -\mathbf{G}^1 = 1/\sqrt{2} \tag{55}$$

with which Eqs. (48)–(49) are applied to encode and/or decode coefficients $\boldsymbol{u}_e^{(n)}$ and/or $\boldsymbol{d}_e^{(n)}$. These coefficients now include only one component representing the piecewise-constant averaged data. The adaptive HFV1 solution is processed as described in Section 2.3, while omitting all the routines involving slope coefficents (e.g. double localisation and limiting). Explicit first-order time marching is applied for time integration, but with Courant number not exceeding 0.5 to ensure depth-posivity (see Section 2.3.3). For comparison purposes, the highest permissible Courant number shared by the MWDG2 and HFV1 adaptive solutions, i.e. $Cr = 0.3$, is chosen to run all the simulations in Section 3.

3. Numerical tests

Seven diagnostic tests are conducted to identify and compare the behaviour of the adaptive HFV1 and MWDG2 solution schemes with reference to the standard first-order finite volume (FV1) and second-order discontinuous Galerkin (DG2) schemes on uniform grids. The first test considers a dam-break flow on a wet and flat domain with a shock wave, on which wavelet-adaptivity related issues and choices are thoroughly analysed to find a setting where the adaptive solvers are as numerically accurate as their uniform grid counterparts at the finest resolution available, while remaining computationally more efficient. In the second test, the predictive accuracy of the adaptive solvers is re-explored for dam-breaks over a dry bed to assess their sensibility in tracking dynamic flow evolution with wet-dry front propagation over frictionless and frictional beds. Shockless dam-break flows over a dry domain are examined in the third test, to further inspect the properties of the HFV1 and MWDG2 solvers in capturing a wet-dry front accelerating downhill and decelerating uphill. The fourth test introduces topography with discontinuities and kinks partially submerged below a lake-at-rest. The test is used to examine the automated mesh generation capability of the adaptive HFV1 and MWDG2 solvers, and to assess their ability to preserve well-balanced adaptive solutions with zero flow. In the fifth test, steady-state flows are explored to study the convergence property of the adaptive solvers to steady-state, and to verify further their well-balancedness for non-zero flows. The sixth test uses an oscillatory flow in a parabolic bowl to measure the numerical conservation of mass and energy in a frictionless and physically closed domain, where the solvers are subjected to a perpetually moving wet-dry fronts with periodically vanishing velocities. The final test simulates a laboratory flume experiment of a frictional dam-break flow over a trapezoidal hump, including an analysis of the trade-off between maximum refinement level and computational efficiency.

Except when clearly stated for a specific test, the following setting is used as a standard. Adaptive HFV1 and MWDG2 solution runs start from a single mother element ($M = 1$) with nine refinement levels ($L = 9$), hence yielding an adaptive grid g_i^A with number of sub-elements between $2^0 = 1$ and $2^9 = 512$. Uniform FV1 and DG2 solution runs are made at the finest resolution accessible to the adaptive solvers, hence on grid $g_i^{(9)}$ with 512 elements. All solution runs are carried out using the same basic parameters, namely $Cr = 0.3$ for the time-step selection, 10^{-4} for dry (sub-)element detection, and 9 for Krivodonova's shock detector (Krivodonova et al., 2004) with the MWDG2/DG2 solvers. All the simulation results presented here are made available for access as supplementary materials (Shaw and Kesserwani, 2018). The Fortran 2003 code used to run these tests is available for download on Zenodo (Shaw et al., 2018). Instructions for running the models and interpreting the data are provided in Appendix 1.

3.1. Dam-break flow on a wet domain with shock

Shock wave transients are characteristic of hydrodynamic flows, which are typically short-lived during a long time simulation. In reality, they could well represent an impact event perturbing the flow over the whole simulation domain. Fine mesh spacing is typically desired over a relatively short period of time when the shock occurs and propagates, but such resolution may no longer be required as the shock dissipates. To explore the characteristics of wavelet-based adaptivity within the HFV1/MWDG2 solutions with discontinuities including shocks, the classical dam-break test with a flat topography is considered. Therefore, a one dimensional frictionless and wet domain is assumed of length between $x = 0$ and $x = 50$ m with a hypothetical dam located at $x = 25$ m. The dam separates two water bodies with different initial values of the water height h. The initial conditions are a zero discharge and a discontinuous water profile given by:

$$h(x,0) = \begin{cases} 6 & if\ x \le 25 \\ 2 & if\ x > 25 \end{cases} \tag{56}$$

This results in a flow profile including a shock wave and rarefaction wave which propagate away from the initial dam position in opposite directions separated by a constant state (Toro, 2001). Assuming open domain boundaries, both waves are expected to be present by $t = 3$ s before entirely exiting the domain by $t = 10$ s. Five series of runs are performed using different solver configurations with the same initial conditions, each with a specific purpose as detailed in the following.

3.1.1. Optimal choice for the error threshold driving wavelet-adaptivity

In this first series of tests, the adaptive HFV1/MWDG2 solvers are employed to identify the *error threshold* (Section 2.3.2) that ensures a fair balance between the numerical accuracy and the computational efficiency of the adaptive solvers. Adaptive and uniform solution schemes are run for the standard setting, which yields a uniform grid with 512 elements for the FV1/DG2 solutions ($\Delta x = 0.098$ m) and an adaptive grid that can allow up to 512 sub-elements ($\Delta x^{(9)} = 0.098$ m) for the HFV1/MWDG2 solutions. To measure accuracy, the normalised ℓ^2 error is calculated while *varying the additivity error threshold* from $\varepsilon = 10^{-6}$ to $\varepsilon = 10^{-1}$ (Fig. 1a). The ℓ^2 errors are evaluated for the water height variable at $t = 2.5$ s, when both shock and rarefaction waves are still present in the domain (see Fig. 2). A normalised ℓ^2 error is calculated as:

$$\ell^2 = \sqrt{\frac{\left(h_e^{0,(L)} - h_T\right)^2 \Delta x^{(L)}}{\left(h_T\right)^2 \Delta x^{(L)}}} \tag{57}$$

where h_T is the analytical water height as described in Delestre et al. (2013). The ℓ^2 error for the adaptive solutions is always evaluated on the finest uniform grid available, namely $g_i^{(L)}$ – by prior conversion from a compressed solution on g_i^A into an assembled solution on $g_i^{(L)}$ (Section 2.2.5). In Fig. 1a, the ℓ^2 errors of the adaptive HFV1/MWDG2 solvers for various *error threshold* values are compared to the ℓ^2 errors relative to their uniform FV1/DG2 counterparts on the finest grid. These results show that both adaptive HFV1/MWDG2 solvers can preserve the ℓ^2 accuracy of the underlying uniform FV1/DG2 solvers, respectively, up to an *error threshold* value of $\varepsilon = 10^{-2}$. Particularly, for $\varepsilon \le 10^{-2}$, the errors of the MWDG2 solution remain lower than the errors of the uniform FV1 solution on the finest grid, as expected due to the second-order accurate nature of the MWDG2 solver. With $\varepsilon = 10^{-1}$, the ℓ^2 errors of HFV1/MWDG2 exceed the ℓ^2 errors of uniform FV1/DG2 counterparts on the finest grid (with 2^9 elements), although they remain bounded by the uniform FV1/DG2's errors that are two order of resolution coarser (on the grid with 2^7 elements). Nonetheless, with $\varepsilon = 10^{-1}$, the ℓ^2 error of MWDG2 is noted to exceed the ℓ^2 error of FV1 on the finest grid, making it a less compelling choice to further benefit from the DG2 accuracy. Hence, the error threshold $\varepsilon = 10^{-3}$ is found to be a rational choice to keep the predictive accuracy of the adaptive solvers at the same level as their uniform counterparts on the finest grid available, and to achieve second-order accuracy with the MWDG2 solver.

Computational efficiency is measured as the CPU time needed to complete a 40 s long simulation and including the pre-processing step (Section 2.3.1). Fig. 1b shows the CPU times for the adaptive HFV1/MWDG2 solvers evaluated for all the error thresholds used in the accuracy analysis (Fig. 1a), along with the CPU times for the uniform FV1 and DG2 simulations on the finest grid (512 elements). As the error threshold increases, the CPU time of the adaptive HFV1/MWDG2 solvers decreases initially and becomes practically constant for $\varepsilon \ge 10^{-3}$. For the considered error thresholds, the MWDG2 solver results in 2.3 to 140 times faster simulations than the uniform DG2 solver on the finest grid. In contrast, the adaptive HFV1 solver could only be faster than the uniform FV1 solver on the finest grid for $\varepsilon \ge 10^{-4}$, most likely due to dominance of the wavelet-adaptivity overhead (Section 3.1.5). On the finest uniform grid, the DG2 solver is found to be around 8 times more expensive than the FV1 solver, although the MWDG2 solver with $\varepsilon = 10^{-3}$ exhibits better performance than the FV1 solver.

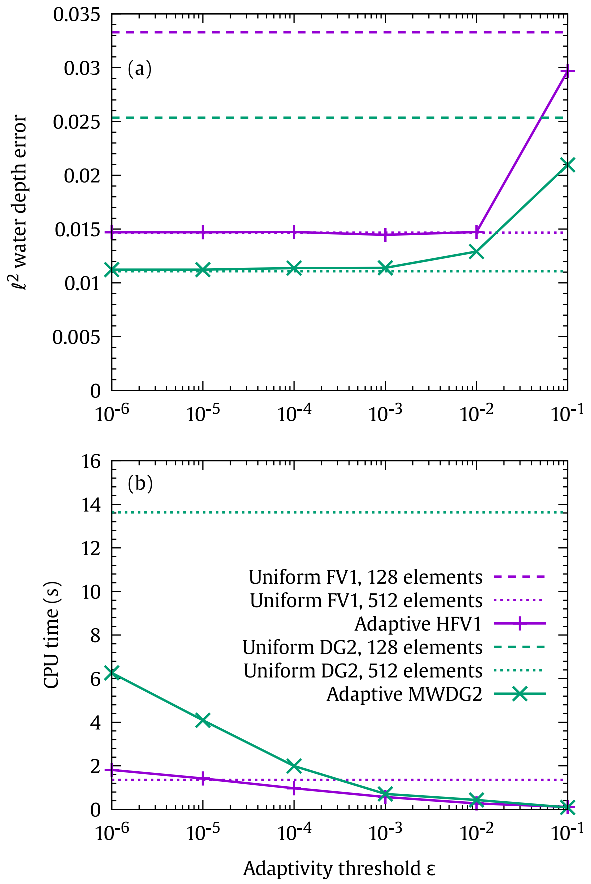


Fig. 1. Variation of (a) normalised ℓ^2 water height error at $t = 2.5$ s and (b) total CPU time for the 40 s long simulation of a frictionless dam-break on a wet domain, using adaptivity thresholds from $\varepsilon = 10^{-6}$ to $\varepsilon = 10^{-1}$. Adaptive HFV1 and MWDG2 results are obtained using a baseline mesh with a single mother element ($M = 1$) and a maximum refinement level $L = 9$. Adaptive solutions are compared with FV1 and DG2 solutions on uniform meshes with $2^7 = 128$ elements (marked by horizontal dotted lines) and $2^9 = 512$ elements (marked by horizontal dashed lines).

These tests indicate that an error threshold of $\varepsilon = 10^{-3}$ is an optimal choice for the adaptive MWDG2 solver to preserve the accuracy of the uniform DG2 solver without exceeding the runtime of the uniform FV1 solver. This choice is also suitable for the adaptive HFV1 solver to deliver simulations that are as accurate as the uniform FV1 solver but computationally more efficient. Unless stated otherwise, in the remainder of Section 3, $\varepsilon = 10^{-3}$ is adopted as a default choice for the error threshold value.

3.1.2. Adaptive solution predictability of relevant flow features ($t = 2.5$ s)

The second series of tests compares adaptive solutions of water height and discharge, and mainly examines the grid prediction ability relevant to the HFV1 and MWDG2 solvers. The adaptive solutions

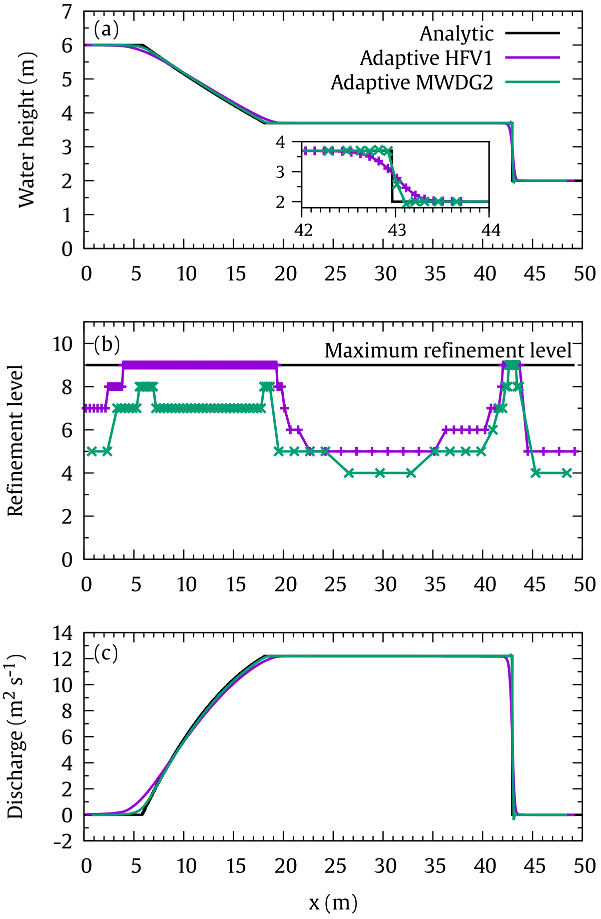


Fig. 2. Solutions of the frictionless dam-break on a wet domain at $t = 2.5$ s obtained using a baseline mesh with a single mother element ($M = 1$) and a maximum refinement level $L = 9$. Solutions obtained with the adaptive HFV1 and MWDG2 solvers are compared with the analytical solution for (a) water height h, and (c) discharge . (b) The refinement levels used by the adaptive solvers.

are analysed at $t = 2.5$ s, when both shock and rarefaction waves still exist. The adaptive solutions are illustrated in Fig. 2, which shows a good agreement with the analytical solutions. The HFV1 predictions (Fig. 2a,c) show more pronounced numerical diffusion than the MWDG2 predictions, which is in fact expected given the first-order nature of the HFV1 scheme.

In terms of resolution predictability, as shown in Fig. 2b, both HFV1 and MWDG2 correctly predict the finest resolution around the shock, i.e. refinement level (9), further showing ability to allow large gaps in resolution levels without failing. In regions of uniform flow, at the constant state and downstream of the shock, the HFV1 and MWDG2 solutions predicted the coarsest resolutions at refinement level (5) and (4), respectively. It is not surprising that MWDG2 yields coarser refinement levels than HFV1 as the former always have smaller errors than the latter for $\varepsilon = 10^{-3}$ (Section 3.1.1). Nonetheless, both HFV1 and MWDG2 solvers seem able to sensibly select suitable refinement levels for their adaptive solution in the locality of a shock and throughout the constant state (Fig. 2 for $20 \le x \le 50$). However, in prediction of the rarefaction wave, MWDG2 presents a remarkable behaviour as compared to HFV1. There, the MWDG2 solution uses refinement level (8) around the rarefaction's head and tail, preserves level (7) in between them, and allows a sharp drop to level (5) downstream of the head. Also, the MWDG2 solution does not even access the maximum refinement level (9), as opposed to the HFV1 solution that deploys it to indistinguishably compute the extent of the rarefaction. These results suggest that the wavelet-adaptivity

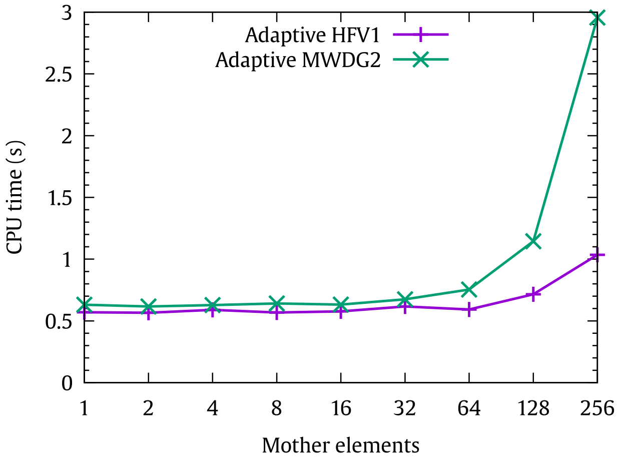


Fig. 3. CPU time to complete the 40 s long simulation of a frictionless dam-break on a wet domain. The number of mother elements and the maximum refinement level are varied together so that the adaptive grid allows maximum of 512 sub-elements.

combined with the MWDG2 solver can produce an adaptive solution that is more accurate and economical on grid resolution demands.

3.1.3. Size of coarse baseline grid vs. maximum refinement level

This third series of runs aims to analyse the trade-off between coarseness of the initial grid versus depth in maximum refinement level. A known adverse effect of conventional adaptive mesh refinement methods is the need of an initial coarse mesh that is yet fine enough for the flow solver to sense the triggering features of the initial flow conditions (Donat et al., 2014, Haleem et al., 2015), among many other adverse effects (Zhou et al., 2013, Kesserwani and Liang, 2015, Liang et al., 2015, Li, 2010, An and Yu, 2014, Delis et al., 2011). Wavelet-based adaptivity can overcome this drawback, permitting the initialisation of simulations from a very coarse initial mesh as small as two elements (Caviedes-Voullième and Kesserwani, 2015) or even a single element (Sections 3.1.1–3.1.2). To study this characteristic for the adaptive HFV1 and MWDG2 solutions, they are here reconsidered with different settings based on doubling the baseline grid size in conjunction with systematic lowering of the maximum refinement level, but on the basis of fixing the maximum allowed number of sub-elements to 512. The parameters {M, L} are varied as {M, L} = {{1,9}, {2,8}, {4,7}, {8,6}, {16,5}, {32,4}, {64,3}, {128,2}, {256,1}}, and runs are made with $\varepsilon = 10^{-3}$. As in Sections 3.1.1, the accuracy of the adaptive solvers is evaluated at $t = 2.5$ s according to Eq. (57), and their computational efficiency is assessed based on the CPU runtime taken to complete a 40 s simulation.

In terms of accuracy, the same qualitative predictions are noted for HFV1 and MWDG2 solvers, respectively, under the different setting for {M, L}. Each of the solvers show identical depth and discharge predictions, which are quite similar to those illustrated in Fig. 2a,c, and for this reason not presented here. They also yield the same number and size for the sub-element forming their assembled solutions, consistent with the profile shown in Fig. 2b. This observation is also reinforced by the fact that the same normalised ℓ^2 error magnitude (plotted in Fig. 1 for $\varepsilon = 10^{-3}$) is retrieved for all the settings.

As for the runtime efficiency, it is found to be different for each solver under the different settings. Fig. 3 shows the CPU time cost for each solver relative to each setting {M, L}. As the number of mother elements exceeds 32 (Fig. 3), the adaptive solvers experience an increase in CPU times, as expected. In fact, by $t > 10$ s, the flow domain contains very smooth profiles, for which the adaptive solvers can at best select an adaptive grid at the coarsest resolution allowable, with M elements, prior to completing the 40 s simulation (Section 3.1.4). In particular, the runtime of MWDG2 becomes significantly more costly with increasing

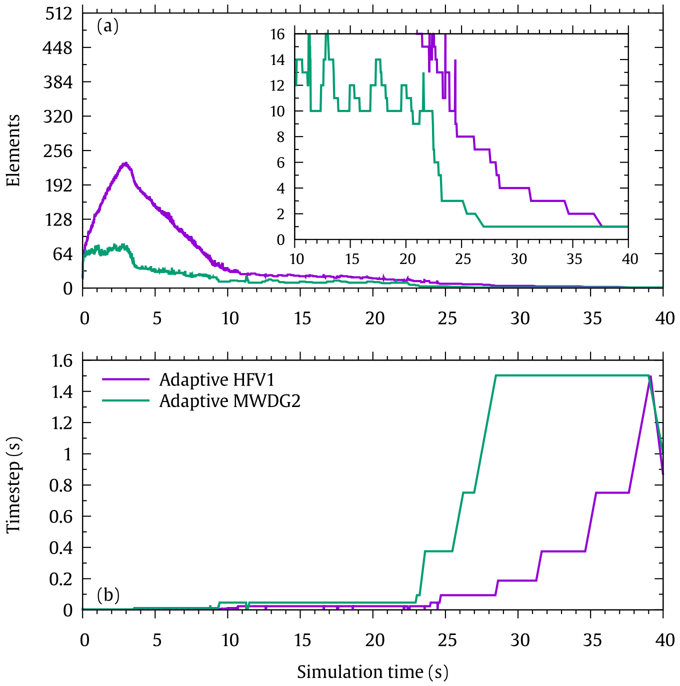


Fig. 4. Evolution of (a) element counts and (b) time-steps over the 40 s long simulation of a frictionless dam-break on a wet domain using the adaptive HFV1 and MWDG2 solvers. The baseline mesh has a single mother element ($M = 1$) with a maximum refinement level $L = 9$, hence meshes have a maximum of $2^9 = 512$ sub-elements. The inset of panel (a) plots the final 30 s of the simulation when the shock and rarefaction waves have exited the domain.

number of mother elements, to an extent that the underlying DG2 operational costs are overwhelming (Fig. 3 for $M \geq 128$). However, as long as the baseline grids do not exceed 32 mother elements, the adaptive HFV1 and MWDG2 solvers required similar runtime costs. These findings indicate that the accuracy of the adaptive solvers is not affected by severe coarsening in the baseline grid, but such an action is necessary to fully exploit wavelet-adaptivity traits to boost efficiency – in particular with MWDG2.

3.1.4. *Coarsening ability and time-step size over long time evolution*

The fourth series of runs investigates the dynamic behaviour of the adaptive solutions as the transient dam-break evolves and dissipates in the open computational domain during the 40 s simulation. The standard setting is used to re-run the HFV1/MWDG2 solvers together with the default error threshold, while inspecting their coarsening ability and the size of their time-step as time evolved. Fig. 4 shows the time history for the number of sub-elements and of the time-step size. During the presence of the rarefaction wave in the domain, $t < 10$ s, Fig. 4a reveals that the HFV1 solver requires 3 times more sub-elements than the MWDG2 solver. In line with the results in Section 3.1.2 (see Fig. 2), Fig. 4a shows that HFV1 – with its piecewise-constant basis – involved a maximum of 233 sub-elements to represent the sloping rarefaction wave, whereas MWDG2 – with its piecewise-linear basis – uses just 83 sub-elements for representing the same rarefaction wave and does that more accurately than HFV1. Beyond $t = 10$ s, the maximum number of sub-elements with MWDG2 shows much faster decrease than with HFV1 and reaches the single mother element about 10 s earlier (see zoom-in portion in Fig. 4a). This behaviour is expected with both solvers as by $t > 10$ s the waves exited the domain and only small solution perturbations remain. Relatedly, the time histories of the adaptive time-step size are illustrated in Fig. 4b, showing predominantly larger time-steps with MWDG2 than with HFV1. The first noticeable increase in time-step size for the MWDG2 solver is achieved by $t = 3.5$ s when the shock wave exits the domain. More increase in time-step size is seen by $t = 10$ s when both waves have exited the domain. This increase becomes more significant from $t > 23$ s, when MWDG2 uses less than four sub-elements. From $t > 27$ s, the MWDG2 solver uses a time-step around $\Delta t = 1.5$ s, which is roughly twice the time-step used by HFV1 over this period. This analysis supports the findings highlighted at the end of Section 3.1.2, suggesting that the MWDG2 solver is more accurate and less CPU intensive for simulations over large spatial domains and long-time scales.

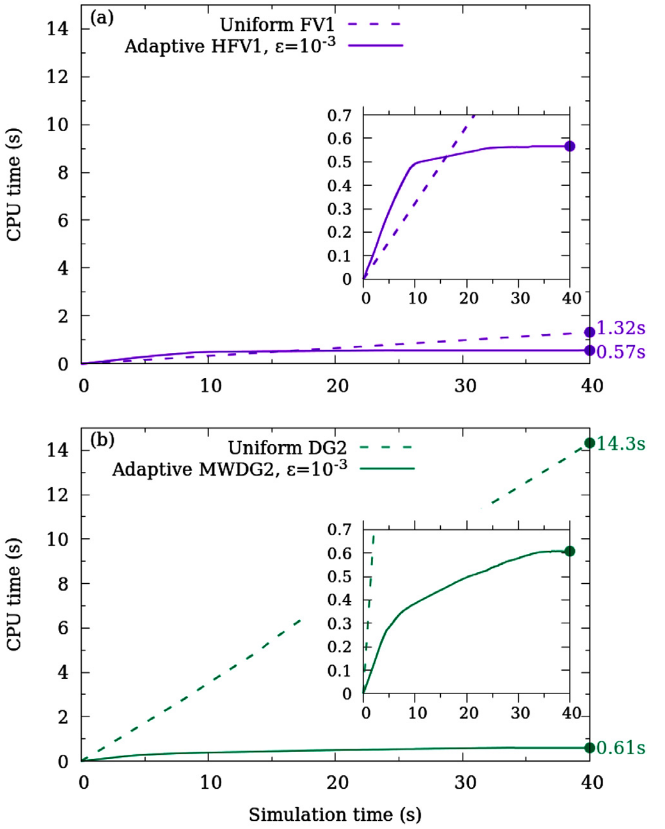


Fig. 5. CPU times for the simulation of a frictionless dam-break on a wet domain using (a) FV1 on a uniform mesh and adaptive HFV1, (b) DG2 on a uniform mesh and adaptive MWDG2. Filled circles mark the end of the simulation at $t = 40$ s. Inset plots show the first 0.6 s of CPU time during which the adaptive HFV1 and MWDG2 simulations have completed.

3.1.5. *Computational overhead due to wavelet adaptivity*

The final series of tests examines the computational overhead associated with wavelet-adaptivity in the HFV1 and MWDG2 solutions. Wavelet-adaptivity reduces the number of sub-elements, producing coarser solutions that allow longer time-steps (Section 3.1.4). Fewer sub-elements and bigger time-steps reduce the overall computational cost (Sections 3.1.1 and 3.1.4), but the compression and assembly mechanisms (via transformations (48) and (49) as detailed in Section. 2.3) involved in the adaptivity calculations introduce some computational overhead that may dominate the overall computational cost (Fig. 1b). To identify the extent of this overhead, the computational trade-off between the adaptive calculations and the uniform ones is analysed considering their cumulative CPU runtimes, respectively, throughout the 40 s simulations (Fig. 5). The adaptive and uniform solvers are run based on the standard setting.

In Fig. 5a, the evolution of the cumulative runtimes generated by the FV1 and HFV1 are compared. For the first 15 s, the adaptive HFV1 solver is found to be slower than the uniform FV1 solver due to the computational overhead associated with wavelet-adaptivity. Later, after the shock and rarefaction waves exit the domain, the adaptive HFV1 solution is coarsened aggressively (Fig. 4) and the associated gain in computational efficiency is seen to outweigh the adaptivity overhead. Nonetheless, the entire 40 s long HFV1 simulation is noted to complete in less than half the CPU time of the uniform FV1 simulation on the finest grid. This indicates that adaptive HFV1 modelling is more practical when simulating flows with smooth profiles. With the adaptive MWDG2 solver, as shown in Fig. 2b, the computational overhead due to wavelet-adaptivity remains insignificant relative to the uniform DG2 simulation. Also, this overhead is found to be lower than the wavelet-adaptivity overhead experienced in the HFV1 simulation (compare the zoom-in portions in Fig. 5a and Fig. 5b) – at least for $t < 15$ s when the rarefaction did not leave the domain. Most strikingly, the adaptive MWDG2 solver is found to complete the 40 s simulation almost as quickly as the adaptive HFV1 solver.

In summary, when simulating a dam-break flow with a shock occurring on a wet domain, the adaptive HFV1/MWDG2 solvers with $\varepsilon = 10^{-3}$ preserve the numerical accuracy of their corresponding uniform FV1/DG2 solvers. HFV1/MWDG2 are most effective on very coarse baseline grids down to a single mother element; once the waves have left the domain, both solvers are able to represent the spatially uniform solution with just one element. HFV1 is about twice as fast as FV1, and MWDG2 is about 20 times faster than DG2, with MWDG2 achieving greater accuracy than HFV1 at the same speed.

3.2. Dam-break flow on a dry domain without shock

As shown in Section 3.1, wavelet-adaptivity can easily refine the solution in the locality of a shock wave because wavelets act as a kind of jump detector (Vuik and Ryan, 2014). However, a dam-break wave usually happens over a dry domain, without experiencing shock formation when topographic effects are neglected. In this case, a wetting front propagation occurs downstream. When friction effects are also neglected the wave-front shape is smooth, including a wet-dry front that should be modelled with enough resolution to properly track arrival time. Friction retards the arrival of the wet-dry front and steepens the wave-front, which must also be captured with fine resolution to represent the wave tip. In this test, some key properties of the adaptive HFV1 and MWDG2 solvers are re-explored when simulating dam-break flows over a dry and flat bed, considering frictionless and frictional cases for which analytical or semi-analytical solutions exist (Delestre et al., 2013).

3.2.1. Frictionless case

The test configuration is the same as the dam-break on a wet domain (Section 3.1.1), except for the initial water height h, which is given by:

$$h(x,0) = \begin{cases} 6 & if\ x \le 25 \\ 0 & if\ x > 25 \end{cases} \tag{58}$$

The adaptive HFV1/MWDG2 solutions are considered with the standard setting. Tests are run for $t = 1.3$ s and normalised ℓ^2 errors are calculated, using Eq. (57) by differencing numerical solutions with the analytical solution for the same range of choices for the error threshold (between $\varepsilon = 10^{-6}$ and $\varepsilon = 10^{-1}$). Fig. 6 illustrates the respective normalised ℓ^2 errors for the HFV1/MWDG2 solvers. Fig. 6 also includes the ℓ^2 errors of the FV1/DG2 solvers on two uniform grids with $2^7 = 128$ elements and $2^9 = 512$ elements, showing lesser magnitudes with DG2 as expected. For all the error thresholds, the HFV1 and MWDG2 solution remained more accurate than the corresponding uniform FV1 and DG2 solutions on the grid with 128 elements (Fig. 6). The MWDG2 solver is always more accurate than FV1, as opposed to the previous test (compare Fig. 6 with Fig. 1a). With $\varepsilon \le 10^{-2}$, the HFV1 and MWDG2 solutions become almost as accurate as their corresponding uniform solutions on the finest grid, although they are somewhat less accurate. This behaviour is not observed in the previous test (compare Fig. 6 with Fig. 1a), where the ℓ^2 errors of the HFV1/MWDG2 solvers overlap with the ℓ^2 errors of the uniform FV1/DG2 solvers on the finest grid. Possibly, in this test, the water height and flow profiles are largely curved, which is the case where the FV1/DG2 solvers benefit more from an increase in the resolution of the uniform grid. Also, the flow states in the previous test remain unchanged over a significant portion in the domain (Fig. 2), which causes less loss of relevant information within the HFV1/MWDG2 solvers – during (de)compression due to propagation of round-off errors in Eqs. (48)–(49). Here, DG2 and MWDG2 achieved lower ℓ^2 errors than in the previous test, most likely owing to the double localisation process that switched off the slope limiter given the shockless nature of this dam-break flow. The results in Fig. 6 indicate that $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-2}$ seem to be good choices to maximise the efficiency for HFV1/MWDG2 runs and deliver comparable accuracy to the uniform FV1/DG2 runs on the finest grid.

A qualitative analysis of the adaptive HFV1 and MWDG2 solutions at $t = 1.3$ s is presented in Fig. 7a and 7b, which includes a comparison

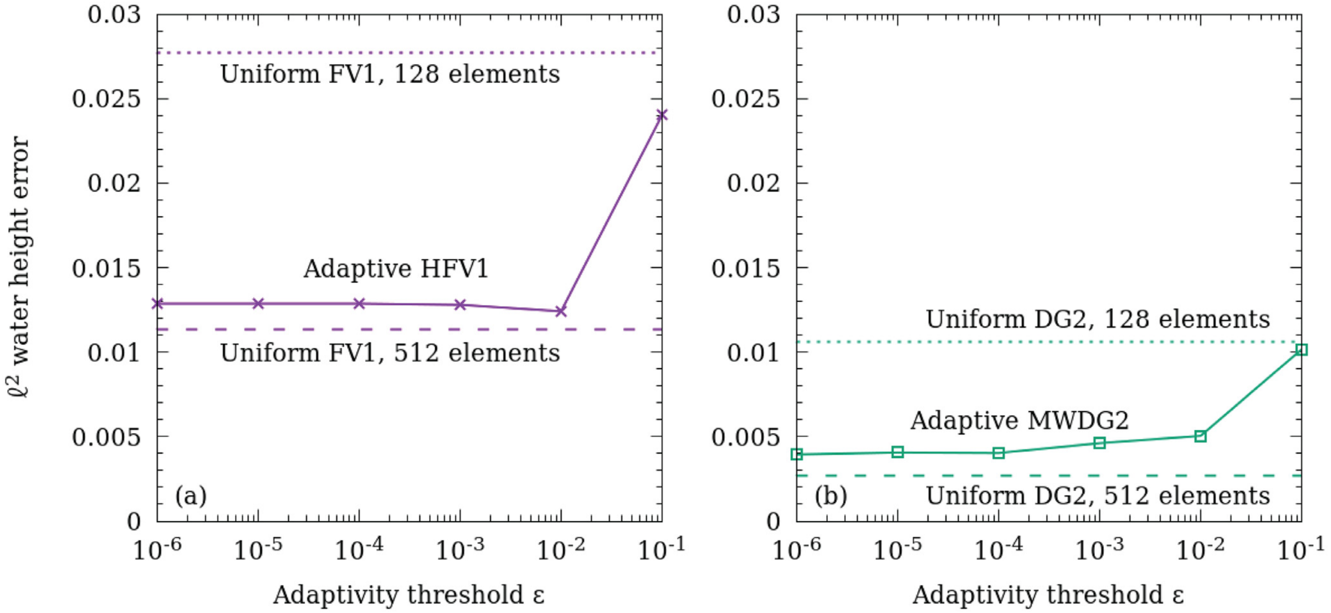


Fig. 6. Normalised ℓ^2 water depth error at $t = 1.3$ s for the simulation of a frictionless dam-break on a dry domain, using adaptivity thresholds from $\varepsilon = 10^{-6}$ to $\varepsilon = 10^{-1}$. Adaptive HFV1 and MWDG2 results are compared with those of the FV1 and DG2 solvers on uniform meshes with 128 elements (dotted lines) and 512 elements (dashed lines).

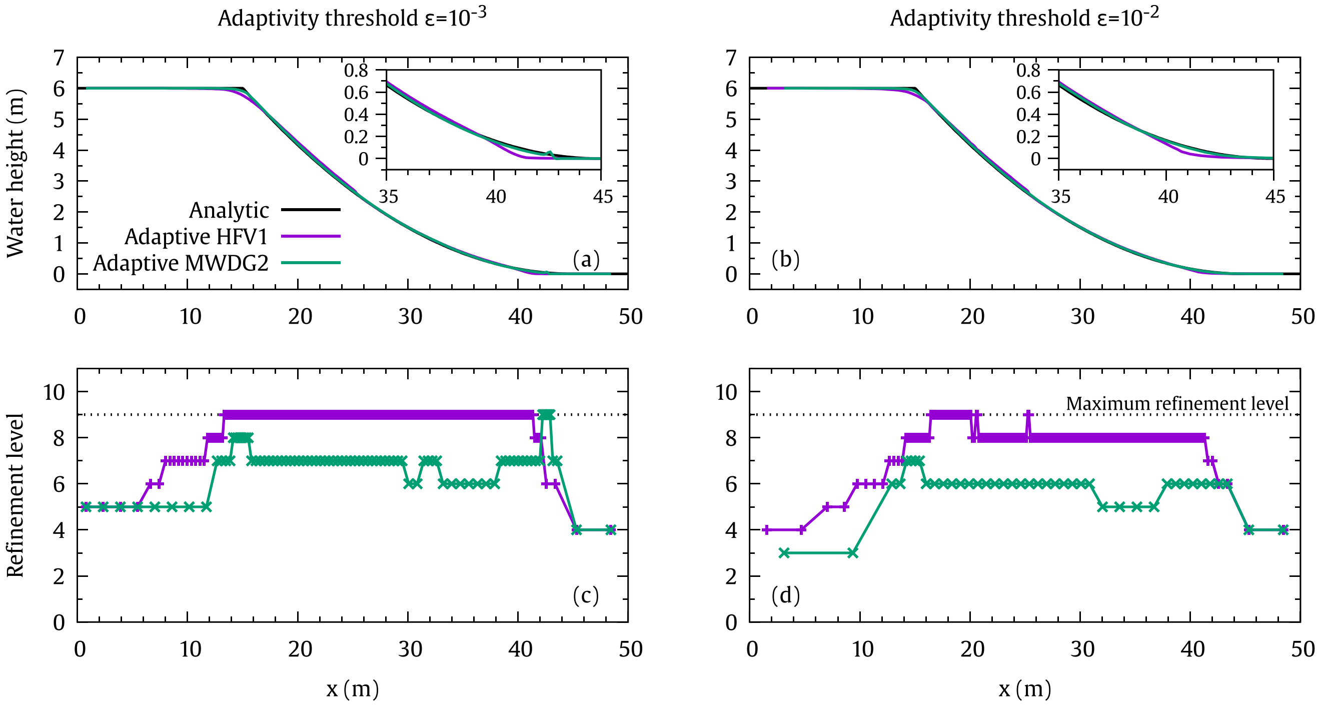


Fig. 7. Water height at $t = 1.3$ s for the simulation of a frictionless dam-break on a dry domain, comparing the analytic solution with numerical solutions for the adaptive HFV1 and MWDG2 solvers with an adaptivity threshold (a) $\varepsilon = 10^{-3}$ and (b) $\varepsilon = 10^{-2}$. (c, d) Refinement levels for the corresponding solutions. Simulations are performed on a baseline mesh with a single mother element and a maximum refinement level $L = 9$ marked by a horizontal dotted line.

between the water height profiles predicted by HFV1 and MWDG2 for the aforementioned error thresholds and the analytical solution. HFV1 and MWDG2 predictions are noted to be in good agreement with the analytical solution. However, the HFV1 solution is seen to experience numerical diffusion at the wet-dry front and at the tail of the wave, slightly overestimating the region upstream of the initial dam position and underestimating the position of the wave-front (see magnified portions within Fig. 7a and 7b). These effects do not seem to improve when lowering the error threshold from $\varepsilon = 10^{-2}$ to $\varepsilon = 10^{-3}$ and are not visible in the MWDG2 solution, which provides better overall alignment with analytical solution as expected from a second-order accurate numerical model.

In terms of resolution demand, as illustrated in Fig. 7c and 7d, MWDG2 allows coarser refinement levels than HFV1 and chooses more sensibly where to use the finest levels. With $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$, the HFV1 solution involved the two finest refinement levels, namely still accessing levels (8) and (9) to represent the full extent of the sloping water surface (Fig. 7c and 7d). The MWDG2 solution does not exceed levels (7) to represent this zone except where it should, namely at the kink and wet-dry front. Notably, with $\varepsilon = 10^{-2}$, MWDG2 uses level (6) and below along the smoothing wave, level (7) at the kink, but without accessing any higher refinement levels despite being available. Considering also that MWDG2 predictions are nearly similar at $\varepsilon = 10^{-2}$ and $\varepsilon = 10^{-3}$ (see Fig. 6b and compare Fig. 7a vs. 7b), lowering ε can reduce model accessibility to the finest refinement levels, as desired for some simulations that do not demand high resolution, while keeping these finest levels re-accessible as needed for other simulations (see also Section 3.7).

3.2.2. Frictional case

For the frictional dam-break case, the configuration is identical, except that the Manning coefficient $n_M = 0.016$ m$^{1/3}$ s^{-1}, which is selected by calibration to fit the semi-analytical solution available in terms of the Chézy factor (Delestre et al., 2013). Adaptive HFV1 and MWDG2 solutions are produced for the same error thresholds $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-2}$, which are illustrated in Fig. 8a and 8b, respectively, together with the semi-analytical solution at $t = 1.3$ s. Outside of the wave tip region upstream of the wet-dry front, HFV1/MWDG2 solutions perform very similarly to those in the corresponding frictionless test (Section 3.2.1). At the wave tip region, the semi-analytical solution is actually based on interpolation assuming a parabola (Delestre et al., 2013). As such, no exact comparisons can be made therein. Nevertheless, HFV1/MWDG2 solutions are found to agree well with the semi-analytical solution in the wave tip region, with MWDG2 producing a steeper wave-front profile. Fig. 8c and 8d illustrate the corresponding refinement levels used by the adaptive solvers with $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-2}$, respectively. The adaptive HFV1/MWDG2 solutions show almost the same behaviour for the refinement levels as the frictionless case (compare Fig. 7c and 7d with Fig. 8c and 8d, respectively). However, at the wet-dry front, MWDG2 retains the maximum refinement level, even with $\varepsilon = 10^{-2}$, due to the steeper wave-front induced by friction.

The frictional and frictionless dam-break tests demonstrate further the ability of the adaptive HFV1 and MWDG2 solvers to simulate the propagation of dynamic waves over a dry domain. MWDG2 alleviates the numerical diffusion errors expected in the FV1 or HFV1 solutions with much lower refinement levels. With an error threshold of $\varepsilon = 10^{-2}$, MWDG2 does not need to access the maximum refinement level, apart at the wet-dry front when the wave-front is steepened by friction. This suggests that the error threshold can be further relied on to reduce model access to the finest resolutions available as relevant for certain simulations, even when they are set to perform at very high resolution.

3.3. Dam-break flow descending and ascending sloping and dry beds

In this test, the performance of the adaptive HFV1/MWDG2 solvers is further examined for dam-break flows featuring a wet-dry front that accelerates or decelerates as it descends or ascends a sloping bed. A dam-break wave upsloping is initially used in Xing et al. (2010). A more challenging variant is considered here, as proposed in Kesserwani and Liang (2012), including a case where the wave downslopes. The initial dam is assumed centred at $x = 0$ m in a [−15 m, 15 m] domain. Upstream of the dam ($x < 0$), the initial water elevation $h + z$ is equal to 8 m and the water height is assumed to be zero downstream of the dam ($x \geq$

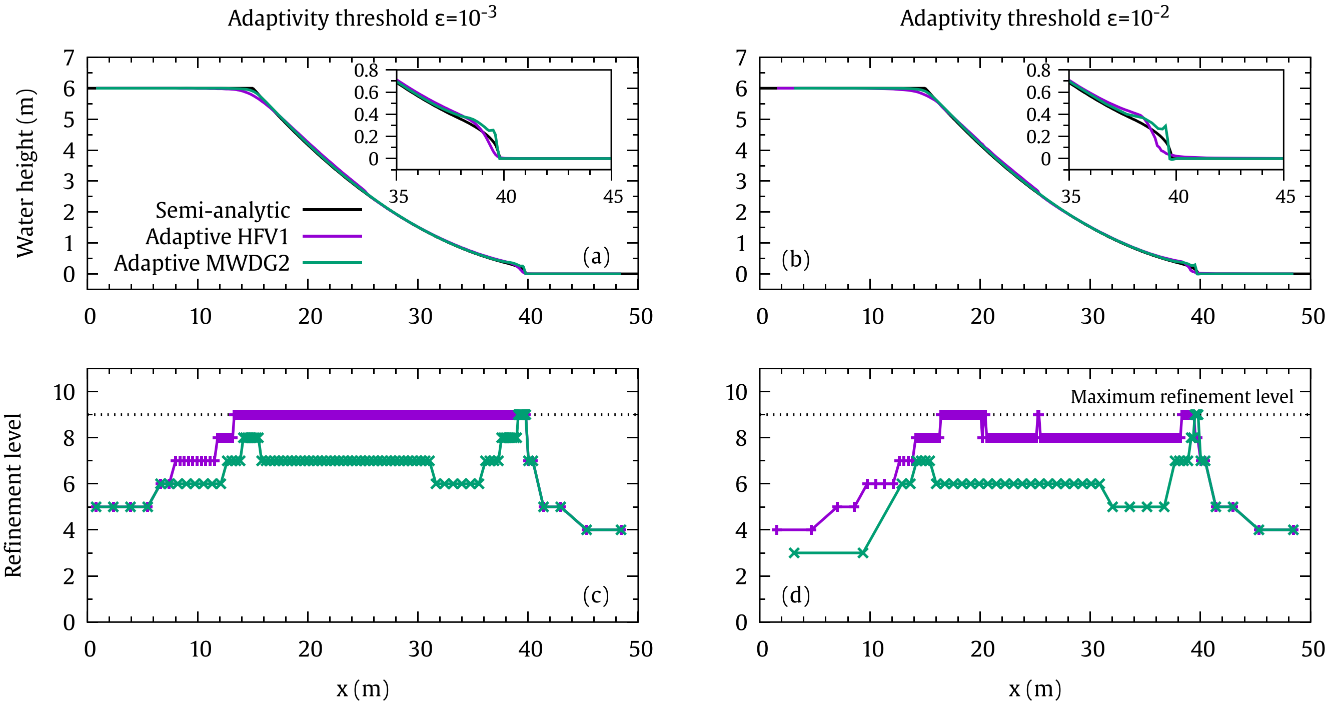


Fig. 8. Water height at $t = 1.3$ s for the simulation of a frictional dam-break on a dry domain, comparing the semi-analytical solution with numerical solutions using the adaptive HFV1 and MWDG2 solvers with an adaptivity threshold (a) $\varepsilon = 10^{-3}$ and (b) $\varepsilon = 10^{-2}$. (c, d) Refinement levels for the corresponding solutions. Simulations are performed on a baseline mesh with a single mother element and a maximum refinement level $L = 9$ marked by a horizontal dotted line.

0). A wall is assumed to exist at the upstream end ($x = -15$ m), which can be accounted for by reflective boundary conditions. Free outflow is assumed at the downstream end ($x = 15$ m) by transmissive boundary conditions. The topography is linear with a slope angle α, namely:

$$z(x) = -1 + x\, tan(\alpha) \tag{59}$$

Two cases are considered with α values in Eq. (59). First, a dam-break ascending with $\alpha = \pi/6$ and, second, a dam-break descending with $\alpha = \text{-}\pi/6$. The upslope dam-break is simulated for $t = 1$ s whereas the downslope dam-break is simulated for $t = 0.75$ s. Both cases are assumed frictionless. Simulations are performed using the standard setting with the uniform FV1 and DG2 solvers (on a grid with 512 elements) and with the adaptive HFV1 and MWDG2 solvers taken with the default error threshold ($M = 1$, $L = 9$ and $\varepsilon = 10^{-3}$).

In Fig. 9a and 9b, the water depth predictions made by the adaptive HFV1/MWDG2 and uniform FV1/MWDG2 solvers are illustrated, showing comparable profiles that also match existing results (Kesserwani and Liang, 2012). The difference between the predictions is more noticeable for the discharge profiles as shown in Fig. 9e and 9f. Compared to MWDG2/DG2, FV1/HFV1 predictions exhibit numerical diffusion at the start of the wave, as expected given the difference in the accuracy orders between the corresponding numerical formulations. Despite this, these discrepancies are more prominent for the upslope dam-break case (see $x = -11$ m in Fig. 9e vs. at $x = -6$ m in Fig. 9f) suggesting that the second-order variants provide better predictions with increased level of vigour in the wave propagation. At the wave-front, the discrepancies become more noticeable in both the upslope and downslope dam-break cases (see $x > 10$ m in Fig. 9e vs. at $x = 12$ m in Fig. 9f). Therein, informed further by the results in Fig. 7a, MWDG2/DG2 are expected to more accurately follow the evolution of the wet-dry front as they both deploy piecewise-linear solutions to integrate topography and wetting and drying, as opposed to HFV1/FV1 that use piecewise-constant solutions.

In terms of refinement level predictions, which are illustrated in Fig. 9c and 9d, the HFV1 solution only used the maximum level (9), hence yielding identical results to those delivered by the FV1 solution in both upslope and downslope dam-break case. This over-prediction is associated with the use of a piecewise-constant basis in HFV1 that yields a staircase pattern for the linear topography approximation, making the solver trigger the maximum refinement level at $\varepsilon = 10^{-3}$. Note that the proposed wavelet-adaptivity formulations indistinguishably use the details of the flow and topography variables to generate the adaptive solution. In contrast, the MWDG2 solver, in both cases, predicted refinement level (8) to track the start of the wave, and levels (6) and (7) thereafter upstream of the wave-front. For the upslope dam-break case, MWDG2 does not access the maximum refinement level (9) at the wave-front but uses refinement level (8) instead. This is in contrast with the downslope case where level (9) is retained therein, and level (4) is selected before upstream of the depression wave. Such differences in refinement level predictions are expected given the different flow physics involved in the upslope and downslope dam-break cases; namely, the wet-dry front advance is slower in the former case, whereas wave recession at the start is delayed in the latter case.

The propagation of the wet-dry front in the numerical simulations can be compared to the analytical position of the wet-dry front $x_f(t)$ given by:

$$x_f(t) = 2t\sqrt{8g\cos(\alpha)} - 1/2\, g\, t^2 \tan(\alpha) \tag{60}$$

The numerical position of the wet-dry front is calculated based on the first (sub-)element at which the water height is bigger than 10^{-2} m scanning (sub-)elements from left to right. Fig. 10a and 10b show the time evolution of wet-dry front positions for the upslope and downslope dam-break cases, respectively. As seen in Fig. 10, FV1 calculates a slower front advance consistently under-predicting the analytical solution. By the end of the simulations, FV1 (and identically HFV1) positions the front about 2 m and 1 m below the true position for the upslope and downslope dam-break cases, respectively. The DG2 solver tracks the upslope and downslope wet-dry fronts more accurately than the FV1 solver, however showing an over-predictive tendency. The adaptive MWDG2 solver is seen to preserve the accurate solution of the underlying DG2 solver. The frontal evolution obtained with the DG2 and adaptive MWDG2 solvers compares favourably with results using

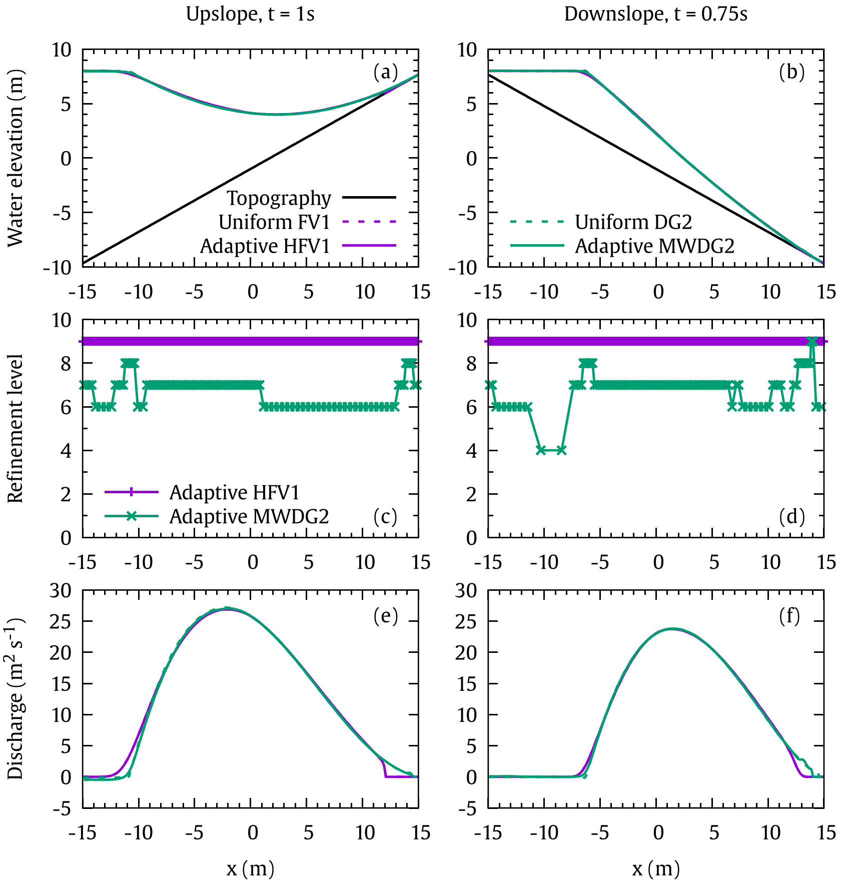


Fig. 9. Numerical solutions of (a, b) water elevation and (e, f) discharge for dam-breaks ascending upslope (left-hand panels) and descending downslope (right-hand panels) over a bed with a constant slope. Tests are performed using FV1 and DG2 solvers on a uniform mesh, and adaptive HFV1 and MWDG2 solvers. For the adaptive solvers, (c, d) illustrate the refinement levels associated with the corresponding numerical solutions.

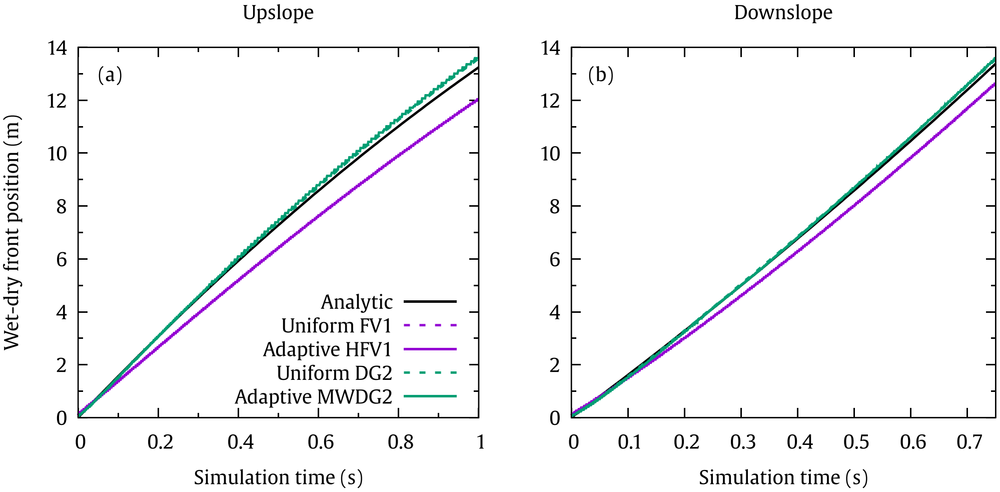


Fig. 10. Evolution of the wet-dry front for dam-breaks (a) ascending upslope and (b) descending downslope over topography with a constant slope.

the RKDG2-LFT solver presented in Kesserwani and Liang (2012).[1] In summary, the adaptive HFV1 solver is not found as effective as in the previous dam-break tests on flat beds because of its piecewise-constant basis that can yield over-refinement when approximating a sloping topography profile. The adaptive MWDG2 solver uses a piecewise-linear basis that can exactly represent the sloping topography at any refinement level, so the MWDG2 solver is able to coarsen more effectively than HFV1 while proving more accurate and economical.

3.4. Well-balanced property and mesh generation ability

This test examines the initial mesh generation ability of the adaptive solvers and their well-balanced property in reproducing a lake-at-rest. Unlike the idealised sloping topography in the previous test, real terrain is fractally multi-scale, non-smooth, and often discontinuous, as in the presence of buildings. Preserving quiescent flow over an irregular topography is challenging for numerical shallow water models, in particular at partially wet zones located at bed discontinuities (Kesserwani, 2013, Kesserwani et al., 2018). To assess the full extent of well-balancedness, a lake-at-rest test has been proposed (Sharifian et al., 2018) based on an idealised topography with smooth, sloping and discontinuous regions (see Fig. 12).

[1] In their Fig. 4b, the analytical front evolution plot for the downslope case is incorrect. Their numerical results are more closely aligned with the correct analytical front evolution presented here in Fig. 10b.

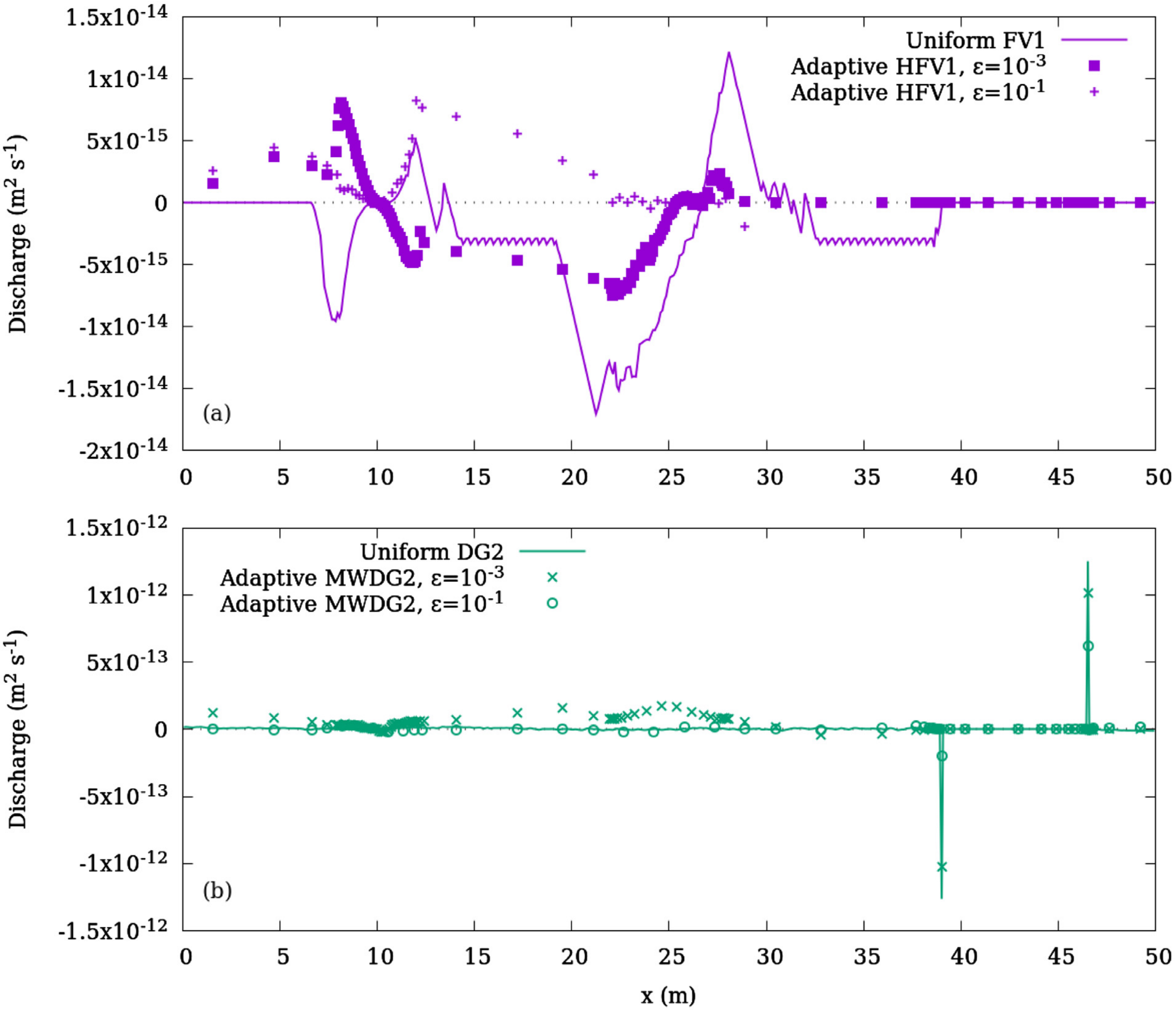


Fig. 11. Discharge after $t = 100$ s for the simulation of the lake-at-rest using (a) the FV1 solver on a uniform mesh and the adaptive HFV1 solver, (b) the DG2 solver on a uniform mesh and the adaptive MWDG2 solver. The analytical solution remains at rest with zero discharge while the numerical discharge is close to machine precision in all cases.

The lake-at-rest is defined on [0 m, 50 m] with an initial water elevation $h + z = 2$ m such that three scenarios occur: exactly dry at a peak ($h = 0$ m at the curved hump), submerged portion ($h > 0$ m at the triangular hump) and unsubmerged portion with two wet-dry fronts ($h < 0$ m at the rectangular hump). The adaptive and uniform solvers are applied to compute the lake-at-rest conditions with zero initial discharge ($q = 0$ m^2 s^{-1}). Simulations are executed for a relatively long time evolution, namely $t = 100$ s corresponding to about 16,000 time-steps, considering two error thresholds $\varepsilon = 10^{-1}$ and $\varepsilon = 10^{-3}$ with the standard setting ($M = 1$ and $L = 9$). A robust and well-balanced solver should preserve the initial water state and the initial zero discharge unperturbed as time evolves.

Fig. 11 shows the discharges computed by the adaptive and uniform solvers. All the numerical discharges are observed to be very close to machine precision (Fig. 11) and the initial water elevation remains unchanged (Fig. 12) for all the solvers throughout the simulation. Slightly larger discharge predictions are noted with MWDG2 at $\varepsilon = 10^{-3}$ than with MWDG2 at $\varepsilon = 10^{-1}$ (Fig. 11b) and with HFV1 (Fig. 11a). This behaviour is expected as the smaller the ε, the more MWDG2 will access Eqs. (48)–(49), causing more knock-on effects due to rounding of the irrational numbers involved in the filter banks. Nonetheless, this increase in error is negligible even after very long time evolution. Fig. 11b also shows two spikes in the discharge predictions occurring around the discontinuities of the rectangular hump for DG2 and MWDG2 at $\varepsilon = 10^{-3}$. These spikes, however, do not grow over the 100 s long simulation, and their magnitude is noted to be smaller with grid coarsening (e.g. compare with the MWDG2 predictions at $\varepsilon = 10^{-1}$). These results confirm that the adaptive HFV1/MWDG2 solvers are well-balanced. Noting also that the negative water height below the rectangular hump remains unmodified with time evolution (Fig. 12), the sharp-edges of the rectangular hump effectively become (internal) boundaries, which there is no need to manually recognise since the initial water elevation can intersect the topography without affecting the well-balancedness of the solution. This property seems therefore to be instrumental to deal with the presence of buildings during the mesh generation process.

Since $h + z$ and q are unvarying in this test, the assembled initial (adaptive) solution is solely selected driven by the topographic features. The well-balanced HFV1/MWDG2 solvers can therefore be used as mesh generators subject to choosing an error threshold. The mesh generation ability of these solvers is particularly explored by further analysing their refinement level predictions. Fig. 12a and 12b include the refinement levels predicted by the HFV1 and MWDG2 solvers, respectively. At the rectangular hump, both HFV1 and MWDG2 solvers are seen to select the maximum level (9) at the sharp edges, and to coarsen effectively

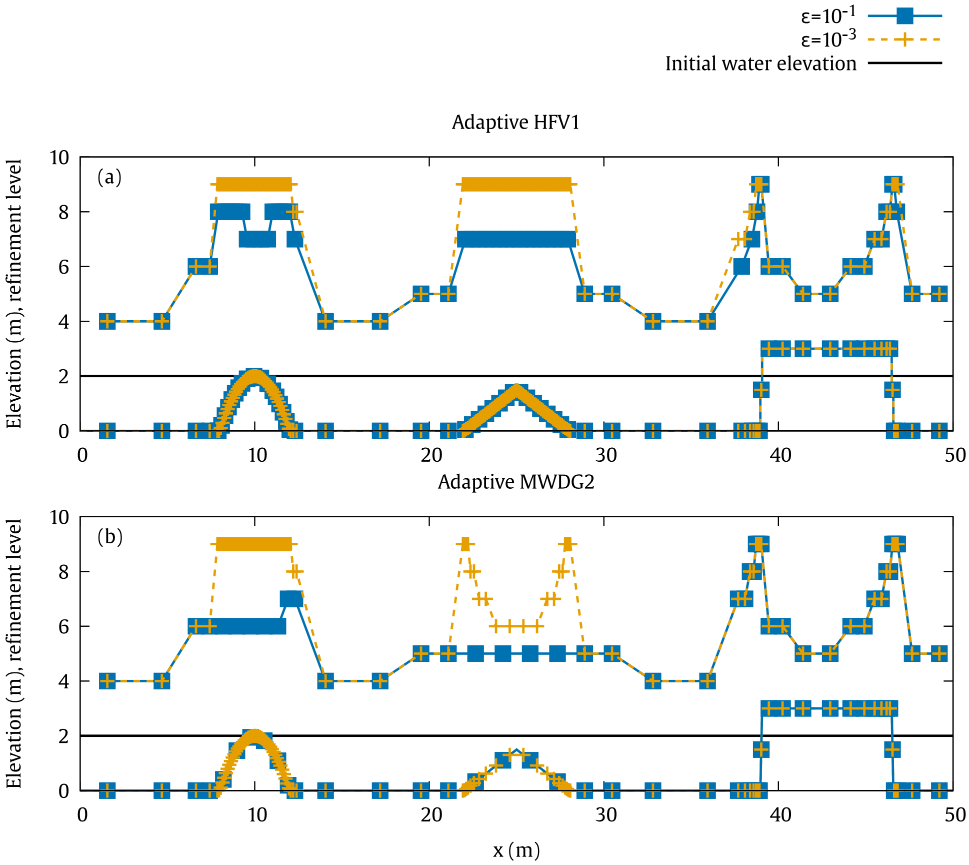


Fig. 12. Topography profiles for the simulation of the lake-at-rest using (a) the adaptive HFV1 solver, (b) the adaptive MWDG2 solver. The idealised topography has a smooth, curved hump (left), triangular hump (centre) and discontinuous, rectangular hump (right). The water elevation, topography profile and corresponding refinement levels are plotted on the same axis. Solutions are obtained using a baseline mesh with a single mother element and a maximum refinement level $L = 9$. Markers show cell centre positions, and the full, piecewise representation of topography is plotted.

in-between them where the topography is smooth. For this hump, the smooth portion is flat and the sharp-edged portions are strongly discontinuous. The former portion is readily represented by coarse piecewise-constant and piecewise-linear data, while the latter portion can easily be detected by both representations. The choice of the error threshold seems to have little effect on representing this obstacle, as very similar refinement levels are predicted therein by both HFV1 and MWDG2 solvers at $\varepsilon = 10^{-3}$ and at $\varepsilon = 10^{-1}$.

The curved and triangular humps are less easily represented by the HFV1 piecewise-constant basis: at $\varepsilon = 10^{-3}$, HFV1 used the maximum refinement level (9) in these two regions (Fig. 12a). More effective coarsening at these two humps is noted by choosing $\varepsilon = 10^{-1}$ where HFV1 uses only refinement levels (8) or below. MWDG2 coarsens the triangular hump much more sensibly than HFV1 at $\varepsilon = 10^{-3}$: it uses the maximum refinement level only at the kinks at the base of the triangle (Fig. 12b), and much coarser levels at the tip that is positioned exactly at the centre of the domain. At the curved hump, MWDG2 still predicts the maximum refinement level (9), even at $\varepsilon = 10^{-3}$, which could be signalling that more resolution is needed to cover curved terrain shapes. With $\varepsilon = 10^{-1}$, the triangular and curved hump are relatively less-resolved with MWDG2 than with HFV1, with MWDG2 predicting level (7) and below. However, taking $\varepsilon \geq 10^{-1}$ is likely to make the HFV1 or the MWDG2 solvers unable to preserve enough accuracy (recall Sections. 3.1.1 and 3.2.1).

With a maximum refinement level $L = 9$ and an error threshold $\varepsilon = 10^{-3}$, MWDG2 used the maximum refinement level at the discontinuities of the rectangular hump and the kinks of the triangular hump as expected, but also throughout the curved hump. To explore whether the usage of level (9) throughout the curved hump is an over-refinement or a requirement, the MWDG2 solver is re-run by increasing the maximum refinement level to $L = 14$ under the same error threshold. Fig. 13 shows the profile of the corresponding refinement levels. Remarkably, now the MWDG2 solver only accesses the maximum refinement level (14) at the strong discontinuities of the rectangular hump. At the kinks, MWDG2 predicts level (12) for the triangular hump and level (13) for the curved hump that has steeper kinks. Moreover, analysis of the MWDG2

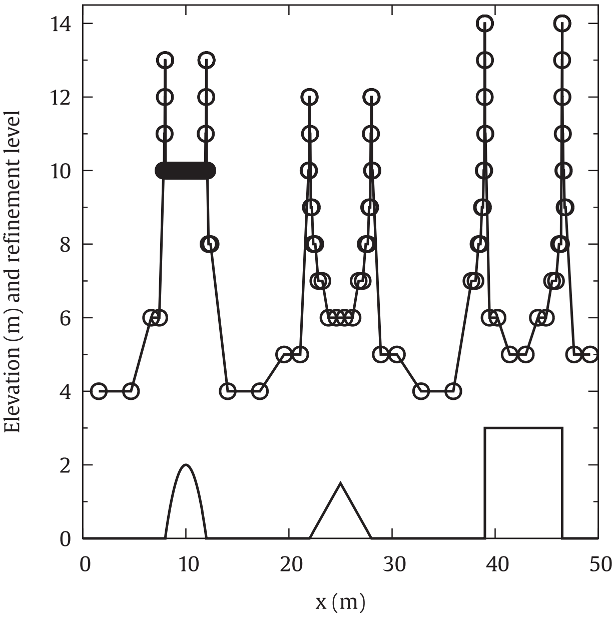


Fig. 13. Adaptive MWDG2 topography profile and corresponding refinement levels for the three humps used in the lake-at-rest simulation. The profile is obtained using a baseline mesh with a single mother element ($M = 1$) and a maximum refinement level $L = 14$.

Table 1
Initial water depth and boundary conditions for the subcritical, supercritical and transcritical steady-state tests. All steady-state tests have an initial discharge $q = 0$ m^2 s^{-1}.

Steady flow test	Initial water height (m)	Upstream discharge (m^2 s^{-1})	Upstream water height (m)	Downstream water height (m)
Subcritical	2.0	4.42	–	2.0
Supercritical	2.0	25.0567	2.0	–
Transcritical with shock	0.33	0.18	–	0.33

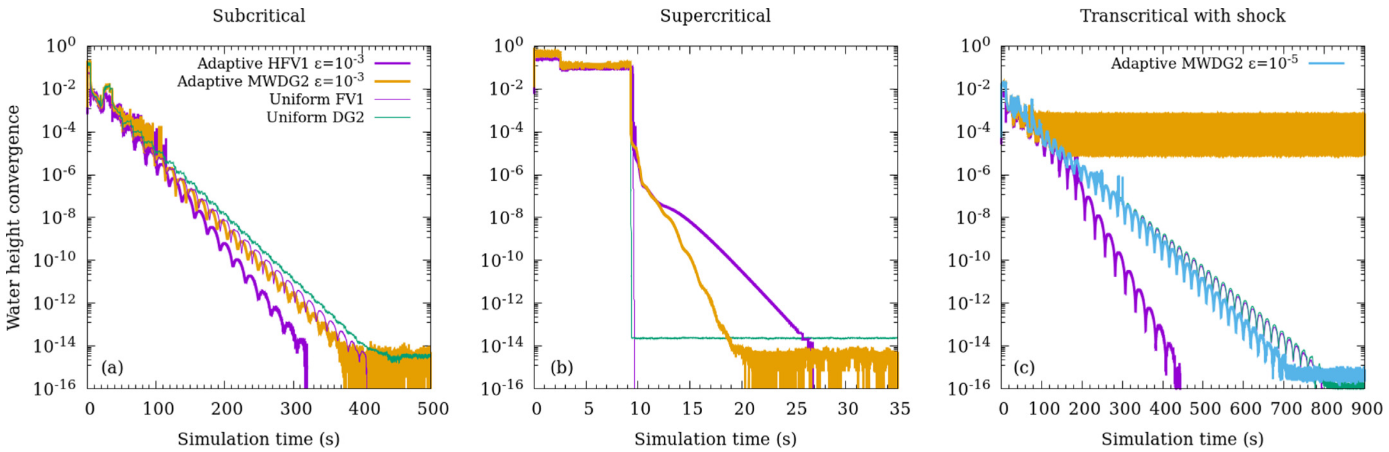


Fig. 14. Convergence to a steady-state solution for (a) subcritical (b) supercritical and (c) transcritical flows. Water height convergence is measured by calculating the ℓ^2 difference between the current and previous time-steps.

solution provides information on the necessary refinement levels required to represent the smooth humps, i.e. suggesting the need for level (6) and (10) to discretise the slope and curvature involved in the triangular and curved humps, respectively. These results imply that MWDG2 can effectively be used to initialise mesh resolution in a localised manner as needed. This property could potentially be useful towards making more effective use of very high resolution Lidar data without overloading the simulation, and gives the user direct control over the extent of resolution deepness at which topography is represented within the model (via choosing ε).

3.5. Convergence to well-balanced steady states with non-zero flows over a hump

In this series of tests, the adaptive HFV1 and MWDG2 solvers are given steady boundary conditions to study their convergence ability in reaching steady states with flows over a hump. Following Delestre et al. (2013), the one-dimensional domain is [0 m, 25 m] with a topographic hump given by:

$$z(x) = \begin{cases} 0.2 - 0.005(x-10)^2 & if \;\; 8\,m < x < 12\,m \\ 0 & elsewhere \end{cases} \tag{61}$$

Tests are performed to assess the rate of convergence upon three steady flow regimes: subcritical, supercritical and transcritical with a stationary shock. The initial and boundary conditions used in each tests are available in Table 1. Simulations are performed with the uniform FV1 and DG2 solvers and the adaptive HFV1 and MWDG2 solvers both taken with the standard setting and $\varepsilon = 10^{-3}$. A simulation is set to stop whenever the ℓ^2 difference in water height between the current and previous time-steps becomes in the range of machine precision. The time history of the ℓ^2 difference for all three tests are shown in Fig. 14.

The FV1, DG2, HFV1 and MWDG2 solvers all converge to machine precision in the subcritical test (Fig. 14a) and supercritical test (Fig. 14b). For the subcritical test, all solvers converge to machine precision within about 300–500 s, with the HFV1 and MWDG2 solvers being slightly faster than their FV1 and DG2 counterparts (Fig. 14a). In the supercritical test, the FV1 and DG2 solvers converge after about 10 s, with the adaptive solvers converging slightly later (Fig. 14b). Compared to the supercritical case, converging to steady subcritical flow takes longer because the flow is relatively weak and adjustment towards balance is consequently slower. The transcritical case involves a transition from subcritical to supercritical flow, with another transition back to subcritical flow downstream of a stationary shock. Unsurprisingly, convergence to this transcritical steady-state is the slowest of all three cases (Fig. 14c): FV1 and DG2 solvers on a uniform mesh converge to machine precision after about 800 s, and the adaptive HFV1 solver after about 450 s. The adaptive MWDG2 solver does not converge beyond 10^{-4} with $\varepsilon = 10^{-3}$. This stagnation in ℓ^2 difference with MWDG2 at $\varepsilon = 10^{-3}$ is likely due to the intrusion of the *slope limiter* triggered by noise eventually accumulating from rounding of irrational numbers at the same location (see also the related discussion in the next paragraph). Regardless, when ε is reduced to 10^{-5} the MWDG2 solver converges to machine precision at a faster rate than the DG2 solver (Fig. 14c). Overall, convergence rates for all solvers are of the same order of magnitude for a given flow regime, and all solvers are able to converge to machine precision.

The steady-state solutions of water elevation and discharge are included in Fig. 15. For all three flow regimes, the numerical solutions of water height are in close agreement, all showing no visual difference with their corresponding analytical profiles (Delestre et al., 2013), which were not illustrated for clarity. As can be seen in Fig. 15g–15i, anomalies in discharge predictions are apparent in the FV1 and HFV1 solutions. These anomalies are usually expected to reduce with an improved FV-based topography discretisation technique apart where a shock develops (Kesserwani, 2013, Haleem et al., 2015, Caleffi and Valiani, 2017). However, all these types of anomaly do not appear when using DG2 and MWDG2 solvers. Compared to the DG2 uniform solver, the MWDG2 solver presents some tiny anomalies in the discharge predictions. These anomalies are different to those induced by the HFV1 and FV1 solvers and are comparatively negligible. They are seen to occur at locations where there are gaps in refinement levels (see also Fig. 15d–15f). Most likely, these tiny anomalies are caused by constant (de)compression of the MWDG2 solution at the same location when the adaptive grid and solution become static in time. This can eventually lead to low levels of noise due to accumulation of round-off errors, which

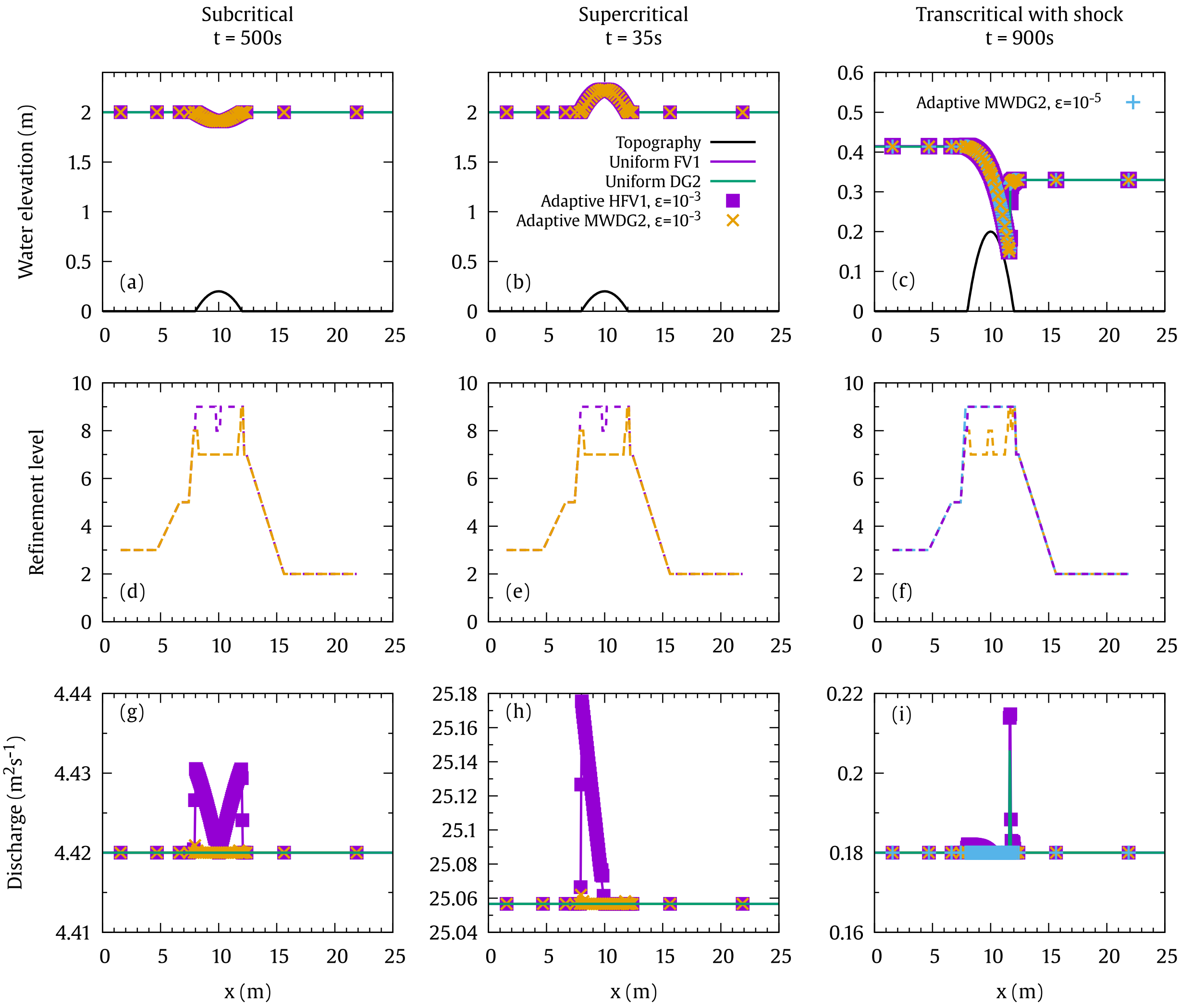


Fig. 15. Steady state solutions of (a, b, c) water elevation and (g, h, i) discharge for subcritical flow (left), supercritical flow (centre) and transcritical flow with a stationary shock (right). For the adaptive HFV1 and MWDG2 solvers, (d, e, f) show the corresponding refinement levels. All adaptive solutions are plotted using an adaptivity threshold $\varepsilon = 10^{-3}$. For the transcritical case, an additional solution is plotted using the adaptive MWDG2 solver with $\varepsilon = 10^{-5}$.

can generate knock-on effects such as triggering the *slope limiter* as discussed in the previous paragraph. Such tiny noises can be avoided by either increasing the convergence tolerance, or lessening ε.

In Fig. 15, the corresponding refinement levels predicted by the adaptive HFV1 and MWDG2 solutions are shown for the subcritical case (Fig. 15d), supercritical case (Fig. 15e), and transcritical case (Fig. 15f). Both solvers require higher refinement levels only in the locality of the hump, with very few sub-elements involving the maximum refinement level (9), corresponding with $\Delta x^{(9)} = 0.049$ m. Elsewhere, the solution is coarsened aggressively down to refinement level (2) corresponding with $\Delta x^{(2)} = 6.25$ m. Using an adaptivity threshold of $\varepsilon = 10^{-3}$, the adaptive MWDG2 solver coarsens the solution more effectively than HFV1 in the locality of the hump. For the transcritical solution to converge to machine precision, MWDG2 required an adaptivity threshold $\varepsilon = 10^{-5}$ and, with this choice, MWDG2 behaves similarly to HFV1, using the maximum refinement level for the entire region of the hump (Fig. 15f). In summary, with a suitable choice of adaptivity threshold, all HFV1 and MWDG2 solvers converge to steady state solutions down to machine precision at about the same rate as the FV1 and DG2 solvers on a uniform mesh. They are also found to be as well-balanced as the underlying FV1 and DG2 uniform solvers. Adaptive HFV1 and MWDG2 solutions are coarsened down to refinement level (2), using elements that are 128 times coarser than the finest elements.

3.6. *Conservation of integral properties for an oscillatory flow in a parabolic bowl*

To analyse conservation properties over a long time evolution, the uniform and adaptive solvers are applied to simulate an oscillatory flow over topography. As shown in Lhomme et al. (2010), excessive numerical diffusion in shallow water models acts to dissipate energy and damp oscillatory flows. Assuming a frictionless topography, there are no sources or sinks of energy, which makes this test suitable to challenge the ability of a shallow water model to conserve mass and energy in the presence of moving wet-dry fronts. As in Delestre et al. (2013), an initially sloping water elevation is contained in a parabolic bowl defined on a one-dimensional domain in the interval [0 m, 4 m], given by:

$$z(x) = h_0\left(\frac{1}{a^2}(x-2)^2 - 1\right) \tag{62}$$

The exact solutions of the water height and the velocity are:

$$h(x,t) = \begin{cases} -h_0\left(\left(\frac{1}{a}(x-2) + \frac{B}{\sqrt{2gh_0}}\cos\left(\frac{\sqrt{2gh_0}}{a}t\right)\right)^2 - 1\right) & if \quad x_1(t) \le x \le x_2(t) \\ 0\ m & elsewhere \end{cases} \tag{63}$$

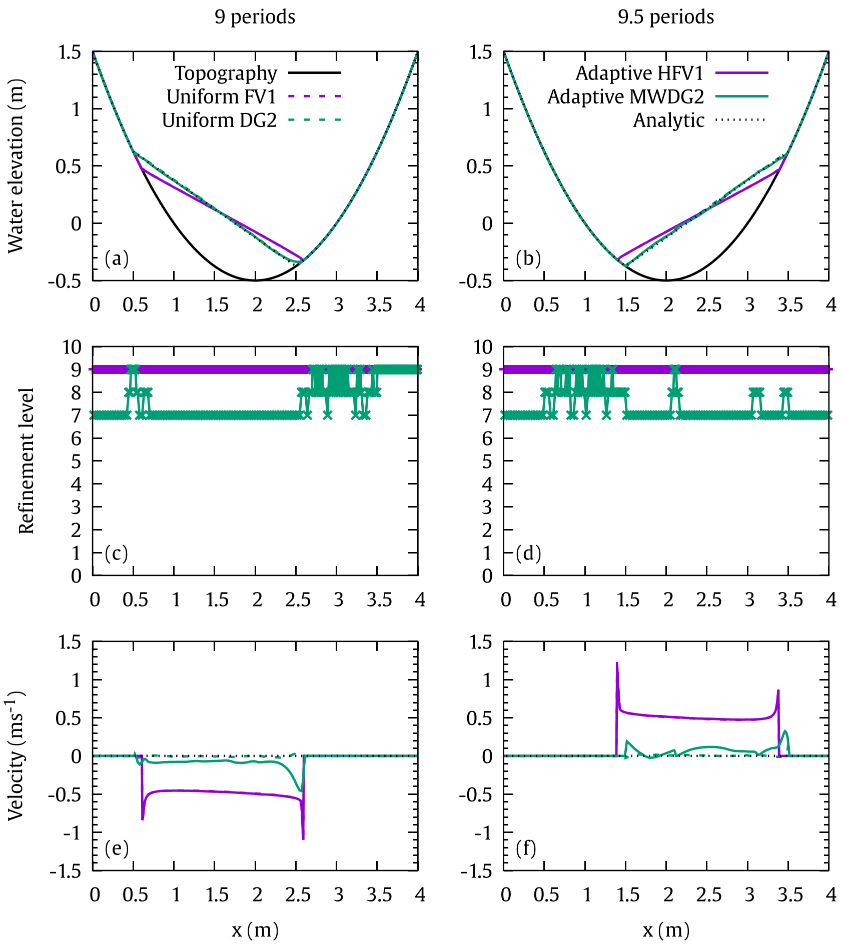


Fig. 16. Solution of (a, b) water elevation and (e, f) flow velocity for the simulation of the frictionless parabolic bowl. The analytical solution is compared to numerical solutions using the FV1 and DG2 solvers on a uniform mesh, and adaptive HFV1 and MWDG2 solvers are compared with are shown after 9 periods (left-hand panels) and 9.5 periods (right-hand panels). For the adaptive HFV1 and MWDG2 solvers, (c, d) shows the refinement levels for the corresponding solutions.

$$v(x,t) = \begin{cases} B\sin\left(\frac{\sqrt{2gh_0}}{a}t\right) & if \quad x_1(t) \le x \le x_2(t) \\ 0\ m/s & \quad elsewhere \end{cases} \tag{64}$$

where $x_1(t)$ and $x_2(t)$ are the locations of the wet-dry interfaces at time t, $h_0 = 0.5$ m, and $a = 1$ m (Delestre et al., 2013). The initial water height and flow velocity conditions can be obtained from Eqs. (63)–(64). Transmissive boundary conditions are imposed at both boundaries, but the parabolic bowl restricts the water to the domain interior. The uniform and adaptive solvers are applied considering the standard setting with the default error threshold (512 elements with the uniform solvers *vs.* $L = 9$, $M = 1$ and $\varepsilon = 10^{-3}$ with the adaptive solvers). Tests are integrated for 36.11 s, corresponding to 18 periods of oscillation. The period to complete one oscillatory cycle is $T = 2\pi a/\sqrt{2gh_0}$. The solution of the parabolic bowl behaves like a pendulum, with turning points occurring every half period, $0T$, $0.5T$, $1T$, $1.5T$, …, when the flow velocity is zero. At each period $0T$, $1T$, $2T$, …, the analytical water elevation is equal to the initial water elevation and at each intermediate period $0.5T$, $1.5T$, $2.5T$, …, the analytical water elevation is a mirror image of the initial water elevation.

3.6.1. Qualitative comparisons after 9 periods

Numerical solutions using the FV1, DG2, adaptive HFV1 and MWDG2 solvers are compared with the analytical solution in Fig. 16. The DG2 and MWDG2 solutions of water elevation closely agree with the analytical solution after 9 periods (Fig. 16a) and 9.5 periods (Fig. 16b). In contrast, oscillations are damped by the first-order accurate FV1 and HFV1 solvers, and the water elevation after 9 periods no longer reaches the maximum initial water elevation. For the velocity predictions, the DG2 solver obtains calculations that are consistently close to the analytical solution of $v = 0$ m s^{-1} after 9 periods (Fig. 16e) and 9.5 periods (Fig. 16f). The adaptive MWDG2 solver also achieves small flow velocities except around the wet-dry fronts. The FV1 and HFV1 solutions have flow velocity errors of about 0.4 m s^{-1} with larger error magnitudes in the locality of the wet-dry fronts. The refinement levels predicted by the adaptive HFV1 and MWDG2 solvers are presented corresponding to the solution after 9 periods (Fig. 16c) and 9.5 periods (Fig. 16d). The HFV1 solver uses the maximum refinement level (9) throughout the domain, as expected given the curved shape of the parabolic topography (recall the analysis in Section 3.4). The adaptive MWDG2 solver uses the maximum refinement level just at the wet-dry fronts, and *temporarily* in some dry regions where small-scale noise occurs in the solutions. Such noise can be reduced by slightly increasing the error threshold. Apart from these isolated regions, MWDG2 uses only refinement level (7), resulting in almost four times fewer elements than the uniform solvers with 512 elements.

3.6.2. Mass conservation and energy conservation

The frictionless parabolic bowl is a closed system with no sources or sinks of mass or energy. As the water oscillates within the bowl, there is an exchange between kinetic and potential energy, but the total energy is conserved. The time evolution of total mass and total energy is measured in order to assess the conservation properties of the numerical solvers. Only the average coefficients are used in both mass and energy calculations, which were evaluated for the assembled solution on g_i^A. That is, the total mass produced by the adaptive solvers on g_i^A is

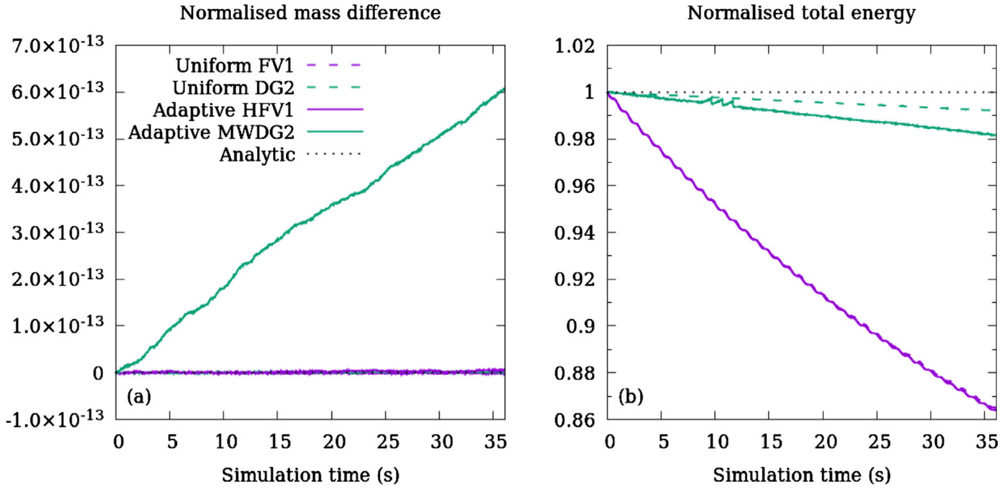


Fig. 17. Evolution of (a) change in mass and (b) normalised total energy for the simulation of the frictionless parabolic bowl. The 36.11 second-long simulation corresponds to 18 periods of oscillation.

calculated as:

$$M = \sum_{e \in g_i^A} \left(h_e^{0,(n)} \Delta x_e^{(n)} \right) \tag{65}$$

From Eq. (65), the mass difference ΔM is evaluated as $\Delta M(t) = M(t) - M_0$, with $M_0 = M(0)$ being the initial mass at $t = 0$ s. The mass difference is normalised relative to the initial mass as:

$$\Delta \hat{M}(t) = \Delta M(t)/M_0 \tag{66}$$

The total energy is calculated as the sum of kinetic and potential energy (Vreugdenhil, 1994):

$$E = \sum_{e \in g_i^A} \left\{ \left[\frac{1}{2} h_e^{0,(n)} \left(v_e^{0,(n)} \right)^2 + \left(h_e^{0,(n)} + z_e^{0,(n)} \right)^2 - \left(z_e^{0,(n)} \right)^2 \right] \Delta x_e^{(n)} \right\} \tag{67}$$

which is normalised relative to the initial total energy $E_0 = E(0)$ such that:

$$\Delta(t) = E(t)/E_0 \tag{68}$$

For the uniform solvers, Eqs. (65)–(68) are applied for their assembled solution on g_i^L instead of g_i^A.

The time histories of the normalised mass difference are illustrated in Fig. 17a for the FV1 and DG2 solvers on a uniform mesh, and the adaptive HFV1 and MWDG2 solvers. The FV1, HFV1 and DG2 solvers conserve mass to machine precision (Fig. 17a). The HFV1 solver retains refinement level (9) yielding simulations on an equivalent grid as the FV1 solver, but at a higher cost: here, HFV1 does not zero any detail coefficient and so gets unnecessarily overloaded with overhead cost due to Haar-wavelet adaptivity (recall the analysis in Section 3.1.5). Unsurprisingly, HFV1 delivers the same level of conservativeness as the uniform FV1 solver for both mass and energy quantities (Fig. 17). The MWDG2 solver constantly altered refinement levels between (7) and (9), resulting in a loss of information due to zeroing of detail coefficients. Given also that the multi-wavelet adaptivity of the MWDG2 solver must filter both average and slope coefficients – via constant rounding of the irrational numbers involved in the filters – these effects result in a very small, linear growth in mass (Fig. 17a). Nonetheless, MWDG2 mass conservation errors are still close to machine precision, even after 18 periods of oscillation. The normalised total energy is also measured at each time-step for the FV1, DG2, HFV1 and MWDG2 solvers (Fig. 17b). As expected for a first-order solver, FV1 and HFV1 dissipate energy quite rapidly, losing about 13% of the initial energy after 18 periods of oscillation. In contrast, the DG2 solver on a uniform mesh achieves excellent energy conservation, losing less than 1% of the initial energy after 18 periods. Despite the adaptive MWDG2 solver coarsening the solution to refinement level (7), it is only slightly more dissipative than the DG2 solver, with MWDG2 losing less than 2% of the initial energy.

For such a dynamic oscillatory flow over a curved topography with wet-dry fronts, HFV1 with $\varepsilon = 10^{-3}$, delivers the same predictive accuracy as the uniform FV1 solver on the finest grid, but is expected to be more costly to run (Section 3.1.5). Employing HFV1 with bigger ε gives an under-performance relative to the present accuracy of FV1 and so may not be a feasible option for this type of simulation. The DG2 solver on the finest uniform grid shows excellent conservation properties for both mass and energy quantities. The adaptive MWDG2 is likely to be more efficient than HFV1 for this type of simulation, and preserves the conservation properties of the DG2 solver with inconsequential effects.

3.7. Numerical simulation of a laboratory dam-break over a trapezoidal hump

Ozmen-Cagatay and Kocaman (2011) conducted a laboratory flume experiment of a dam-break flow over a trapezoidal hump. This test involves a wet-dry front advancing over a frictional topography, wave overtopping on a building-like hump and a topographically-reflected shock wave. In particular, it is an ideal benchmark to validate the practicality of the HFV1 and MWDG2 solvers in modelling realistic aspects of shallow water flows in a multi-scale setting and in relation to the increase in maximum refinement level. The physical experiment (Ozmen-Cagatay and Kocaman, 2011) was conducted in an 8.9 m long acrylic glass flume, with the configuration illustrated in Fig. 18. The topography and initial water elevation profile are the same for the numerical tests presented here, with an initial zero discharge. A reflective boundary condition is imposed at the upstream boundary and a transmissive boundary condition is imposed downstream. The Manning coefficient for acrylic glass is 0.01 $\mathrm{m}^{1/3}\,\mathrm{s}^{-1}$. The water in the flume was photographed at regular time intervals and the water elevation profile was measured to an accuracy of about ±1 mm. Experimental measurements of water elevation are compared with numerical solutions at time $T = 11.9$, $T = 23.05$ and $T = 41.84$, where T is a nondimensionalised time $T = \sqrt{gh_0}\, t$ with $h_0 = 0.25$ m denoting the initial height behind the gate located at $x_0 = 4.65$ m.

Numerical solutions are obtained using the FV1 and DG2 solvers on a uniform mesh with 2^L elements, and adaptive HFV1 and MWDG2 solvers on a baseline grid with a single mother element, a maximum refinement level L and with the default error threshold ($\varepsilon = 10^{-3}$). Tests are performed with $L = 7$, 9 and 11 corresponding to a finest grid spacing of $\Delta x^{(7)} = 0.070$ m, $\Delta x^{(9)} = 0.017$ m and $\Delta x^{(11)} = 0.0043$ m, or respectively to 128, 512 and 2048 elements for the finest uniform grid.

As shown in Fig. 19a–19c, at $L = 9$, the adaptive and uniform solutions closely agree with the experimental observations at $T = 11.9$, $T = 23.05$ and $T = 41.84$, since the topography and fine-scale flows are well-resolved at $\Delta x^{(9)} = 0.017$ m. While a similar behaviour for the

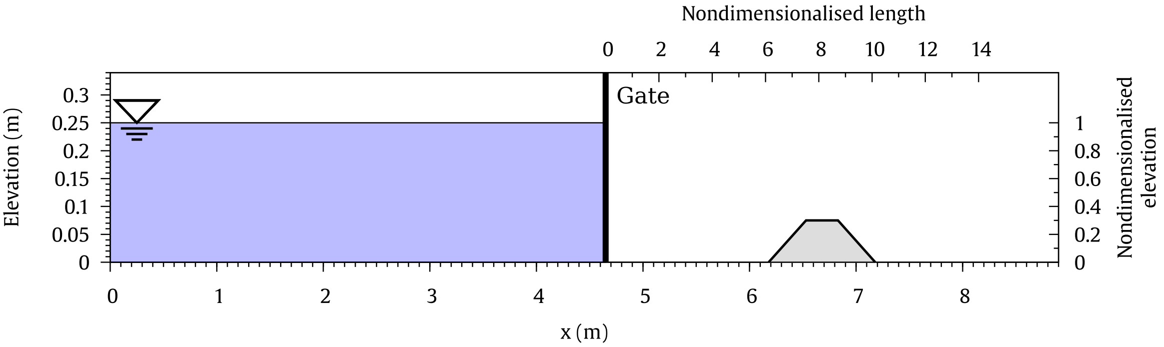


Fig. 18. Initial configuration of the dam-break over a trapezoidal hump following Ozmen-Cagatay and Kocaman (2011). Nondimensionalised scales are used in subsequent figures. Illustrated aspect ratio is 5:1.

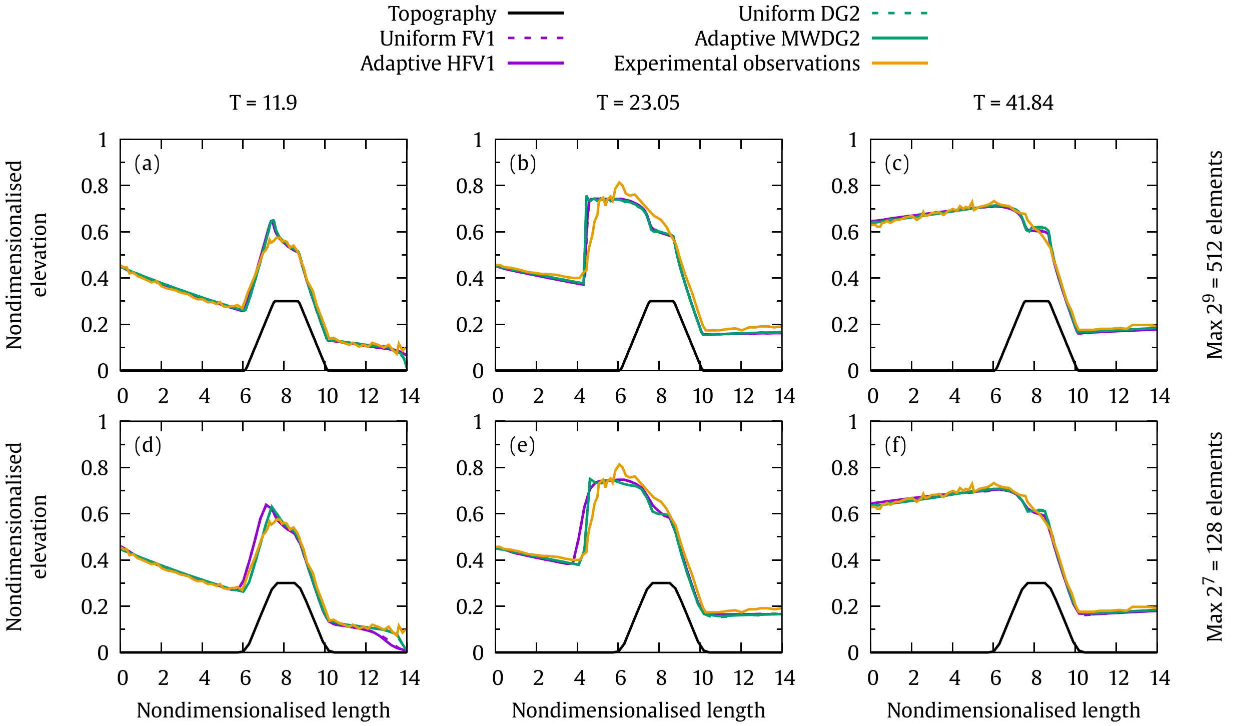


Fig. 19. Snapshots of water elevation for the dam-break over a trapezoidal hump with friction at nondimensionalised times (a, d) $T = 11.9$ (b, e) $T = 23.05$ and (c, f) $T = 41.84$, where T is a nondimensionalised measure of time given by equation. Numerical solutions are obtained using FV1 and DG2 solvers on a uniform mesh with 2^L elements, and adaptive HFV1 and MWDG2 solvers on a baseline mesh with a single mother element and a maximum refinement level L, with (a, b, c) $L = 9$, and (d, e, f) $L = 7$. The nondimensionalised elevation is h/h_o and the nondimensionalised length is $(x - x_0)/h_0$, with the plotted origin being the gate position $x_0 = 4.65$ m.

adaptive and uniform solutions is expected at $L = 11$ as $\Delta x^{(11)} < \Delta x^{(9)}$, with $L = 7$, the topography and fine-scale flow cannot be sufficiently resolved by the FV1 and HFV1 solvers using a piecewise-constant basis (Fig. 19d and 19e). At $T = 11.9$, FV1 and HFV1 simulations produce insufficient overtopping on the lee side of the obstacle (Fig. 19d) and, at $T = 11.9$ and $T = 23.05$ (Fig. 19e), the reflected wave is positioned far upstream compared to the experimental observations. Numerical diffusion is particularly evident in the FV1 and HFV1 solutions at $T = 23.05$ which is not present in the same solutions on the finer mesh using $L = 9$. In contrast, since the DG2 and MWDG2 solvers use a piecewise-linear basis, the fine-scale features are still well-resolved even at $L = 7$ with $\Delta x^{(7)} = 0.070$ m. Using the same test, Kesserwani and Wang (2014) achieved accurate DG2 solutions using a significantly coarser mesh of $\Delta x = 0.22$ m, and obtained second-order MUSCL-FV solutions with errors similar to those obtained with the FV1 and HFV1 solvers. In terms of refinement level predictions, both adaptive HFV1 and MWDG2 solvers are observed to fully refine around the trapezoidal obstacle given the sloping character of its sides and the dynamic nature of the flow. To realistically analyse efficiency benefits of the adaptive solvers, their cumulative CPU time costs are further recorded for completing 30 s numerical simulations (corresponding to $T = 188$ s).

The elapsed CPU time is measured at every time-step, and these time series are illustrated for $L = 7$ (Fig. 20 – upper part), $L = 9$ (Fig. 20 – middle part) and $L = 11$ (Fig. 20 – lower part). At $L = 7$ with $\Delta x^{(7)} = 0.070$ m, the FV1 and adaptive HFV1 solvers complete the simulation the fastest (Fig. 20 – upper part), but produce somewhat inaccurate solutions since the grid is relatively coarse (Fig. 19 – lower parts). Accurate solutions are achieved using the DG2 and MWDG2 solvers, but the adaptive MWDG2 solver completes the simulation in about half the time of DG2 on a grid with $2^7 = 128$ uniform elements. At $L = 9$ with $\Delta x^{(9)} = 0.017$ m, the HFV1 and MWDG2 solvers complete the simulation around the same time (Fig. 20 – middle part). The DG2 solver is about five times more computationally expensive and completes the simulation after 10.3 s of CPU time. At this grid resolution, the FV1 solver remains the most

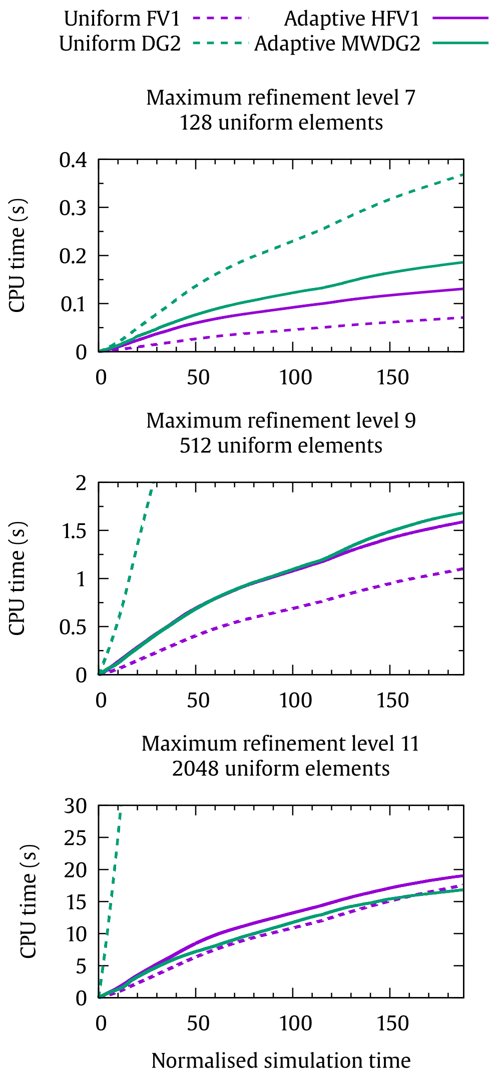


Fig. 20. Cumulative CPU times to compete a 30 s numerical simulations (corresponding to T = 188 s) for the uniform FV1 and DG2 solvers on a uniform mesh with 2^L elements, and the adaptive HFV1 and MWDG2 solvers on a baseline mesh with a single mother element and a maximum refinement level L: upper part $L = 7$, medium part $L = 9$ and upper part $L = 11$.

computationally efficient choice, and produces a solution with similar accuracy to the other solvers (Fig. 19 – upper parts). At $L = 11$ with $\Delta x^{(11)} = 0.0043$ m, no improvement in solution accuracy is expected since the flow in the 8.9 m-long flume is already well-resolved with coarser meshes. However, at $L = 11$, the adaptive MWDG2 solver is, surprisingly, the first to finish the simulation, followed by the FV1 and adaptive HFV1 solvers (Fig. 20 – lower part) and, compared to the DG2 solver on a uniform mesh, the MWDG2 solver is 27 times faster. Clearly, with increased maximum refinement level, MWDG2 tends to become faster than the uniform FV1 solver on the finest grid and, ultimately than the HFV1 solver. In terms of resolution accuracy, taking $L = 11$ is unnecessary for this test, as $L = 9$ provide sufficient resolution, but does still pay off with an increase in MWDG2 solver's efficiency. Given also that MWDG2 provides superior accuracy with $L = 7$ (i.e. up to a resolution of 0.070 m), the MWDG2 solver could be even more beneficial, in favour of accuracy, when the finest resolution involved in the adaptive grid is roughly ≥ 0.1 m. Hence, the MWDG2 solver seems to be a promising alternative for simulations over a large domain (10 km and more in horizontal length scale) allowing multi-scale features that are as small as 0.1 m, nonetheless at a lower runtime cost than the uniform FV1 solver on the finest grid available and at nearly the same accuracy as the expensive uniform DG2 solver on the finest grid.

4. Summary and conclusions

A scaled second-order Discontinuous Galerkin (DG2) solver of the Shallow Water Equations (SWE) was presented (Section 2.1), with guiding principles on how it extends to incorporate multiresolution analysis (Section 2.2) based on multiwavelets (MW) to form the so-called adaptive MWDG2 solver (Section 2.3). Our aim has been to explain this framework in a way that is understandable by water engineers and modellers, and to unravel its relevant benefits for improving the accuracy, efficiency and autonomy of Godunov-type hydrodynamic models. In the adaptive MWDG2 solver, flow and topography data at various resolution levels are compressed in a single dataset of details, or wavelet coefficients (Section 2.3.1). From these details, a multiresolution DG2 solution can be created and assembled on a non-uniform grid by retaining the significant details and adding them to the coarsest solution discretisation. Significant details were identified by comparing their magnitude to an error threshold ε (Section 2.3.2). The scaled DG2 solver can directly be applied to evolve the multiresolution DG2 solution on an adaptive non-uniform grid (Section 2.3.3). Zero-valued detail coefficients were imposed to complete the dataset of details as time evolved (Section 2.3.4). A first-order version was produced based on the Haar wavelet within the Finite Volume (HFV1) method (Section 2.4). The behaviour of the adaptive HFV1 and MWDG2 solvers was studied systematically and compared against the standard first-order Finite Volume (FV1) and second-order Discontinuous Galerkin (DG2) solvers on a uniform grid. Seven tests were used to diagnostically explore the performance of the adaptive (multi)wavelet-based solvers, which covered all the elementary aspects relevant to accurate, efficient and robust hydraulic modelling (Section 3). Adaptive solver simulations started from a coarsest grid discretisation with M mother elements, with each allowing a maximum of 2^L sub-elements (a maximum refinement level L yielding $M \leq$ number of sub-elements $\leq M\,2^L$). The uniform solver simulations considered the grid at the finest resolution available (with $M\,2^L$ elements). The numerical results consistently reinforced the conclusion that the (multi)wavelet-based solvers offer many attractive properties including the ability to: (i) automate the formulation of an initial multiresolution mesh, (ii) use very few, or a single, mother element(s) as a baseline grid, (iii) allow large gaps across resolution levels, (iv) preserve robustness, accuracy and conservation properties of the standard uniform solvers, and (v) adapt modelling resolution and data simply with reference to the user-prescribed error threshold ε.

More strikingly, findings from this study newly identify a range for the error threshold ε where the adaptive MWDG2 solver can deliver simulations that are not only as accurate as the uniform DG2 simulations but also faster than the simulations delivered by both the adaptive HFV1 solver and the uniform FV1 solver. Mainly, MWDG2 outperformed HFV1 as a result of the sloping nature of its local piecewise-linear solutions, which allowed much more aggressive coarsening at the zones in the flow solution and topographic data involving different levels of smoothness. At these zones, the adaptive HFV1 solver consistently over-refined up to becoming even more expensive than the uniform FV1 solver since HFV1 was dominated by a wavelet-adaptivity overhead. In contrast, the adaptive MWDG2 solver more sensibly predicted coarser solutions and did not access the finest resolution level unless necessary around very steep solution gradients. The efficiency of the adaptive MWDG2 solver was found to increase by increasing the maximum refinement level L, though its predictive accuracy remained visually close to the first-order solver predictions at a very fine resolution, namely around $\Delta x^{(L)} \leq 0.07$ m. Our results therefore offer new evidence that an MWDG2 modelling approach has the potential to increase the accuracy, runtime efficiency and spatial coverage for hydraulic modelling applications for which the maximum refinement level is associated with an urban resolution grid (approx. around 0.1 m in horizontal length-scale). A robust two dimensional (2D) extension of the MWDG2 approach on quadrilateral elements is under development and testing to enable a more realistic assessment of the true potential of (multi)wavelet-based approaches for 2D hydraulic modelling applications.

Acknowledgements

The authors wish to thank Dr Matthias Rauter (Norwegian Geotechnical Institute) for providing the semi-analytical solution of the 1D frictional dam break test case, and Mr. Xitong Sun for his insightful discussions on the derivation of the high-pass filters. We also thank Prof. Onno Bokhove (University of Leeds) for their helpful advice during the testing and validation of the adaptive wavelet-based numerical solver. This work is part of the SEAMLESS-WAVE project (SoftwarE infrAstructure forMulti-purpose fLood modElling at variouS scaleS based on WAVElets), which is funded by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/R007349/1. For information about the SEAMLESS-WAVE project visit https://www.seamlesswave.com.

Appendix 1. Instructions for running the FV1, DG2, HFV1 and MWDG2 solvers

Compilation

The seamless-wave numerical solvers are implemented in Fortran 2003 and can be compiled using a recent version of GFortran and CMake. Other fortran compilers have not been tested. To compile the code from the root directory of the unzipped Zenodo download:

```
mkdir build && cd build
cmake ..
make -j
```

Running the numerical solvers

The FV1, DG2, HFV1 and MWDG2 solvers are all implemented in a single executable, `run_simulation`. To display usage information about required and optional command line switches:

```
./run_simulation --help
```

All the test cases that appear in this article are preconfigured. To run one of the test cases:

`./run_simulation <testCase> <maxRefinementLevel> --solver <solver> --writer <writer>` where *<testCase>* is one of

`dambreakwet`	section 3.1
`dambreakdry` (frictionless), `dambreakmanning` (frictional)	section 3.2
`dambreakupslope`, `dambreakdownslope`	section 3.3
`lakeatrest`	section 3.4
`steadysubcritical`, `steadysupercritical`, `steadytranscriticalshock`	section 3.5
`parabolicbowlswashes`	section 3.6
`dambreakonehump`	section 3.7

To solve on a uniform mesh, use *<maxRefinementLevel>* to create a mesh with 2^L elements, and choose *<solver>* to be either `fv1` or `dg2`. To calculate an adaptive solution, include the switch `--epsilon` *<value>* with *<value>* being a double precision number between 0 and 1. When `--epsilon` is specified, adaptive refinement is allowed up to the given `<maxRefinementLevel>`. *<solver>* is still either `fv1` or `dg2` for an adaptive solution.

The solver will write space-delimited plain text data depending on the choice of *<writer>*. The following writers output data corresponding to the end of the simulation:

`cellCentreSolution`	topography, water depth, discharge and refinement level data
`piecewiseSolution`	as `cellCentreSolution`, but data is at the interface limits
`l2error`	calculate the ℓ^2 error between numerical and analytic solutions

The following writers output data at every timestep:

`cpu`	elapsed CPU time
`timestep`	size of Δt
`elementCount`	total element count
`convergence`	ℓ^2 convergence in water depth
`energy`	domain integrals of mass and energy
`wetDryFront`	the position of the wet-dry front
`sample`	sample data at a specified `--sample-position`

Additional, optional switches are documented by using `./run_simulation --help`.

References

Alpert, B., 1993. A class of bases in L^2 for the sparse representation of integral operators. SIAM J. Math. Anal. 24 (1), 246–262.
Alpert, B., et al., 2002. Adaptive solution of partial differential equations in multiwavelet bases. J. Comput. Phys. 182 (1), 149–190.
An, H., Yu, S., 2014. An accurate multidimensional limiter on quadtree grids for shallow water flow simulation. J. Hydraul. Res. 52 (4), 565–574.
Archibald, R., Fann, G., Shelton, W., 2011. Adaptive discontinuous Galerkin methods in multiwavelets bases. Appl. Numer. Math. 61 (7), 879–890.
Bokhove, O., 2005. Flooding and drying in discontinuous Galerkin finite element method for shallow water flows. J. Sci. Comput. 22-23 (1), 47–82.
Caleffi, V., Valiani, A., 2017. Well balancing of the SWE schemes for moving-water steady flows. J. Comput. Phys. 342, 85–116.
Caviedes-Voullième, D., Kesserwani, G., 2015. Benchmarking a multiresolution discontinuous Galerkin shallow water model: Implications for computational hydraulics. Adv. Water Res. 86 (Part A), 14–31.
Cockburn, B., Shu, C.-W., 2001. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. J. Sci. Comput. 16 (3), 173–261.
Cohen, A., et al., 2003. Fully adaptive multiresolution finite volume schemes for conservation laws. Math. Comput. 72 (241), 183–225.
Delestre, O., et al., 2013. SWASHES: a compilation of shallow water analytic solutions for hydraulic and environmental studies. Int. J. Numer. Methods Fluids 72 (3), 269–300.
Delis, A.I., Nikolos, I.K., Kazolea, M., 2011. Performance and comparison of cell-centered and node-centered unstructured finite volume discretizations for shallow water free surface flows. Arch. Comput. Meth. Eng. 18 (1), 57–118.
Donat, R., et al., 2014. Well-balanced adaptive mesh refinement for shallow water flows. J. Comput. Phys. 257 (0), 937–953 Part A.
Gerhard, N., et al., 2015. A high-order discontinuous Galerkin discretization with multiwavelet-based grid adaptation for compressible flows. J. Sci. Comput. 62 (1), 25–52.
Gerhard, N., et al., 2015. Multiwavelet-based grid adaptation with discontinuous Galerkin schemes for shallow water equations. J. Comput. Phys. 301, 265–288.
Gerhard, N., Müller, S., 2014. Adaptive multiresolution discontinuous Galerkin schemes for conservation laws: multi-dimensional case. Comput. Appl. Math. 35 (2), 1–29.
Haleem, D.A., Kesserwani, G., Caviedes-Voullième, D., 2015. Haar wavelet-based adaptive finite volume shallow water solver. J. Hydroinf. 17 (6), 857–873.
Harten, A., 1995. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. Commun. Pure Appl. Math. 48 (12), 1305–1342.
Hou, J., et al., 2013. A stable 2D unstructured shallow flow model for simulations of wetting and drying over rough terrains. Comput. Fluids 82 (0), 132–147.
Hovhannisyan, N., Müller, S., Schäfer, R., 2014. Adaptive multiresolution discontinuous Galerkin schemes for conservation laws. Math. Comput. 83 (285), 113–151.
Keinert, F., 2004. Wavelets and Multiwavelets. Chapman and Hall/CRC, Boca Raton, Florida.
Kesserwani, G., et al., 2010. Well-balancing issues related to the RKDG2 scheme for the shallow water equations. Int. J. Numer. Methods Fluids 62 (4), 428–448.
Kesserwani, G., 2013. Topography discretization techniques for Godunov-type shallow water numerical models: a comparative study. J. Hydraul. Res. 51 (4), 351–367.
Kesserwani, G., et al., 2015. Multiwavelet discontinuous Galerkin h-adaptive shallow water model. Comput. Meth. Appl. Mech. Eng. 294, 56–71.
Kesserwani, G., Ayog, J.L., Bau, D., 2018. Discontinuous Galerkin formulation for 2D hydrodynamic modelling: Trade-offs between theoretical complexity and practical convenience. Comput. Meth. Appl. Mech. Eng. 342, 710–741.
Kesserwani, G., Liang, Q., 2012. Influence of total-variation-diminishing slope limiting on local discontinuous Galerkin solutions of the shallow water equations. J. Hydraul. Eng. 138 (2), 216–222.
Kesserwani, G., Liang, Q., 2012. Locally limited and fully conserved RKDG2 shallow water solutions with wetting and drying. J. Sci. Comput. 50 (1), 120–144.
Kesserwani, G., Liang, Q., 2012. Dynamically adaptive grid based discontinuous Galerkin shallow water model. Adv. Water Res. 37, 23–39.
Kesserwani, G., Liang, Q., 2015. RKDG2 shallow-water solver on non-uniform grids with local time steps: application to 1D and 2D hydrodynamics. Appl. Math. Model. 39 (3–4), 1317–1340.
Kesserwani, G., Wang, Y., 2014. Discontinuous Galerkin flood model formulation: luxury or necessity? Water Resour. Res. 50 (8), 6522–6541.

Krivodonova, L., et al., 2004. Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. Appl. Numer. Math. 48 (3-4), 323–338.

Lacasta, A., et al., 2013. Preprocess static subdomain decomposition in practical cases of 2D unsteady hydraulic simulation. Comput. Fluids 80 (0), 225–232.

Lhomme, J., et al., 2010. Testing a new two-dimensional flood modelling system: analytical tests and application to a flood event. J. Flood Risk Manage. 3 (1), 33–51.

Li, S., 2010. Comparison of refinement criteria for structured adaptive mesh refinement. J. Comput. Appl. Math. 233 (12), 3139–3147.

Liang, Q., Borthwick, A.G.L., 2009. Adaptive quadtree simulation of shallow flows with wet-dry fronts over complex topography. Comput. Fluids 38 (2), 221–234.

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

Medeiros, S.C., Hagen, S.C., 2013. Review of wetting and drying algorithms for numerical tidal flow models. Int. J. Numer. Methods Fluids 71 (4), 473–487.

Minatti, L., De Cicco, P.N., Solari, L., 2016. Second order discontinuous Galerkin scheme for compound natural channels with movable bed. Applications for the computation of rating curves.. Adv. Water Res. 93, 89–104.

Minbashian, H., Adibi, H., Dehghan, M., 2017. An adaptive wavelet space-time SUPG method for hyperbolic conservation laws. Numer. Methods Part. Differ. Equ. 33 (6), 2062–2089.

Müller, S., 2002. Adaptive Multiscale Schemes for Conservation Laws. Springer.

Ozmen-Cagatay, H., Kocaman, S., 2011. Dam-break flow in the presence of obstacle: experiment and CFD simulation. Eng. Appl. Comput. Fluid Mech. 5 (4), 541–552.

Sanders, B.F., Schubert, J.E., Detwiler, R.L., 2010. ParBreZo: a parallel, unstructured grid, Godunov-type, shallow-water code for high-resolution flood inundation modeling at the regional scale. Adv. Water Res. 33 (12), 1456–1467.

Sharifian, M.K., Kesserwani, G., Hassanzadeh, Y., 2018. A discontinuous Galerkin approach for conservative modeling of fully nonlinear and weakly dispersive wave transformations. Ocean Modell. 125, 61–79.

Shaw, J. and G. Kesserwani, One-dimensional hydrodynamic solution data of wavelet-based adaptive finite volume and discontinuous Galerkin shallow water solvers [Data set]. 2018, Zenodo, doi:10.5281/zenodo.1491311.

Shaw, J., G. Kesserwani, and M.K. Sharifian, One-dimensional Fortran 2003 finite volume and discontinuous Galerkin hydrodynamic models with wavelet-based adaptivity. 2018, Zenodo, doi:10.5281/zenodo.1745471.

Teng, J., et al., 2017. Flood inundation modelling: a review of methods, recent advances and uncertainty analysis. Environ. Model. Softw. 90, 201–216.

Toro, E.F., 2001. Shock-Capturing Methods for Free-Surface Shallow Flows, ed John Wiley & Sons, Ltd.

Toro, E.F., Garcia-Navarro, P., 2007. Godunov-type methods for free-surface shallow flows: a review. J. Hydraul. Res. 45 (6), 736–751.

Vreugdenhil, C.B., 1994. Numerical Methods for Shallow-Water Flow. Water Science and Technology Library. Springer, Netherlands, p. 262.

Vuik, M.J., Ryan, J.K., 2014. Multiwavelet troubled-cell indicator for discontinuity detection of discontinuous Galerkin schemes. J. Comput. Phys. 270, 138–160.

Wang, Z., et al., 2016. Sparse grid discontinuous Galerkin methods for high-dimensional elliptic equations. J. Comput. Phys. 314, 244–263.

Xing, Y., Zhang, X., Shu, C.-W., 2010. Positivity-preserving high order well-balanced discontinuous Galerkin methods for the shallow water equations. Adv. Water Res. 33 (12), 1476–1493.

Zhou, F., et al., 2013. An adaptive moving finite volume scheme for modeling flood inundation over dry and complex topography. Water Resour. Res. 49 (4), 1914–1928.