

(Multi)wavelet-based Godunov-type simulators of flood inundation: Static versus dynamic adaptivity

Georges Kesserwani*, Mohammad Kazem Sharifian

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



ARTICLE INFO

Keywords:

Flood modelling
FV1 and DG2 solvers
Multiresolution analysis
GPU parallelisation
Static vs. dynamic adaptivity
Practical recommendations

ABSTRACT

Real-world flood simulators often use first-order finite volume (FV1) solvers of the shallow water equations with efficiency enhancements exploiting parallelisation on Graphical Processing Units (GPUs) and the use of static adaptivity on fixed grids. A second-order discontinuous Galerkin (DG2) solver greatly increases the accuracy in the predictions on uniform grids, where it is comparatively costly to run, but its practical utility as an alternative for flood simulations using static adaptivity is not yet assessed. This is also the case for the dynamic adaptivity using the multiresolution analysis (MRA) of the Haar wavelet (HW) scaling FV1 piecewise-constant solutions (HWFV1) and of the smoother Multiwavelets (MWs) that scales DG2 piecewise-planar solutions (MWDG2) to adapt the resolution of their grids over time. Therefore, dynamic MWDG2 and HWFV1 adaptivity is newly explored for practical real-world simulations, to find out when they yield better predictions than static DG2 and FV1 adaptivity. A new GPU implementation is proposed to include dynamic MWDG2 adaptivity to also assess how far GPU parallelisation renders its runtime practically feasible. Dynamic and static adaptivity are assessed for three tests involving slow, gradual to rapid flood flows with analyses of their predictive accuracy and computational costs with reference to uniform grid DG2 simulations at the finest resolution of the digital elevation model (DEM). Findings suggest favouring static FV1 adaptivity for long-duration simulations of slowly to gradually propagating floods and dynamic MWDG2 adaptivity to simulate events driven by rapidly propagating flows. On the GPU, dynamic MWDG2 adaptivity is faster than uniform DG2, leading to a higher speedup ratio with higher reduction in the elements on its initial, fixed grid.

1. Introduction

Flood modelling is essential to understand flood flow dynamics and define mitigation strategies to support decision-making in flood risk management (Spiekermann et al., 2015). With the increasing availability of accurate raster-based terrain datasets, two-dimensional (2D) flood models have gained popularity for complex, field-scale flood inundation modelling (Hunter et al., 2008; Teng et al., 2017; Nkuronwo et al., 2020). Raster-based flood models use a numerical solver on a uniform resolution grid, defined by a given raster-formatted digital elevation model (DEM) file. Flood models used to support real-world flood simulations have often been based on solving the full shallow water equations in a first-order finite volume (FV1) Godunov-type numerical solver, on exploiting parallelisation on Graphical Processing Units (GPUs) (Xia et al., 2019; Echeverribar et al., 2019; Zhao and Liang, 2022) and/or deploying static adaptivity of the grid resolution (BMT, 2018; Hou et al., 2018; Vacondio et al., 2017) in order to reduce

runtime costs (Bellos et al., 2020; Saksena et al., 2020; Guo et al., 2020). On a uniform grid, raster-based flood models based on the second-order discontinuous Galerkin (DG2) solver can be built by increasing the local formulation's complexity of the FV1 solver (Shaw et al., 2021), leading to greater quality predictions than finite volume industrial models (Ayog et al., 2021). The locality of the DG2 solver formulation should also make it a better candidate to increase the quality of the flood flow predictions within standard static adaptivity, but this is yet to be investigated for real-field simulations including the efficient generation of the resolution scales for the fixed grid from a thorough analysis of the topographical features within the DEM.

Alternatively, modern (multi)wavelet-based solvers with dynamic adaptivity were formulated based upon the multiresolution analysis (MRA) of the Haar wavelet (HW) or the smoother Multiwavelets (MWs) to adapt the grid resolution in time, driven by scaled piecewise-constant FV1 numerical representations of the flow solutions and the DEM or smoother piecewise-planar DG2 representations, respectively

* Corresponding author at: Department of Civil and Structural Engineering, University of Sheffield, Sheffield S1 3JD, UK.

E-mail address: G.kesserwani@sheffield.ac.uk (G. Kesserwani).

(Kesserwani et al., 2019; Kesserwani and Sharifian, 2020). These combinations will be termed as the HWFV1 and MWDG2, respectively. In essence, dynamic MWDG2 (respectively HWFV1) adaptivity involves the MRA process to generate the grid every time step and while using high- and low-pass filter formulae to rigorously scale and reconstruct the numerical representations across disparate resolutions. The (multi)wavelet-based dynamic adaptivity of the MRA process is practical as it needs only one user-specified parameter, an error threshold ϵ , to select the scales of grid resolution. Moreover, it uses the rigorous filter formulae to reconstruct intrinsically well-balanced solution limits to integrate the fluxes across heterogeneously-sized, adjacent elements as robust as they are integrated in the uniform grid solver counterparts (Haleem et al., 2015; Gerhard et al., 2015; Kesserwani et al., 2015; Caviedes-Voullième and Kesserwani, 2015; Caviedes-Voullième et al., 2020).

The serial implementation of dynamic MWDG2 (respectively HWFV1) adaptivity for hydraulic modelling on the central processing unit (CPU) was already devised and extensively validated for a series of synthetic and laboratory-scale test cases mostly dominated by wet areas (Kesserwani et al., 2019; Kesserwani and Sharifian, 2020). The findings identify an ϵ value in the magnitude of 10^{-3} with dynamic MWDG2 (respectively HWFV1) adaptivity to preserve the predictive accuracy of the DG2 solver (respectively FV1) run on the finest uniform grid at a competitively affordable runtime, namely less or close to that of the FV1 solver run on the finest uniform grid (Kesserwani and Sharifian, 2020). In serial, the efficiency gain of the dynamic MWDG2 adaptivity arises from its MRA of MWs operating on piecewise-planar representations, which causes more aggressive, yet sensible, grid coarsening than with the MRA of HW that operates on piecewise-planar representations within dynamic HWFV1 adaptivity. However, dynamic MWDG2 (respectively HWFV1) adaptivity must apply the MRA process every time step, and this entails severe overhead costs in a serial implementation; in particular, for real-world simulations where the flooding inundates large dry areas. For such simulations, dynamic MWDG2 and HWFV1 adaptivity has yet to be evaluated when reworked to run in parallel on the GPU, to assess the extent of their runtime affordability for practical flood modelling applications.

On the other hand, the findings make a clear case for using the MRA of MWs, instead of that of HW, to efficiently generate a fixed grid with sensibly selected resolution scales (Kesserwani et al., 2019; Kesserwani and Sharifian, 2020). In this context, the MRA process is applied only once to analyse the features of the DEM. With such a static adaptivity, the rigorous filter formulae of the (multi)wavelet-based dynamic adaptivity are no longer applicable; therefore, only extrinsic well-balanced solution limits can be reconstructed across the heterogeneously-sized adjacent elements based on heuristic averaging on a graded grid, as in conventional adaptive grid refinement methods (Kesserwani and Liang, 2012a; Liang et al., 2015). Previous works suggested that static adaptivity with a finite volume solver is suited to simulate slow to gradual, fluvial flood flow propagation (Hou et al., 2018), but have rarely assessed the ability of the (multi)wavelet-based MRA to generate the fixed grid. Within this scope, the work of Özgen-Xian et al. (2020) only explored the MRA of HW to generate unstructured triangular grids over which a zero-inertia flow solver was adapted to simulate overland flow, reporting moderate speedups. Also, the authors concluded that pre-processing the DEM data into topographic slopes for use as input to the MRA of HW significantly improve the sensibility of the predicted resolution scales for the fixed grid (Özgen-Xian et al., 2020). This suggests that there is a benefit from using the MRA of MWs as a better alternative to generate the fixed grid. This will be demonstrated in this work for static DG2 and FV1 adaptivity alongside with dynamic MWDG2 and HWFV1 adaptivity.

The rest of the paper is organised as follows. Section 2 overviews the dynamic MWDG2 and HWFV1 adaptivity that uses the MRA process to analyse and scale their numerical solutions (DG2 piecewise-planar and FV1 piecewise-constant) and update the grid over time (Section 2.1). It

follows with the description of the static DG2 and FV1 adaptivity (Section 2.2) and, of the GPU implementation of the MRA process to enable GPU resident dynamic MWDG2 and HWFV1 adaptivity (Section 2.3). In Section 3, the (multi)wavelet-based dynamic MWDG2 and HWFV1 adaptivity is assessed against the standard static DG2 and FV1 adaptivity for three field-scale test cases featured by slow, gradual and rapidly propagating flood flows with reference to finest uniform grid DG2 simulations. The assessments include demonstrating the merit of the MRA of MWs to generate the fixed grid for static DG2 and FV1 adaptivity that is also the initial grid for dynamic MWDG2 and HWFV1 adaptivity. Section 4 provides an efficiency analysis in terms of runtime cost and percentage of reduction in the number of elements with respect to the finest uniform grid to identify when static or dynamic adaptivity can be used as an alternative to uniform grid flood simulators. Section 5 concludes on the extent of practical utility of static and dynamical adaptivity in relation to the properties of the flood simulation project.

2. Computational methods

Flow solvers with dynamic and static adaptivity are explored as alternatives to their established uniform grid DG2 and FV1 solvers (Shaw et al., 2021). The solvers are formulated to provide numerical solutions of the depth-averaged shallow water equations in a conservative form (Eq. (1)).

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

∂ represents a partial derivative operator, $\mathbf{U}(x, y, t) = [h, q_x, q_y]^T$ is the vector of flow variables at time t and location (x, y) which contains the water depth h (m) and the discharges per unit width, $q_x = hu$ ($m^2 s^{-1}$) and $q_y = hv$ ($m^2 s^{-1}$), involving the depth-averaged horizontal velocities u ($m s^{-1}$) and v ($m s^{-1}$), respectively; $\mathbf{F} = [q_x, q_x^2 h^{-1} + 0.5gh^2, q_x q_y h^{-1}]^T$ and $\mathbf{G} = [q_y, q_x q_y h^{-1}, q_y^2 h^{-1} + 0.5gh^2]^T$ are the components of the physical flux and g ($m s^{-2}$) denotes the gravity acceleration constant. $\mathbf{S}_b = [0, -gh\partial_x z, -gh\partial_y z]^T$ is the topography source term vector including the bed slopes and the friction source term vector incorporating friction effects is denote by $\mathbf{S}_f = [0, -C_f u \sqrt{u^2 + v^2}, -C_f v \sqrt{u^2 + v^2}]^T$ where $C_f = gn_M^2/h^{1/3}$ is a friction function in which n_M ($m^{1/6}$) refers to the Manning's roughness parameter.

Eq. (1) is numerically solved on a 2D spatial domain of dimension $l_x \times l_y$. The domain is meshed using a grid made of non-overlapping, heterogeneously-sized square elements. The finest resolution R on this grid will be taken to match the raster-formatted digital elevation model

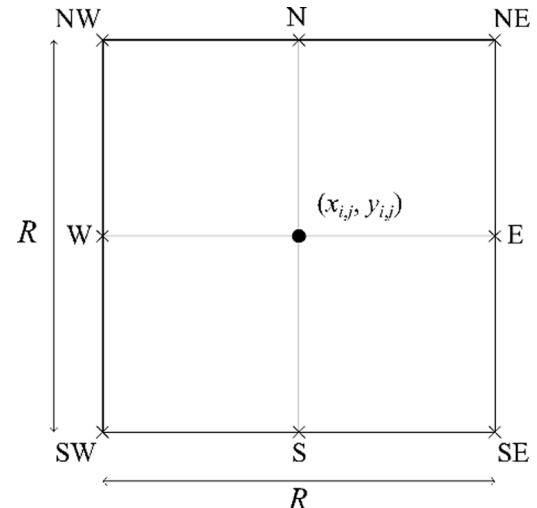


Fig. 1. Sketch of an element, centred at $(x_{i,j}, y_{i,j})$, belonging to the finest grid, at DEM resolution, R . N, E, S and W mark the northern, eastern, southern and western interface centres and NW, NE, SW and SE mark the four vertices.

(DEM). A grid element at resolution R is shown in Fig. 1, indexed by i, j and centred at points $(x_{i,j}, y_{i,j})$. Two categories of solvers are presented: (multi)wavelet-based dynamic MWDG2 and HWFV1 adaptivity, scaling their numerical solutions and grid resolutions in time; and, standard static DG2 and FV1 adaptivity, adapting their numerical solutions on a fixed grid.

Dynamic MWDG2 adaptivity uses the MRA of MWs to decompose, analyse and assemble piecewise-planar DG2 solutions on a dynamically adaptive grid (Section 2.1). From dynamic MWDG2 adaptivity, that of HWFV1 can be obtained by degrading its formulation to use the HW to decompose, analyse and assemble piecewise-constant FV1 solutions (Section 2.1.4). The CPU implementation of dynamic MWDG2 and HWFV1 adaptivity can be found in Kesserwani et al. (2019) and Kesserwani and Sharifian (2020), with an exhaustive analysis of their performance for academic and laboratory-scale test cases. As the aim is to assess the potential of dynamic MWDG2 and HWFV1 adaptivity for complex real-world flood simulation, only the (multi)wavelet-based MRA process is briefly overviewed (next in Section 2.1).

(Multi)wavelet-based dynamic adaptivity will be compared with standard static adaptivity for three realistic test cases (Section 3). The initial grid for launching dynamic MWDG2 and HWFV1 adaptivity, will be the same as the fixed grid used with static DG2 and FV1 adaptivity (Section 3.1). This fixed grid will be generated by first applying the MRA of MWs to the DEM raster file (Section 2.1.2) and then ensuring that it is graded (Section 2.2.1). The descriptions of the DG2 and FV1 solvers' formulation for a uniform resolution grid can be found in Shaw et al. (2021). Therefore, only their solution adaptation procedure on the fixed grid will be described for a portion with an interface shared by heterogeneously-sized adjacent elements (Section 2.2). A new GPU implementation of dynamic MWDG2 and HWFV1 adaptivity is proposed to make them run in parallel on the GPU (Section 2.3) and boost speedups with respect to the CPU counterparts (Section 4).

2.1. Dynamic MWDG2 adaptivity

For dynamic MWDG2 adaptivity, the discrete vector of flow variables, $\mathbf{U}_h(x,y, t)$, and the discrete scalar topography, $z_h(x,y)$, are shaped as local piecewise-planar solutions over each element starting from the finest uniform grid, at resolution R (Fig. 1). This is done within the framework of a simplified DG2 formulation, which spans piecewise-planar flow solutions and topography function, on a 2D truncated tensor-product of the scaled Legendre basis Φ to which MWs bases are adapted (Kesserwani and Sharifian, 2020). With a truncated tensor-product basis, the simplified DG2 formulation is “slope-decoupled” (Kesserwani et al., 2018), and therefore operates on a similar stencil to that of Godunov-type finite volume flood models (Ayog et al., 2021). On the slope-decoupled stencil, the simplified DG2 formulation evaluates the fluxes at the centres of the elements' interfaces where it must ensure continuity of the topography function to avoid spurious momentum errors at wet-dry fronts crossing steep bed-slopes – a desired property to achieve within dynamic MWDG2 adaptivity (Caviedes-Voullième et al., 2020). The simplified DG2 formulation was demonstrated to deliver as second-order accurate predictions as the classical DG2 formalism based on a full tensor-product stencil, leading to a speedup of 2.6 times. Moreover, it achieves superior predictions than finite volume Godunov-type flood models for real-field flood modelling without activating local slope limiting, and this boosts the speedups to 5 times (Ayog et al., 2021), maximising efficiency for flood modelling applications (Shaw et al., 2021).

As shown in Eq. (2), $\mathbf{U}_h(x,y, t)$ contains an equation of a plane per physical component, each defined by three coefficients of an average (denoted by subscript 0) and of two slopes in the x and y directions (denoted by subscripts $1x$ and $1y$, respectively).

$$\mathbf{U}_h(x, y, t) = \mathbf{U}_{i,j} \cdot \Phi = \begin{bmatrix} h_{i,j,0} & h_{i,j,1x} & h_{i,j,1y} \\ q_{xi,j,0} & q_{xi,j,1x} & q_{xi,j,1y} \\ q_{yi,j,0} & q_{yi,j,1x} & q_{yi,j,1y} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2\sqrt{3}(x - x_{i,j})/R \\ 2\sqrt{3}(y - y_{i,j})/R \end{bmatrix} \quad (2)$$

In Eq. (2), “ \cdot ” stands for scalar product and therefore, $\mathbf{U}_{i,j}$ is a matrix containing the flow coefficients defining the piecewise-planar solutions for the three components of the flow vector. The piecewise-planar representation for $z_h(x,y)$ can be produced in a similar way,

$$z_h(x, y) = \mathbf{z}_{i,j} \cdot \Phi = [z_{i,j,0} \ z_{i,j,1x} \ z_{i,j,1y}] \cdot \begin{bmatrix} 1 \\ 2\sqrt{3}(x - x_{i,j})/R \\ 2\sqrt{3}(y - y_{i,j})/R \end{bmatrix} \quad (3)$$

In Eq. (3), $\mathbf{z}_{i,j}$ is the vector of the topography coefficients for defining a piecewise-planar representation. The topography coefficients do not vary in time and should be carefully initialised from a DEM raster file (Section 2.1.1). The time-dependant flow coefficients $\mathbf{U}_{i,j}$ can be initialised (component-wise) in a similar way from the initial conditions for the flow variables converted into raster files.

2.1.1. Initialisation of coefficients on the finest uniform grid

The initialisation procedure is based on ensuring that the piecewise-planar topography representation, $z_h(x,y)$, remains continuous at the interface centres of an element (denoted by N, S, E and W, in Fig. 1). As demonstrated in Kesserwani et al. (2018), ensuring continuity at these interface centres is key to robustly evaluate the discrete fluxes and topography source terms, which respects the well-balanced property with the presence of wet-dry fronts across steep bed-slopes. Such a piecewise-planar topography projection can be constructed from a DEM, by first taking the DEM raster values to be located at the NW, NE, SW and SE vertices (Fig. 1), and then aggregating an average value $z_{i,j}^N = 0.5(z_{i,j}^{NW} + z_{i,j}^{NE})$ at N, and similarly $z_{i,j}^E, z_{i,j}^S$ and $z_{i,j}^W$ values at S, E and W. From these values, the desired average, $z_{i,j,0}$, and slope coefficients, $z_{i,j,1x}$ and $z_{i,j,1y}$, can be defined,

$$z_{i,j,0} = \frac{1}{2} [z_{i,j}^E + z_{i,j}^W] = \frac{1}{2} [z_{i,j}^N + z_{i,j}^S] = \frac{1}{4} [z_{i,j}^{NE} + z_{i,j}^{NW} + z_{i,j}^{SE} + z_{i,j}^{SW}] \quad (4a)$$

$$z_{i,j,1x} = \frac{1}{2\sqrt{3}} [z_{i,j}^E - z_{i,j}^W] = \frac{1}{4\sqrt{3}} [z_{i,j}^{NE} - z_{i,j}^{NW} + z_{i,j}^{SE} - z_{i,j}^{SW}] \quad (4b)$$

$$z_{i,j,1y} = \frac{1}{2\sqrt{3}} [z_{i,j}^N - z_{i,j}^S] = \frac{1}{4\sqrt{3}} [z_{i,j}^{NE} - z_{i,j}^{SE} + z_{i,j}^{NW} - z_{i,j}^{SW}] \quad (4c)$$

Even when there are no initial flow conditions, Eq. (4) is also applied to initialise zero-valued vectors for $[h_{i,j,0}, h_{i,j,1x}, h_{i,j,1y}], [q_{xi,j,0}, q_{xi,j,1x}, q_{xi,j,1y}]$ and $[q_{yi,j,0}, q_{yi,j,1x}, q_{yi,j,1y}]$, to initialise the matrix of flow coefficients $\mathbf{U}_{i,j}$ (Eq. (2)). In what follows, for simplicity of presentation, the MRA process, which is responsible for dynamic adaptivity, will be described component-wise using the scalar $s \in \{h, q_x, q_y, z\}$ to represent the physical components of the flow vector (time-variant) and the topography (time-invariant).

2.1.2. MRA of MWs to generate the 2D adaptive grid

The MRA of MWs starts from the finest uniform grid, at resolution R , using the initial coefficients $\mathbf{s}^{\text{fine}} = [s_{i,j,0}, s_{i,j,1x}, s_{i,j,1y}]$ attributed to a maximum resolution level L (leading to the finest resolution of the DEM). Now, the finest uniform grid, at resolution level L , contains $(2^L M) \times (2^L N)$ elements, $M = l_x/(2^L R)$ and $N = l_y/(2^L R)$, below which there are uniform grids with subsequently lower resolution levels of $L - 1, \dots, 1, 0$ that forms a hierarchy with the coarsest uniform grid, at resolution level 0, made of $M \times N$ elements (i.e. a single element in this work, with $M = N = 1$). The MRA process first applies the encoding operation to extract and analyse the flow vector and topography coefficients at the grids with lower resolution levels, $L - 1, \dots, 1, 0$, to then apply the decoding operation to derive the relevant coefficients that will form the

2D adaptive grid.

From the coefficients on the grid at resolution level L , \mathbf{s}^{fine} , the encoding operation can be applied to produce the coefficients on the grid with twice-coarser resolution at level $L - 1$, $\mathbf{s}^{\text{coarse}}$, and to also extract their vectors of the details $\mathbf{d}_H^{\text{coarse}}$, $\mathbf{d}_V^{\text{coarse}}$ and $\mathbf{d}_D^{\text{coarse}}$ living at resolution level $L - 1$. These vectors of details belong to the basis of MWs, ψ , and represent the encoded difference between the scaling basis ϕ at resolution level L and its parent basis at resolution level $L - 1$ (e.g. as shown in Fig. 1 in Shaw et al., 2020). Therefore, $\mathbf{d}_H^{\text{coarse}}$, $\mathbf{d}_V^{\text{coarse}}$ and $\mathbf{d}_D^{\text{coarse}}$ encapsulate the encoded differences in coefficients between the two resolution levels, L and $L - 1$, along the horizontal, vertical, and diagonal directions, respectively. Namely, $\mathbf{s}^{\text{coarse}}$, $\mathbf{d}_H^{\text{coarse}}$, $\mathbf{d}_V^{\text{coarse}}$ and $\mathbf{d}_D^{\text{coarse}}$ on a parent element at resolution level $L - 1$ can be encoded from its four coefficients of their child elements at resolution level L using the formulae described below in Eq. (5),

$$\mathbf{s}^{\text{coarse}} = \mathbf{HH}^0 \mathbf{s}_{[0]}^{\text{fine}} + \mathbf{HH}^1 \mathbf{s}_{[2]}^{\text{fine}} + \mathbf{HH}^2 \mathbf{s}_{[1]}^{\text{fine}} + \mathbf{HH}^3 \mathbf{s}_{[3]}^{\text{fine}} \quad (5a)$$

$$\mathbf{d}_H^{\text{coarse}} = \mathbf{GA}^0 \mathbf{s}_{[0]}^{\text{fine}} + \mathbf{GA}^1 \mathbf{s}_{[2]}^{\text{fine}} + \mathbf{GA}^2 \mathbf{s}_{[1]}^{\text{fine}} + \mathbf{GA}^3 \mathbf{s}_{[3]}^{\text{fine}} \quad (5b)$$

$$\mathbf{d}_V^{\text{coarse}} = \mathbf{GB}^0 \mathbf{s}_{[0]}^{\text{fine}} + \mathbf{GB}^1 \mathbf{s}_{[2]}^{\text{fine}} + \mathbf{GB}^2 \mathbf{s}_{[1]}^{\text{fine}} + \mathbf{GB}^3 \mathbf{s}_{[3]}^{\text{fine}} \quad (5c)$$

$$\mathbf{d}_D^{\text{coarse}} = \mathbf{GC}^0 \mathbf{s}_{[0]}^{\text{fine}} + \mathbf{GC}^1 \mathbf{s}_{[2]}^{\text{fine}} + \mathbf{GC}^2 \mathbf{s}_{[1]}^{\text{fine}} + \mathbf{GC}^3 \mathbf{s}_{[3]}^{\text{fine}} \quad (5d)$$

The subscripts of the coefficients $\mathbf{s}_{[]}^{\text{fine}}$ refer to the indexing of the four child elements at level L , with reference to their parent element at level $L - 1$. $\mathbf{HH}^{0,1,2,3}$ are 3×3 low-pass filter matrices and $\mathbf{GA}^{0,1,2,3}$, $\mathbf{GB}^{0,1,2,3}$ and $\mathbf{GC}^{0,1,2,3}$ are 3×3 high-pass filter matrices. These filters were derived from the relationship linking ψ to ϕ at resolution level L and its father basis at resolution level $L - 1$. The explicit expressions of these filters are available in Kesserwani and Sharifian (2020), including the ones for HWFV1.

The formulae in Eq. (5) are generalisable to any two subsequent resolution levels (Kesserwani and Sharifian, 2020). By recursive application of Eq. (5), \mathbf{s}^{fine} at resolution level L is now compressed into a series of details living on the hierarchy of uniform grids at subsequently lower resolution levels ($L - 1, \dots, 1, 0$). These details form a tree-like structure and can be summed over the coefficients $\mathbf{s}^{\text{coarse}}$ associated with the coarsest uniform grid (resolution level 0). On this hierarchy, the details extracted for all the physical components (flow vector and topography) are analysed all together. These details become increasingly significant with increasing non-smoothness in any of the physical features but remain small otherwise. The time-invariant details of the topography must be reconsidered with those extracted from the flow vector (time-variant) to ensure that no coarsening in grid resolution is allowed beyond the resolution dictated by the DEM.

The significance of all the details is measured by comparing their normalised magnitude to an error threshold parameter ϵ (user-specified). For shallow flow modelling, a value in the magnitude of $\epsilon = 10^{-3}$ was identified appropriate with dynamic MWDG2 (respectively HWFV1) adaptivity to preserve the predictive accuracy of the DG2 solver (respectively FV1) run on the finest uniform grid resolution R at a competitively affordable runtime, namely less or close to that of the FV1 solver run on the finest uniform grid resolution R (Kesserwani and Sharifian, 2020). For realistic simulations involving real DEMs with largely dry portions, using stricter ϵ values lead to a marginal improvement in the accuracy (Kesserwani and Sharifian, 2020), but would introduce too much rise in the runtime cost up to making dynamic adaptivity impracticable.

A normalised details magnitude of any component $s \in \{h, q_x, q_y, z\}$ is computed as $\bar{d} = |d|/\max(1, |s_0|)$. The term $|d|$ denotes the maximum of the magnitude of the three details along the horizontal, vertical and diagonal directions, and $|s_0|$ is the maximum of all the average coefficients available on the finest uniform grid (at resolution level L). The normalised details are computed for all the elements on the hierarchy of

uniform grids (with resolution levels $L - 1, \dots, 0$). Details at elements where $\bar{d} > \epsilon$ are classified as significant and should be retained, and such elements are flagged to produce their coefficients, via decoding (Kesserwani and Sharifian, 2020). These elements will be referred to as “leaf elements” and are such that their details on the tree of details either stopped being significant or, otherwise, belong to the finest grid. A schematic view of leaf elements on the hierarchy of grids can be found below in Fig. 4, as part of the GPU parallelisation (Section 2.3).

To produce the leaf elements’ coefficients, the process of decoding is applied for the tree of details all over the hierarchy of grids. Decoding starts from applying Eq. (6) to the coefficients and their encoded details on the coarsest uniform grid at resolution level 0 ($\mathbf{s}^{\text{coarse}}$, $\mathbf{d}_H^{\text{coarse}}$, $\mathbf{d}_V^{\text{coarse}}$ and $\mathbf{d}_D^{\text{coarse}}$) to create the coefficients of their four child elements on the grid at resolution level 1 ($\mathbf{s}_{[0]}^{\text{fine}}, \mathbf{s}_{[1]}^{\text{fine}}, \mathbf{s}_{[2]}^{\text{fine}}$ and $\mathbf{s}_{[3]}^{\text{fine}}$).

$$\mathbf{s}_{[0]}^{\text{fine}} = [\mathbf{HH}^0]^T \mathbf{z}^{\text{coarse}} + [\mathbf{GA}^0]^T \mathbf{d}_H^{\text{coarse}} + [\mathbf{GB}^0]^T \mathbf{d}_V^{\text{coarse}} + [\mathbf{GC}^0]^T \mathbf{d}_D^{\text{coarse}} \quad (6a)$$

$$\mathbf{s}_{[2]}^{\text{fine}} = [\mathbf{HH}^1]^T \mathbf{z}^{\text{coarse}} + [\mathbf{GA}^1]^T \mathbf{d}_H^{\text{coarse}} + [\mathbf{GB}^1]^T \mathbf{d}_V^{\text{coarse}} + [\mathbf{GC}^1]^T \mathbf{d}_D^{\text{coarse}} \quad (6b)$$

$$\mathbf{s}_{[1]}^{\text{fine}} = [\mathbf{HH}^2]^T \mathbf{z}^{\text{coarse}} + [\mathbf{GA}^2]^T \mathbf{d}_H^{\text{coarse}} + [\mathbf{GB}^2]^T \mathbf{d}_V^{\text{coarse}} + [\mathbf{GC}^2]^T \mathbf{d}_D^{\text{coarse}} \quad (6c)$$

$$\mathbf{s}_{[3]}^{\text{fine}} = [\mathbf{HH}^3]^T \mathbf{z}^{\text{coarse}} + [\mathbf{GA}^3]^T \mathbf{d}_H^{\text{coarse}} + [\mathbf{GB}^3]^T \mathbf{d}_V^{\text{coarse}} + [\mathbf{GC}^3]^T \mathbf{d}_D^{\text{coarse}} \quad (6d)$$

Decoding is recursively applied, in ascending order, while adding up the retained details to create the coefficients on child elements (Kesserwani and Sharifian, 2020). The derived coefficients and the indices of their elements can finally be sorted and assembled on the 2D adaptive grid made of non-overlapping elements involving various resolutions between R and $2^L R$.

2.1.3. Time-update of coefficients and of the 2D adaptive grid

On the 2D adaptive grid the flow coefficients \mathbf{U}_{ij} at time level n should be updated to time level $n + 1$, by a two-stage Runge-Kutta scheme involving discrete spatial DG2 operators (Kesserwani et al., 2018), as shown in Eq. (7). The flow coefficients are updated at the wet elements, which are detected before each time stage.

$$\mathbf{U}_{ij}^{n+1/2} = \mathbf{U}_{ij}^n + \Delta t \mathbf{L}(\mathbf{U}^n) \quad (7a)$$

$$\mathbf{U}_{ij}^{n+1} = \frac{1}{2} \left[\mathbf{U}_{ij}^n + \mathbf{U}_{ij}^{n+1/2} + \Delta t \mathbf{L}(\mathbf{U}^{n+1/2}) \right] \quad (7b)$$

The time step Δt is calculated according to the CFL stability condition using the maximum stable Courant number of 0.33 (Cockburn and Shu, 2001), and \mathbf{L} is a 3×3 matrix including the discrete spatial operator vectors \mathbf{L}_0 , \mathbf{L}_{1x} and \mathbf{L}_{1y} . These local operators contain robust evaluation of the spatial fluxes and topography gradient terms, to ensure the well-balanced property with wet-dry front treatments (Kesserwani and Sharifian, 2020). These include Riemann flux evaluation at the four interface centres, in Fig. 1, from the reconstructed limits of the piecewise-planar solutions at both sides of each interface centre using the HLL Riemann solver, which is one recommend choice (Kesserwani et al., 2008). The friction source term \mathbf{S}_f is not explicitly included in \mathbf{L} and should instead be integrated elementwise at the start of each time step as described in Shaw et al. (2021).

The updated coefficients \mathbf{U}_{ij}^{n+1} are now available on the 2D adaptive grid assembled at the previous time level n . The information in \mathbf{U}_{ij}^{n+1} , i.e. $\mathbf{s} = [s_0, s_{1x}, s_{1y}]$ with $s \in \{h, q_x, q_y, z\}$, should now be used to regenerate a new 2D adaptive grid. This can be done by reapplying MRA of MWs for adaptive grid generation (Section 2.1.2), after initialising zero details for the flow vector’s coefficients, $s \in \{h, q_x, q_y\}$, at the elements in the hierarchy of grids that did not belong to the 2D adaptive grid at time level n (Kesserwani and Sharifian, 2020).

2.1.4. Degradation to dynamic HWFV1 adaptivity

For dynamic HWFV1 adaptivity, the discrete vector of flow variables, $\mathbf{U}_h(x,y, t)$, and the discrete scalar topography, $z_h(x,y)$, are shaped as local piecewise-constant solutions over each element and the basis ϕ is a scalar equal to 1. Each physical component of $\mathbf{U}_h(x,y, t)$ takes the form of a flat plane that is defined by one coefficient,

$$\mathbf{U}_h(x,y, t) = \mathbf{U}_{i,j} \cdot \phi = \begin{bmatrix} h_{i,j,0} \\ q_{x,i,j,0} \\ q_{y,i,j,0} \end{bmatrix} \cdot 1 = \begin{bmatrix} h_{i,j,0} \\ q_{x,i,j,0} \\ q_{y,i,j,0} \end{bmatrix} \quad (8)$$

Here, $\mathbf{U}_{i,j}$ is a vector of flow coefficients and $z_h(x,y)$ is represented in the same way as,

$$z_h(x,y) = \mathbf{z}_{i,j} \cdot \phi = z_{i,j,0} \cdot 1 = z_{i,j,0} \quad (9)$$

The initialisation of coefficients on the finest uniform grid only uses Eq. (4a). Then, MRA of HW is applied for the 2D adaptive grid generation in a similar manner, considering simplified (scalar) expression for the high-pass and low-pass filters (Kesserwani and Sharifian, 2020). On the 2D adaptive grid, the update of the coefficients is performed using a forward Euler time-stepping scheme and L only includes the L_0 operator (Eq. (10)). This operator is based on the FV1 spatial discretisation where the HLL solver's Riemann flux evaluations consider piecewise-constant solution limits at the two sides of each interface.

$$\mathbf{U}_{i,j}^{n+1} = \mathbf{U}_{i,j}^n + \Delta t \mathbf{L}(\mathbf{U}^n) \quad (10)$$

The time step Δt is calculated according to the same CFL stability condition as the DG2 solver but with the maximum stable Courant number of 0.5 (Kesserwani and Liang, 2012b). Once $\mathbf{U}_{i,j}^{n+1}$ are produced, the regeneration of the 2D adaptive grid is performed as with dynamic MWDG2 adaptivity (Section 2.1.3).

Within the dynamic MWDG2 and HWFV1 adaptivity, the filter formulae (Eqs. 5 and 6) are applied every time step within the MRA process to reconstruct intrinsic well-balanced solution limits for any matching resolution required across the heterogeneously-sized adjacent elements. The reconstruction procedure is generally applicable without grading (Fig. 2a) the 2D adaptive grid (Kesserwani and Sharifian 2020).

2.2. Static DG2 and FV1 adaptivity on a fixed, graded grid

Static DG2 and FV1 adaptivity operates on a fixed, or static, grid on which elementwise update of the flow coefficients is achieved as described previously, respectively via Eqs. (7) and (10). The 2D adaptive grid is thus generated once, after applying the MRA of MWs (only once) to analyse the time-invariant topography coefficients (Section 2.1.2). Compared to the grid generated from using the MRA of HW, that generated from using the MRA of MWs sensibly boosts the resolution coarsening (Kesserwani et al., 2019; Kesserwani and Sharifian, 2020) and will be preferred to run the DG2 and FV1 solvers on a graded grid

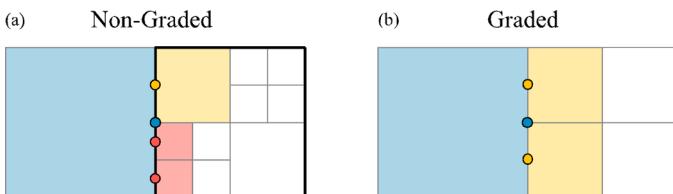


Fig. 2. Schematic layout of non-homogeneous interfaces shared between neighbouring elements at different levels: (a) non-graded grid, where a coarse element (in blue) is adjacent to one- and two-level finer elements (in yellow and red, respectively); (b) graded grid, where a coarse element (in blue) can only be adjacent to one-level finer elements (in yellow). The yellow and red points denote the fine element interface centres, and the blue point denotes the coarse element interface centre. At these points, the limits of the local solutions are required for Riemann flux evaluations.

(shown later in Section 3.1), which ensures that no element is more than one resolution level finer or coarser than its neighbouring elements (Fig. 2b). The grading of the fixed grid is necessary within the static adaptivity, as the MRA process is switched off over time, which makes the rigorous filter formulae not applicable.¹ After grading, conventional adaptive grid refinement treatments can be applied to reconstruct extrinsic well-balanced solution limits at both side of the interface separating the heterogeneously-sized adjacent elements (detailed next in Section 2.2.1).

2.2.1. Averaging of solution limits across adjacent elements of different sizes

Without grading, the grid could include adjacent elements of different sizes, where the numerical solutions are more than one resolution level apart from each side of the elements' interface. An example is shown in Fig. 2a, where the red elements are two resolution levels finer than the neighbouring blue element. At both sides of the finer element interface centres, precisely well-balanced solution limits can be unproblematically evaluated to integrate the fluxes within the update steps (Eqs. 7 or 10), and thereby ultimately update the flow coefficients at the finer interface elements. However, the challenge arises when updating the flow coefficients at the coarse interface element: From its right side (Fig. 2a), there is no rigorous way to aggregate the local solution limits along the finer element interface into a precisely well-balanced local solution limit at the coarse element interface centre. With grading, this challenge can be reduced to estimating a coarse resolution limit by heuristic averaging from the limits available at the centres of the finer interface elements (from the right side) following the common reported procedure in conventional adaptive grid refinement techniques (Borthwick et al., 2000; Popinet, 2003; Kesserwani and Liang, 2012a; Donat et al., 2014; BMT (2018); Ghazizadeh et al., 2020; Dunning et al., 2020). For example, as shown in Fig. 2b, the required solution limit, at the right side, is averaged from the solution limits at the interface centres of the two finer elements (yellow points). This averaging is exact for flat, piecewise-constant solution representations, as with the FV1 solver, taking the same value along each of the small element interfaces, including at the centres. With uneven piecewise-planar solution representations, as with the DG2 solver, the averaging is heuristic as it only considers the values at the centres of each of the small element interfaces, any of which may be different than the other values along a small element interface. The heuristic nature of the averaging process within the DG2 solver may manifest in local inaccuracies in the predictions when there is high variability in a flow variable (shown in Section 3).

2.3. GPU parallelisation with dynamic adaptivity

To effectively parallelise (multi)wavelet-based dynamic adaptivity on the GPU, ensuring coalesced memory access and avoiding warp divergence are essential (NVIDIA, 2021). In CUDA programming, data of the coefficients and details should reside in GPU memory that is accessed by threads, or workers. Threads should access contiguous memory locations for coalescing, and batches of 32 threads, or warps, must avoid the execution of differing instructions to avoid divergence. These requirements are not difficult to achieve when parallelising FV1 and DG2 on a uniform grid with equally-sized elements (Shaw et al., 2021). Similar can be said to the updates (Eqs. 7 and 10) within dynamic HWFV1 and MWDG2 adaptivity, as matching resolution representations is a priori ensured from the filter formulae (Eqs. 5 and 6). Therefore, it suffices to describe how these requirements can be met for the MRA process that is responsible for achieving the dynamic

¹ Applying the rigorous filter formulae (Eqs. 5 and 6) would lead to spurious freezing the fluxes in time, across the interface, as the details of the flow vector do not get updated over time.

adaptivity every time step (Section 2.1.2). To do so, the recursive operations featured in the MRA process must be reworked; namely to ensure coalescing during the encoding of details and the decoding of leaf elements' data, and to avoid warp divergence when identifying the indices of the leaf elements. These will be described for a hierarchy of grids up to $L=3$, without loss of generality. The algorithmic implementation of the MRA process on the GPU is presented in Chowdhury et al. (2022), which is newly extended to encompass dynamic MWDG2 adaptivity.

Coalescing has been ensured by spanning the hierarchy of grids using Z order curves to continuously index all the elements living within the hierarchy. For each grid in the hierarchy, a Z order curve is created by following the so-called Morton code of each grid element, by first interleaving the bits of the indices X and Y of the element in binary representation (Fig. 3a) and then converting of the resulting bit interleaved index to decimal form as shown in Fig. 3b (Sagan, 1994; Bader, 2013; Baert et al., 2013; Chowdhury et al., 2022). Z order curves are created for every grid in the hierarchy starting from the coarsest to the finest resolution grid to end up with a continuous span of the element indices all over the hierarchy (Fig. 3c). With this span, the coefficients and details needed for encoding and decoding are now resident in adjacent memory locations (zoomed-in portion in Fig. 3c). Although the leaf elements' coefficients and indices are available on the hierarchy, the coefficients must be fetched to assemble the adaptive grid in parallel on the GPU, and this can only be achieved by simultaneous traversal of the tree of details.

Simultaneous tree traversal has been achieved based on the algorithm presented by Karras (2012), initially designed for computer graphic applications. The algorithm is applied to simultaneously climb the tree of details after continuous indexing of Z order curves, as shown in Fig. 4a. The starting point is to launch as many threads as the number of elements on the finest uniform grid, namely $2^3 \times 2^3 = 64$ threads. The threads are tasked to start simultaneous traversal from the single element on the coarsest grid. Each thread attempts to climb to the

element on the finest grid indexed with the decimal converted from the Morton code (Fig. 4a). During the simultaneous traversal, by checking against the tree of details, a thread stops as soon as it reaches a leaf element and records its index in memory. An example can be seen in the hierarchy in Fig. 4a-4b, where a thread will attempt to reach the element with index 45 climbing along the elements with indices {0, 2, 11, 45} to stop at the leaf element with index 45 and record it in memory. In doing so, warp divergence is avoided because adjacent threads perform similar traversals due to the continuous indexing of the Z curves spanning the hierarchy of grids, and many of the threads climb to the same leaf elements without reaching any element on the finest grid. For example, as indicated in Fig. 4a-4b, four threads climb up the elements with indices 0, 2 and 11, one by one, to finally settle for the elements with indices {45, 46, 47, 48} located on the finest grid. In contrast, sixteen threads settle for the leaf element with index 1, which does not belong to the finest grid, each of them recording the same index 1 in memory and this results in duplicates.

After the parallel tree traversal, an array of indices is recorded in memory including the duplicate indices, shown in Fig. 4b. These duplicate indices are then scrutinised by the threads to also identify and record the indices of the four elements neighbouring the leaf elements from the east, west, north and south. For example, the thread that has already recorded the leaf element with index 1 (at the bottom right of the array of 16 elements with index 1), will then be tasked to record the indices of the leaf element's direct neighbours, namely the indices 1, 47, 1 and 14 shown in Fig. 4b. The indices of the four neighbour elements are also stored in memory, in turn resulting in duplicates.

The recorded duplicate indices of the leaf elements and their four neighbours are then removed from memory using stream compaction (Billeter et al., 2009). After compaction, the remaining indices that are unique to the leaf elements, relevant to form the 2D adaptive grid, are identified together with those of their four neighbouring elements. The leaf elements' coefficients and indices are then used to fetch the 2D adaptive, as shown in Fig. 4c, on which elementwise update of the

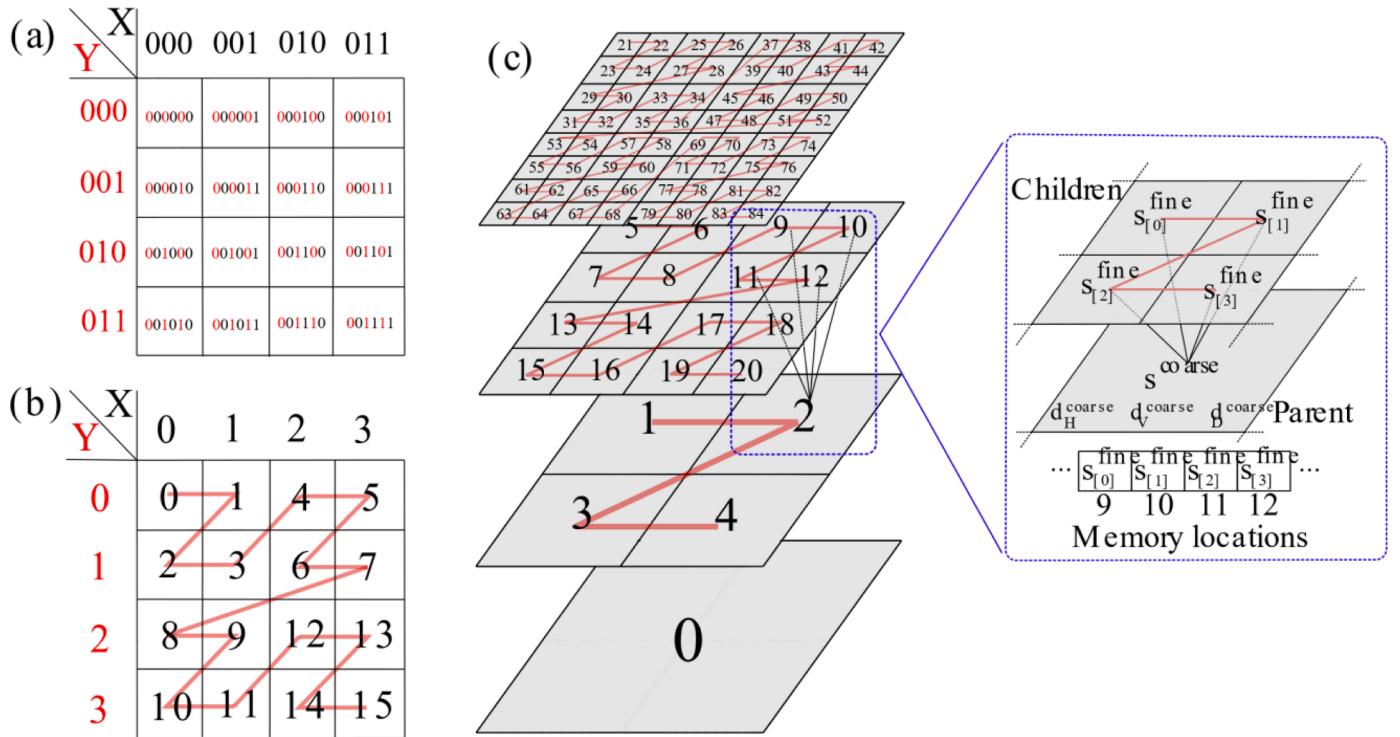


Fig. 3. (a) A demonstration on a 4×4 grid of how the bits of the indices X and Y in binary representation are interleaved; (b) The decimal form of the bit interleaved indices on the same 4×4 grid leading to Z order curves; and (c) Morton codes generated from Z order curves spanning the hierarchy of grids. The zoomed-in portion shows the coefficients and details on a parent element and the coefficients on its four children elements, which are resident in adjacent memory locations.

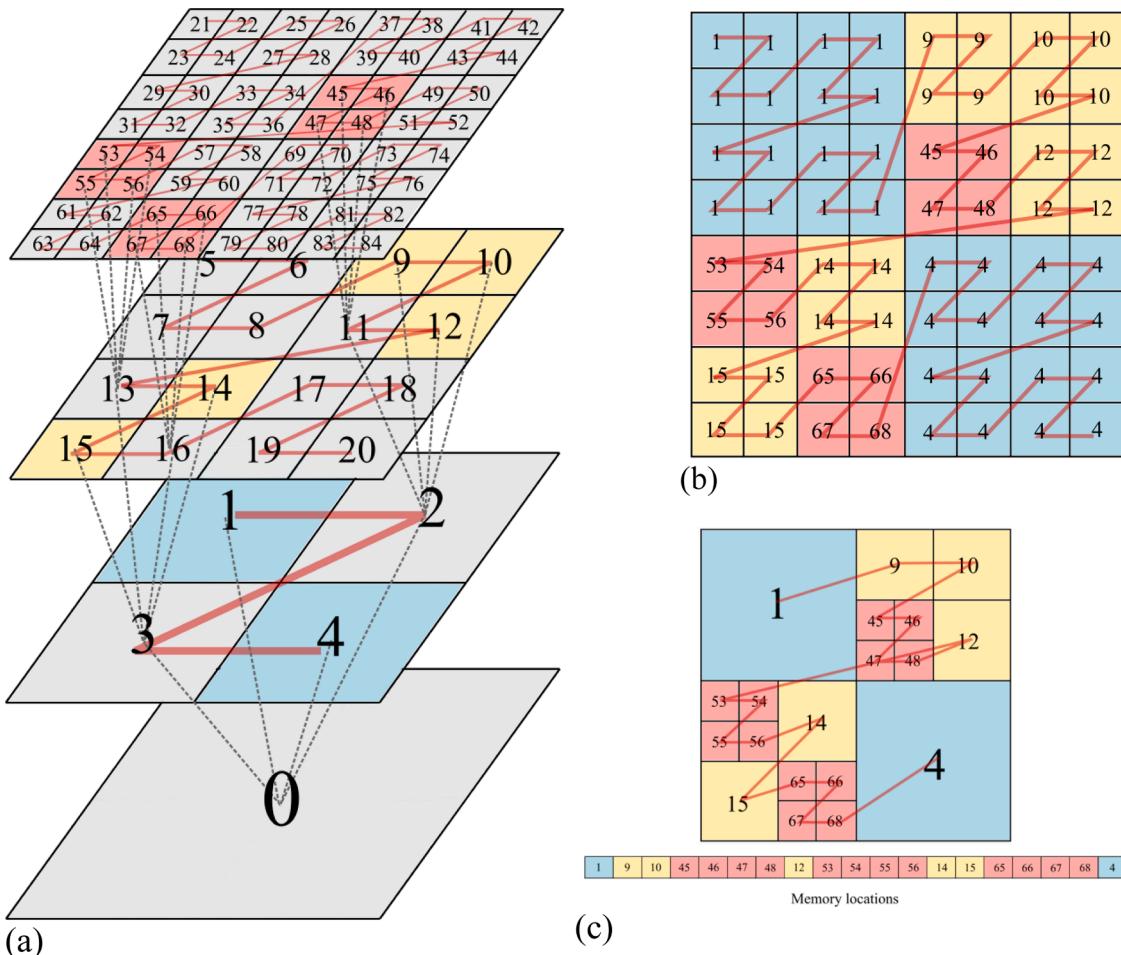


Fig. 4. Description of the parallel tree traversal algorithm. (a) Climbing the hierarchy after continuous indexing of Z order curves up to leaf elements (red, yellow and blue coloured); (b) Leaf elements' indices recorded by the threads after climbing. There are duplicate indices for elements not belonging to the finest grid, which have to be removed after reusing them to detect the indices of the leaf elements' four neighbours; and, (c) Assembly of the unique indices of the leaf elements making up the 2D adaptive grid.

coefficients is applied. This completes the process for dynamic HWFV1 adaptivity. However, another step remains needed to complete the process with dynamic MWDG2 adaptivity, namely after the first Runge-Kutta stage (Eq. (7a)) to ensure that the coefficients at the leaf elements' neighbours are also up to date prior to applying the second Runge-Kutta stage (Eq. (7b)). This can be achieved by re-encoding the coefficients obtained after the first Runge-Kutta stage. Arguably, running dynamic adaptivity on the GPU comes at a price of storing in memory, at all times, the entire data arrays spanning the hierarchy of grids, which could entail enormous memory load with increase in size of the domain area and/or in the fineness in the DEM resolution.

3. Results and discussion

Dynamic MWDG2 and HWFV1 adaptivity is benchmarked against static DG2 and FV1 adaptivity on the same fixed grid that is also used to start dynamic adaptivity (Section 3.1). The benchmarking is aimed to evaluate how far their simulated results are close to a reference DG2 simulation performed on a uniform grid using the finest DEM resolution, R (uniform DG2). Three field-scale test cases are selected with a uniform Manning roughness parameter, involving initially dry areas. The simulations with dynamic adaptivity used an $\epsilon = 10^{-3}$ to adapt grid resolution over time based on the details of both the flow variables and the topography (Section 2.1). In all the simulations, the maximum resolution level, L , is selected to match the resolution, R , of the DEM. The CPU-based solvers were run on a 3.2GHz Intel i7-8700 using a single CPU

core, while the GPU-based solvers were run on an Nvidia Quadro RTX 4000, for a total simulation time, T_s . The selected test cases, their flow characteristics, and associated simulation parameters are summarised in Table 1.

Table 1

The selected field-scale test cases to benchmark dynamic MWDG2 and HWFV1 adaptivity against static DG2 and FV1 adaptivity.

Test Case	Source	Characteristics	L	R (m)	n_M	T_s (hr)
Carlisle 2005 flooding	Fewtrell et al., 2011; Neal et al., 2009	Gradually propagating flood at city scale from multiple fluvial inflows. Post-event measurements for validation	10	5	0.055	68
Hypothetical flood propagation and inundation in Thamesmead	Liang et al., 2008; Wang and Liang, 2011	Slowly propagating flood from defence failure at district scale	10	10	0.035	10
Flood wave along a valley	Neel and Pender, 2013	Rapidly to gradually propagating flood	11	10	0.04	30

The closeness of the results simulated from the four solvers to the reference uniform DG2 simulation, or measured data, is evaluated using the mean error and root-mean-square error (RMSE). These error metrics are recommended for point data evaluations (Nguyen et al., 2016), and are computed as follows:

$$\text{mean error} = \frac{\sum_{i=1}^n (p - p_{\text{ref}})}{N_s}, \text{RMSE} = \sqrt{\frac{\sum_{i=1}^n [p - p_{\text{ref}}]^2}{N_s}} \quad (11)$$

where p refers to the predicted flow variable (water depth, or level, or velocity), p_{ref} indicates the reference or measured variable, and N_s is the number of data points. For quantitative comparison of flood inundation extent predictions, the hit rate (H), false alarm (F) and critical success index (C) metrics are recommended (Wing et al., 2017; Hoch and Trigg, 2019). H measures how much the reference flood extent is covered by the simulated flood extent, with $H=1$ indicating full coverage and $H=0$ otherwise. F measures how much of the simulated flood extent is outside of the reference flood extent, with $F=0$ indicating that there is no simulated portion outside of the reference flood extent and $F=1$ indicating that all the simulated flood extent is outside the reference flood extent. The C metric combines H and F to weigh how much of the sum of the simulated and the reference flood extents is covered by the simulated flood extent. $C=1$ indicates full coverage and $C=0$ otherwise.

3.1. Selection of the initial grid for the simulations

To perform a fair comparison amongst dynamic MWDG2 and HWFV1 adaptivity and static DG2 and FV1 adaptivity, simulations should start from the same fixed grid that refers to the grid predicted by MWDG2 with grading, as justified in the following. Since all simulations started over dry areas, the *initial* grids predicted by the MRA of HW and MWs, within dynamic HWFV1 and MWDG2 adaptivity, respectively, were selected based on the details of the DEMs for each of the three test cases (Table 1). They are here analysed compared to the fixed grid predicted by the MRA of MWs after grading. Fig. 5 shows the percentage of resolutions selected for grids predicted by the MRA of MWs (with and without grading) and that of HW covering the whole study areas. The grids predicted by the MRA of HW are more refined, by selecting the highest resolution, over most of the domain area in all the test cases (black bars in Fig. 5). In contrast, the grids predicted by the MRA of MWs (without grading) allow for coarser resolutions, only using the finest resolution over 65%, 58% and only 20% over the domain area for the three test cases, respectively (red bars in Fig. 5). The grids predicted by the MRA of MWs (with grading) have finer portions, covering 80% of the domain area with the finest resolution for the first two cases and 37% for the third test case (green bars in Fig. 5). Compared to the predicted grids by the MRA of HW, those predicted with the MRA of MWs (with grading) still allow coarser resolution, providing better allocation of resolutions for the initial grids. Therefore, all the initial grids predicted by dynamic HWFV1 adaptivity are not as efficient as a choice to start the simulations

compared to the fixed grid predicted by the MRA of MWs (with grading) that will be used instead to start dynamic HWFV1 adaptivity.

The grids predicted by the MRA of MWs with and without grading are compared in Fig. 6 for the three test cases considering zoomed-in portions (framed in Figs. 7, 11 and 14 appearing below). Both the non-graded and graded grids are seen able to retain the finest resolutions along sharp topographic features, while allowing coarsening in the areas with smooth topographies. As expected, the non-graded grids have coarser resolutions than the graded grids, which were used hereafter to run the simulations with static DG2 and FV1 adaptivity and to start the simulations with dynamic MWDG2 and HWFV1 adaptivity.

3.2. Carlisle 2005 flooding

This test case is characterised by gradually propagating flood at city scale from multiple fluvial inflows. Because it has post-event measurements for validation, it is investigated first to validate the adaptive and non-uniform grid solvers for real-world flood simulation. The study area is shown in Fig. 7, covering about 14.5 km^2 of the city of Carlisle. The three-day-long flooding is initiated from three inflow hydrographs at the upstream points of Rivers Eden, Petteril and Caldew with a maximum flow rate of $1200 \text{ m}^3 \text{ s}^{-1}$ (shown at the right panels of Fig. 7). Surveyed maximum free-surface elevations are available at 217 wrack marks and 46 water level points (black dots and red crosses in Fig. 7). These will be compared against the maximum free-surface elevations simulated by uniform DG2 and the four solvers (i.e. with dynamic MWDG2 and HWFV1 adaptivity and with static DG2 and FV1 adaptivity). In addition, water depth time series will be recorded at 15 sampling points (blue squares in Fig. 7), including at the three gauges of Sheeplmount, Botcherby Bridge and Denton Holme for which in-situ water depth measurements are available to further support the validation of the solvers.

The four solvers are first verified by comparing their maximum free-surface elevation predictions against the measured data at the water level and wrack mark points. Fig. 8 includes the histogram errors generated by the differences in free-surface elevation between simulated and observed data at the measurement points (x-axis of Fig. 8), and the RMSE and mean error for each solver simulation. The uniform DG2 solver is the most accurate solver by having an RMSE of 0.29 and a mean error of -0.05. Also, its histogram error peak is in the vicinity of 0 with the least skewness compared to the other solvers. This indicates that the DG2 simulations on the uniform grid at the finest resolution R (uniform DG2) can be reliably used as the reference solution in the next two test cases that lack validation data. Dynamic MWDG2 adaptivity exhibits the closest histograms to those of uniform DG2 with RMSE of 0.37 and mean error of -0.09, which compare well with the errors reported in other studies (Neal et al., 2009; Horritt et al., 2010; Fewtrell et al., 2011). Static DG2 adaptivity leads to a slightly higher RMSE error of 0.39 and a mean error of 0.12, and its histogram errors are centred at -0.1, implying an underprediction tendency for the free-surface elevation.

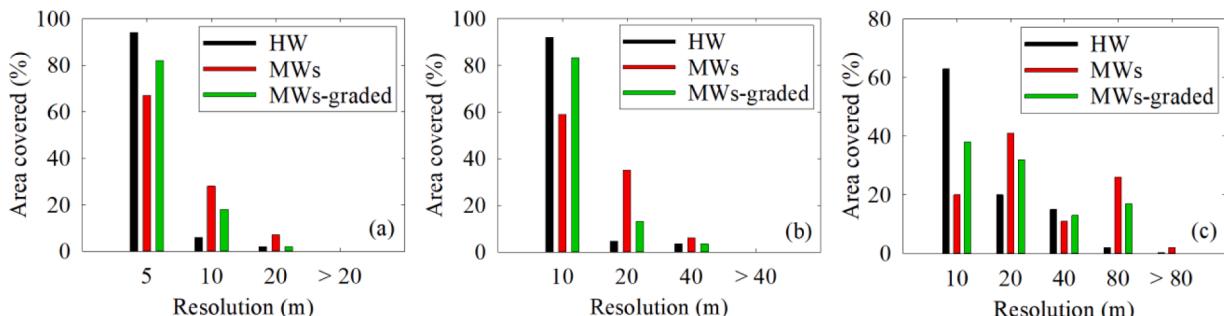


Fig. 5. The percentage of the resolutions selected for grids predicted by the MRA of MWs and HW over the study area along with the fixed grids obtained after grading those predicted by the MRA of MWs: (a) Carlisle 2005 flooding, (b) hypothetical flood in Thamesmead and (c) flood wave along the valley. Note that the number of elements associated with coarser resolution levels, > 20 in (a), > 40 in (b), and > 80 in (c), were negligible (thus not visible).

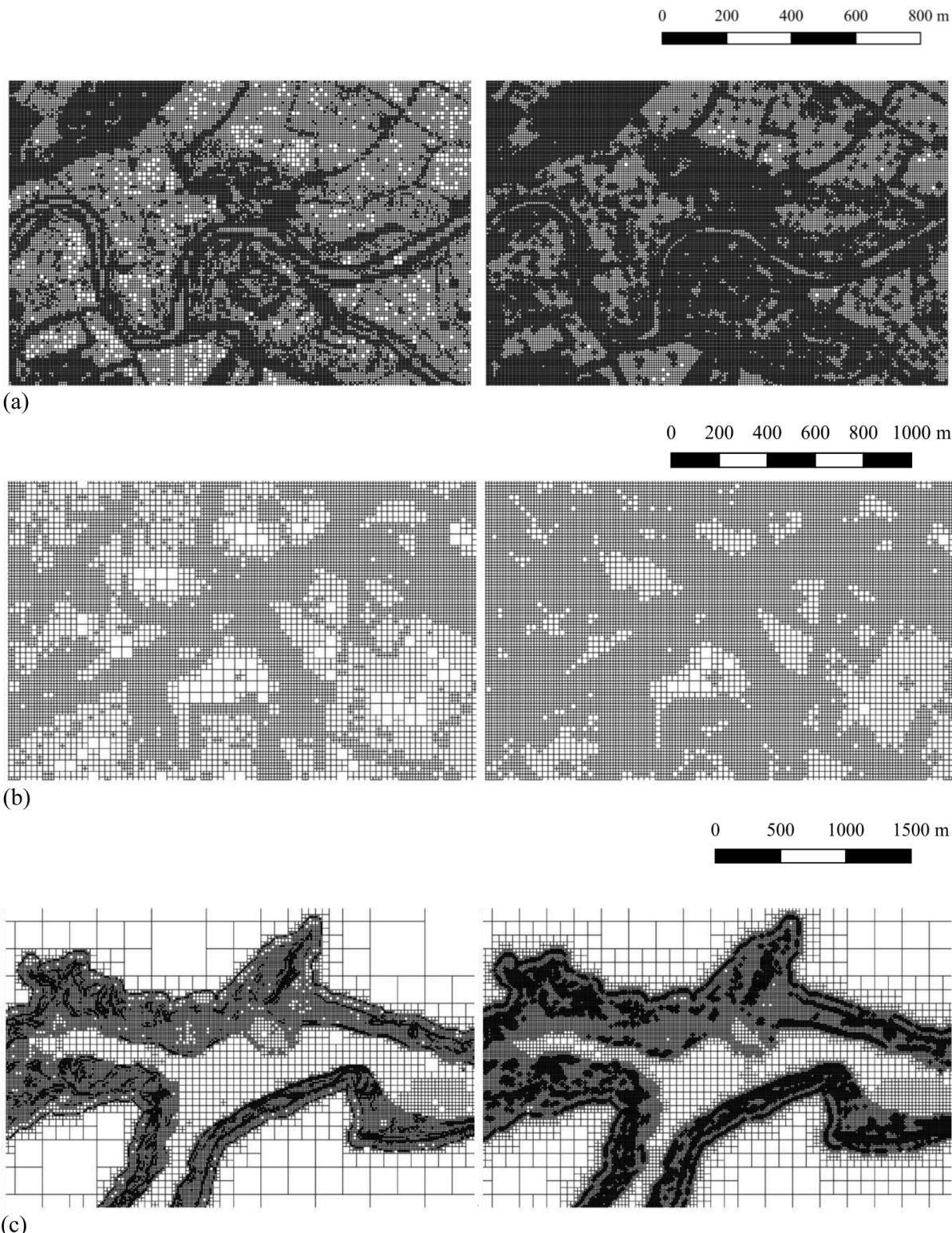


Fig. 6. Initial grids predicted by the MRA of MWs without grading (left) compared to those with grading, or the so-called fixed grid for static adaptivity (right): (a) Carlisle 2005 flooding, (b) hypothetical flood in Thamesmead and (c) flood wave along the valley. The grids are shown on the framed portion of the study areas for each test case (specified later in Figs. 7, 11 and 14, respectively).

Dynamic HWFV1 adaptivity and static FV1 adaptivity resulted in RMSEs of 0.52 and 0.47, and mean errors of -0.43 and -0.32 , respectively. These, together with the fact that their histogram errors are centred around 0.4 and 0.2, suggest that the FV1-based solvers, on their grids involving non-uniform resolutions, tend to significantly overpredict the free-surface elevation compared to the measurements for this test case.

The water level time series recorded at 15 sampling points are compared in Fig. 9. As shown earlier in Fig. 7, these sampling points are distributed over the whole study area, three of which are in the river channels (Sheepmount, Botcherby Bridge and Denton Holme), six in the urban areas (Pallet yard, Substation, Brown Bros, Bus depot, Water mark 1 and Water mark 2), and six others in the rural areas (Building 1, and

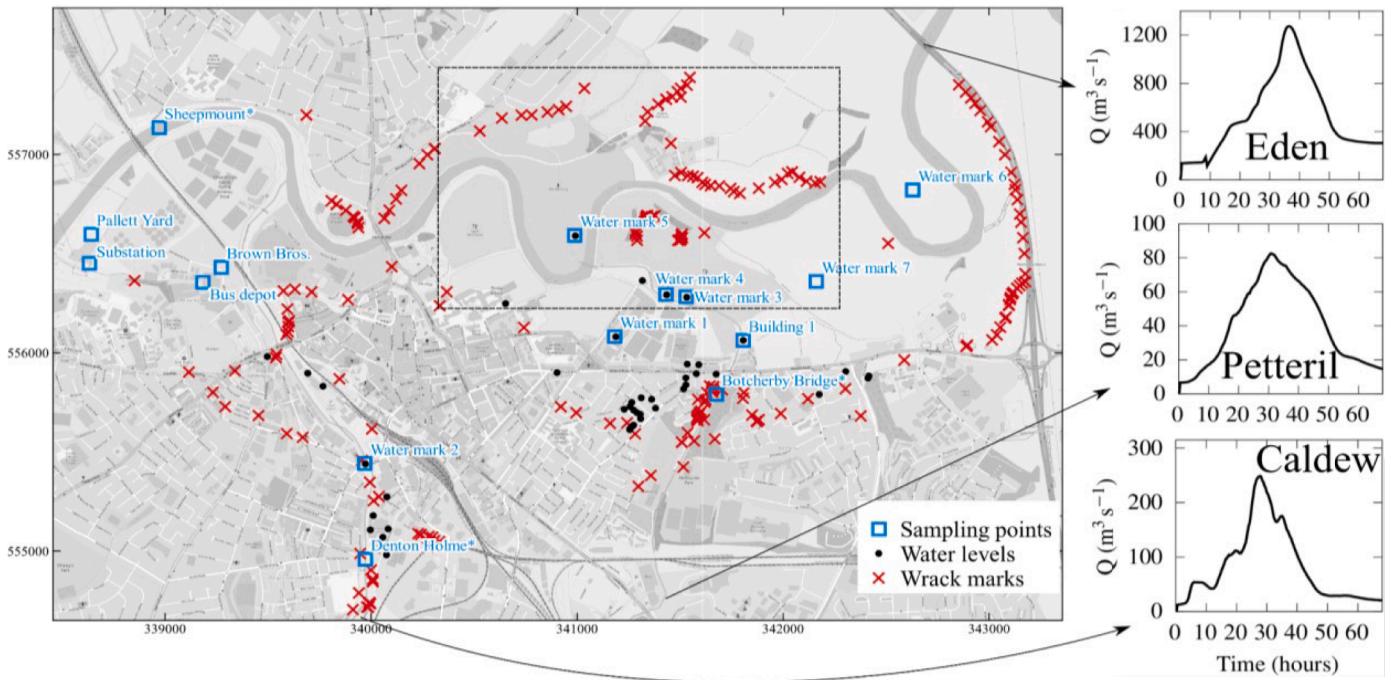


Fig. 7. Carlisle 2005 flooding (Section 3.2): The 14.5 km^2 study area and the positions of the sampling, water level and wrack mark points. In-situ water depth time series are available at three sampling points denoted by asterisks. The arrows show the positions where the inflow hydrographs are imposed at the upstream of Rivers Eden, Petteril and Caldew. The dotted line shows the subset area where the generated grids are compared. Map data ©2020 OpenStreetMap.

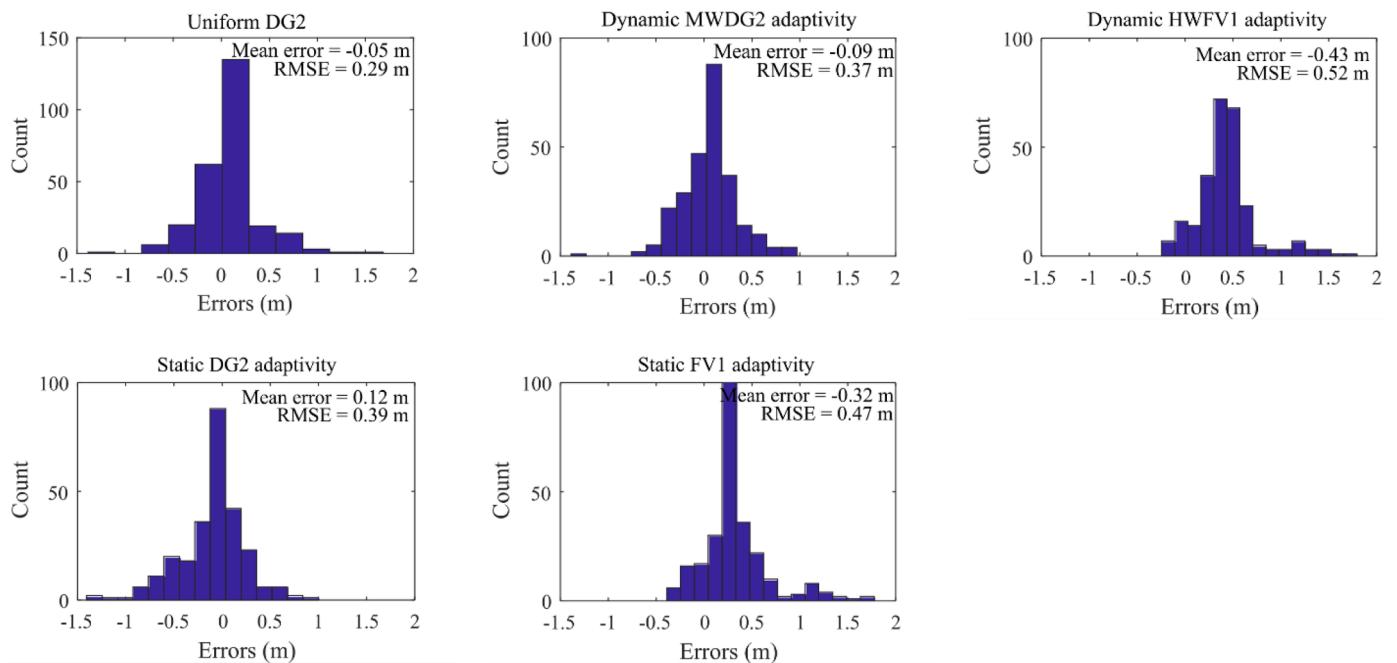


Fig. 8. Carlisle 2005 flooding (Section 3.2). RMSE, mean error and histogram errors calculated from the difference between the simulated and measured maximum free-surface elevations at 217 wrack marks and 46 water level points.

Watermarks 3 to 7). The DG2-based solvers yield water level predictions that have the best agreement with the reference solution, irrespective of the locations of the sampling points. In contrast, the FV1-based solvers significantly overpredict water depths at all the staging points. This difference in the predictions is in line with the analysis of Fig. 8 and alternative studies (Liu and Pender, 2013; Shaw et al., 2021). It mainly arises from the differences in the accuracy-order, amongst the solvers, in the representation of flow and terrain data. The DG2-based solvers use

piecewise-planar representations for the flow variables, the topography and the friction integration, whereas the FV1-based solvers use piecewise-constant representations. Therefore, the predictions from the FV1-based solvers would be more impacted by the growth of numerical diffusion, compared to those from the DG2-based solvers (Ayog et al., 2021; Shaw et al., 2021). This impact is particularly magnified with the FV1-based solvers because they have coarse resolution portions on their grids, at which larger numerical diffusion accumulate, yet affecting the

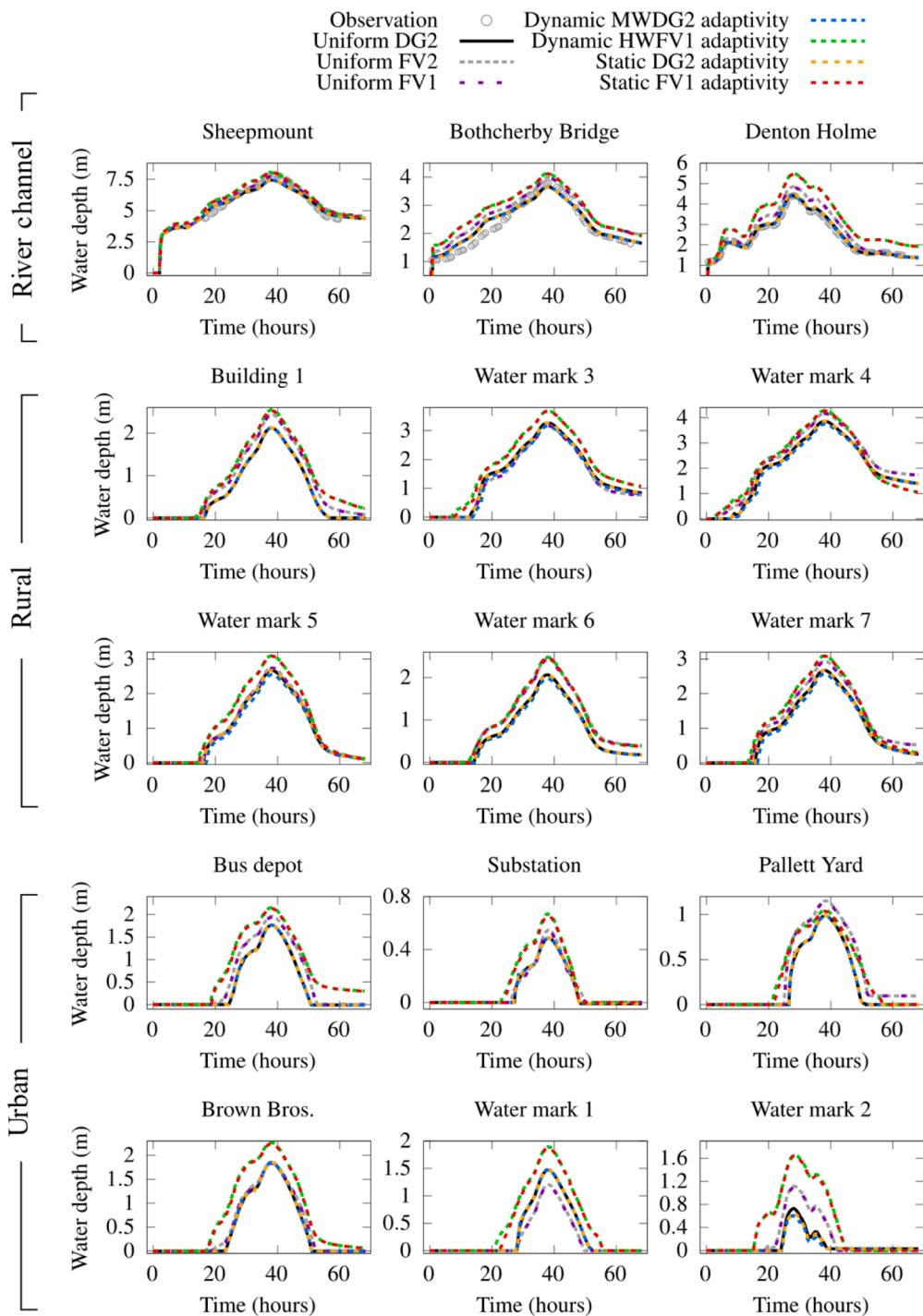


Fig. 9. Carlisle 2005 flooding (Section 3.2). Water level time-series predicted by the solvers with static and dynamic adaptivity compared those predicted by uniform DG2 at 15 sampling points (marked in Fig. 7). The in-situ measured water depths are shown for the first three sampling points.

overall predictions (i.e., for this long-duration simulation over a large and rough area). This observation can also be supported by looking at the predictions by uniform FV1 at the finest resolution R (Fig. 9), which are closer to reference solution than those by FV1-based adaptivity. As in Zhao and Liang (2022), uniform FV2 (Ayog et al., 2021) at the finest resolution R does not improve the uniform FV1 predictions, indicating that resolution coarsening of piecewise-constant representations is the key factor that deteriorated the predictions.

Fig. 10 includes the maximum flood extent predictions by the solvers with dynamic and static adaptivity along with the reference predictions by uniform DG2. The flood extent predicted by the dynamic MWDG2

adaptivity is the closest to the reference extent, with the lowest F of 0.005 and the highest C of 0.98. Static DG2 adaptivity also shows a good agreement with the reference extent, with a C of 0.95, with slight underpredictions in limited areas close to the urban banks of rivers Caldew and Petteril. The FV1-based solvers lead to the widest flood extent compared to the reference extent, most notably over the urban area in the west of river Caldew. This is reflected in their lowest C , of 0.89 resulting from dynamic HWFV1 adaptivity and 0.90 resulting from static FV1 adaptivity.

When modelling water level and flood extent for gradually propagating floods across urban and rural areas over long periods, Static DG2

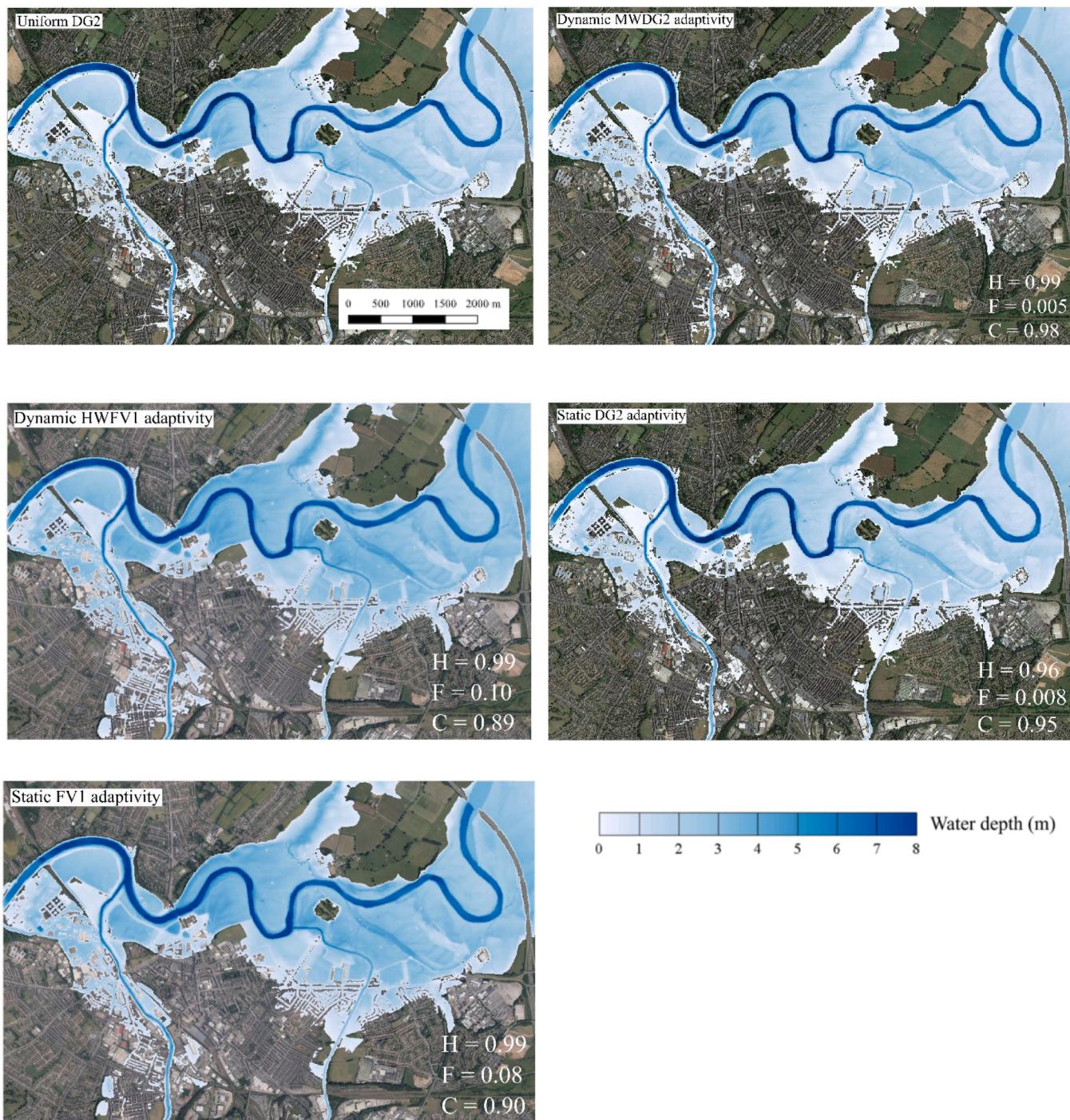


Fig. 10. Carlisle 2005 flooding (Section 3.2). Maximum flood extents predicted by the solvers with static and dynamic adaptivity compared those predicted by uniform DG2, including the hit rate (H), false alarm (F) and critical success index (C) values computed against the reference predictions (uniform DG2). Map data ©2020 Google.

adaptivity seems to lead to accurate enough predictions. These predictions marginally improve with dynamic MWDG2 adaptivity, making it a compelling alternative. amongst the FV1-based solver, static FV1 adaptivity overpredicts the reference predictions with no major improvement seen by instead using HWFV1's dynamic adaptivity. The performance of these solvers is further compared next for test cases with smaller areas and shorter simulation times with a more elaborate analysis that includes velocity predictions over smoother topographies with lower Manning parameters.

3.3. Hypothetical flood propagation and inundation in thamesmead

The test case involves a slowly propagating flood from a hypothetical defence failure over complex terrain in the Thamesmead district. The flood is assumed to occur from a breach along a 150m long defence structure, through which water breaches from river Thames into a 9 km × 4 km floodplain. An aerial view of the floodplain is shown in Fig. 11, along with the location of the inflow breach and four sampling points G1, G2, G3 and G4 where water level and velocity time series are

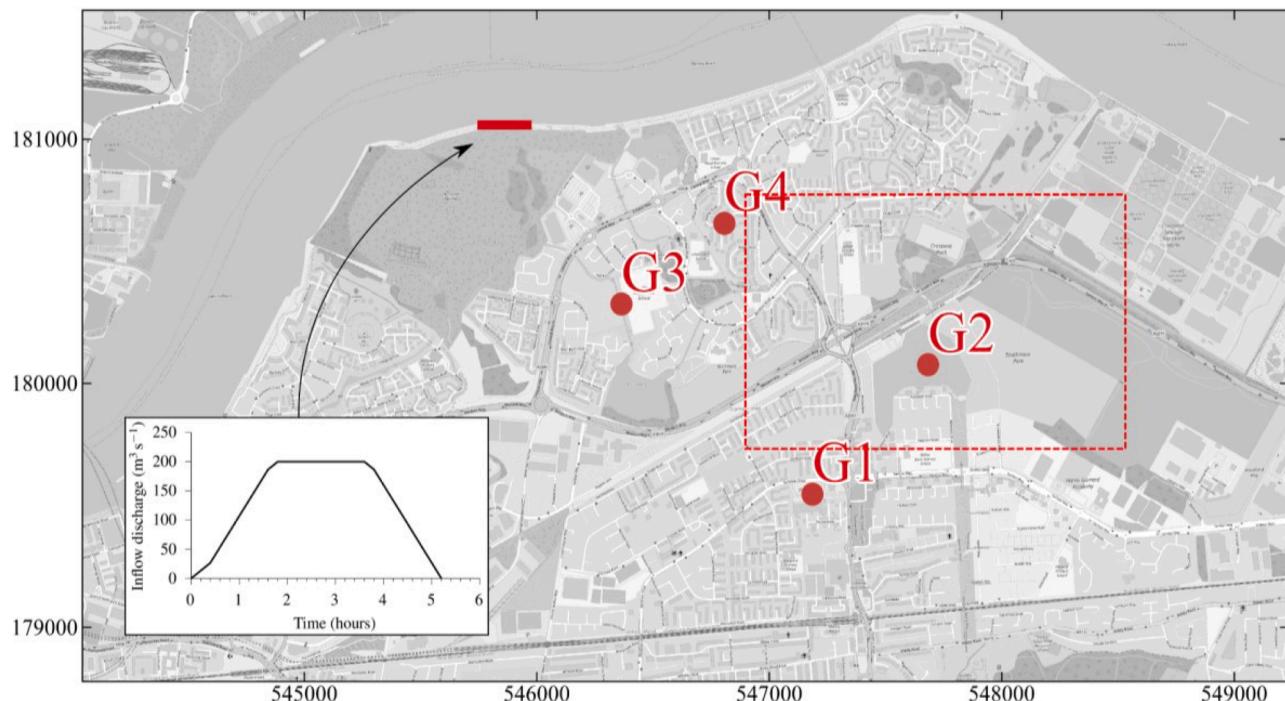


Fig. 11. Hypothetical flood propagation and inundation in Thamesmead (Section 3.3). 9km × 4km study area including the positions of the sampling points and the hypothetical breach in the flood defence structure with the specified inflow hydrograph. The dotted line shows the subset area where the generated grids are compared. Map data ©2020 OpenStreetMap.

recorded. The inflow discharge lasts for 10 hr with a peak of $200 \text{ m}^3 \text{ s}^{-1}$, reached during the second to the fourth hour of flooding (see Fig. 11). The flood wave is expected to pass a narrow channel that cuts through a railway embankment located across most of the domain from southwest to northeast.

Fig. 12 includes the water level and velocity time series recorded at G1 to G4 for the adaptive and non-uniform grid solvers and from the

reference solution. The RMSE values comparing the predicted time series against the reference time series are listed in Table 2. The water levels predicted by both DG2-based solvers are reasonably close to the reference solution, with RMSEs being less than 0.02. Static FV1 adaptivity is also shown to closely predict the reference water level time series, with small deviations leading to RMSEs limited to 0.08. Dynamic HWFV1 adaptivity, however, shows more obvious deviations by

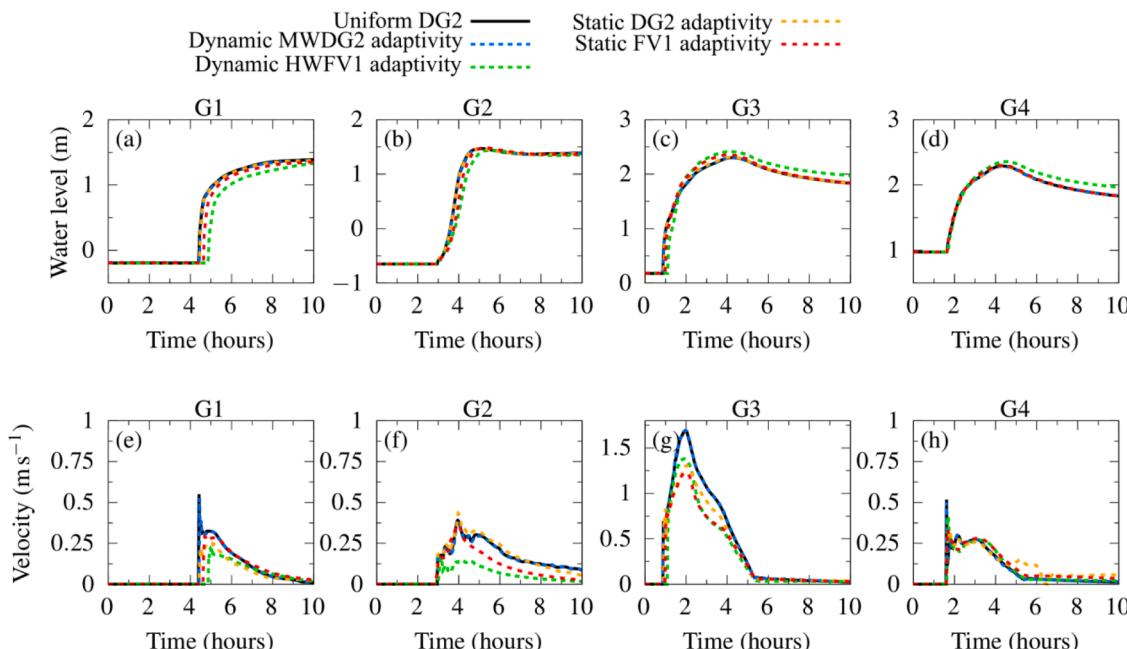


Fig. 12. Hypothetical flood propagation and inundation in Thamesmead (Section 3.3). Water level and velocity time-series predicted the solvers with static and dynamic adaptivity compared those predicted by uniform DG2 at sampling points G3 and G4 (close to the inflow), and points G1 and G2 (away from the inflow). The water levels are referenced to the mean sea level.

Table 2

Hypothetical flood propagation and inundation in Thamesmead (Section 3.3). RMSEs for the predicted water level (m) and velocity ($m s^{-1}$) time-series measured against the predictions made by the uniform DG2 solver.

	Dynamic adaptivity		Static adaptivity		
	MWDG2	HWFV1	DG2	FV1	
Water level	G1	0.013	0.136	0.011	0.076
	G2	0.007	0.090	0.004	0.060
	G3	0.002	0.103	0.017	0.050
	G4	0.002	0.052	0.002	0.008
Velocity	G1	0.017	0.052	0.027	0.033
	G2	0.003	0.063	0.011	0.033
	G3	0.004	0.099	0.078	0.109
	G4	0.002	0.023	0.028	0.022

overpredicting the water levels at points G4 and G3, which are closer to the inflow, while at G2 and G1 where the flow is relatively slower, it leads to delayed arrival times. These deviations are also reflected in higher RMSEs up to 0.14. The different behaviour of dynamic HWFV1 adaptivity compared to static FV1 adaptivity likely occurred from the accumulation of thresholding errors that becomes significant due to deeper traversal across resolution levels on a grid that has more refined portions. As shown in Kesserwani and Sharifian (2020), excess in the refined portions on the grid within the dynamic HWFV1 adaptivity arise from the first-order, piecewise-constant representations, and using a stricter ϵ value will introduce further refinements without adding notable gain in the predictive accuracy.

In terms of the velocity, dynamic MWDG2 adaptivity leads to almost identical predictions to those from the reference uniform DG2 solver with negligible RMSEs limited to 0.02. Static DG2 adaptivity, however,

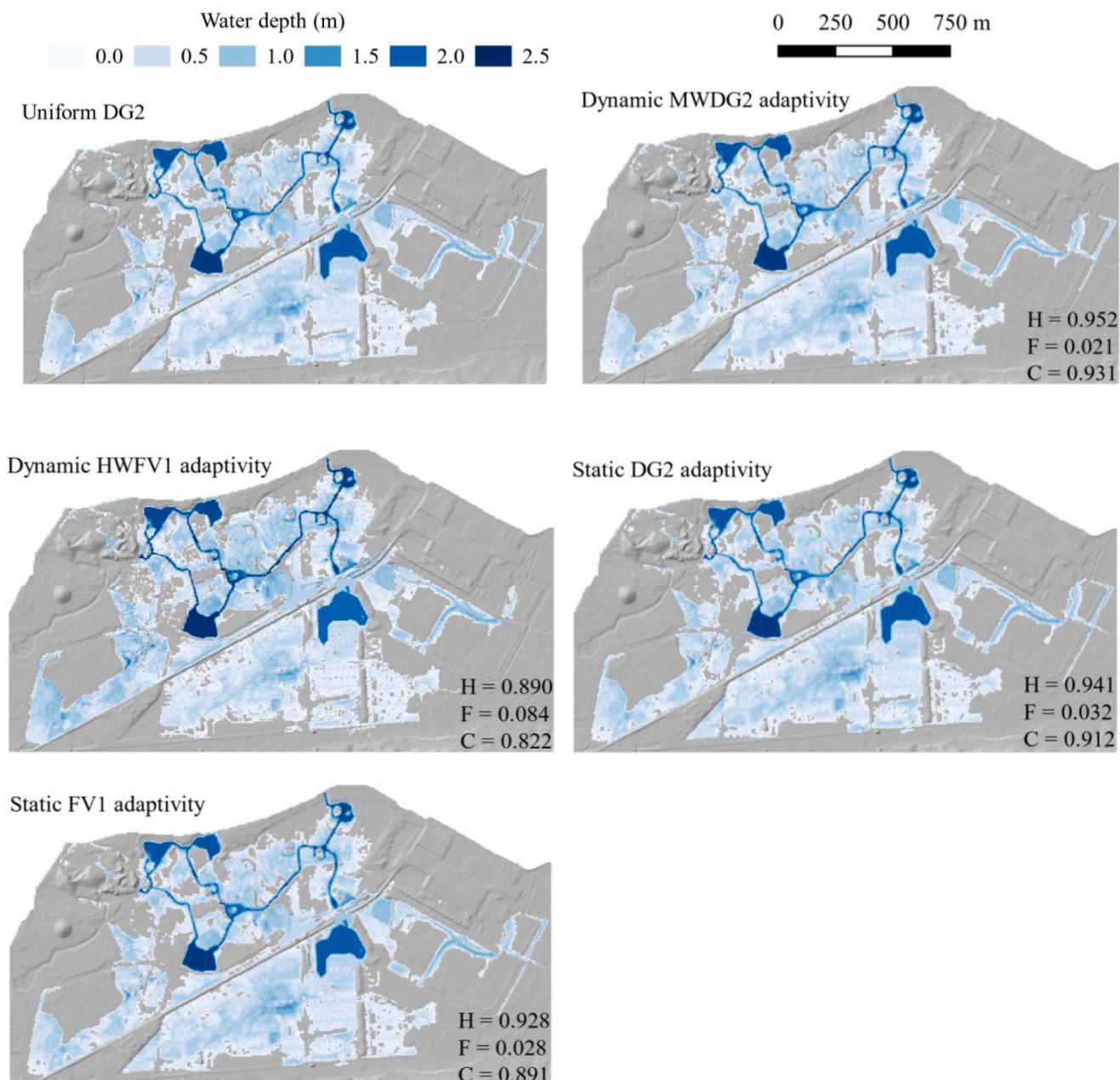


Fig. 13. Hypothetical flood propagation and inundation in Thamesmead (Section 3.3). Final flood inundation maps predicted by the solvers with static and dynamic adaptivity compared those predicted by uniform DG2, including the hit rate (H), false alarm (F) and critical success index (C) values computed against the reference predictions (uniform DG2).

shows small deviations at all four points. These deviations are expected as static DG2 adaptivity could be affected by the errors from the heuristic nature in averaging the slope coefficients (discussed in Section 2.2.1), which are expected to be more present in the velocity predictions as the velocity has a more sensitive variability than the water level. This can also be supported by the fact that the highest deviation is observed at G3, which recorded higher velocities resulting in a higher error, an RMSE of 0.078. Static FV1 adaptivity predicts the reference velocities at G4, with an RSME of 0.022, but underpredicts the velocity at the other three points with RMSEs up to 0.109. Compared to static FV1 adaptivity, dynamic HWFV1 adaptivity performs equally well at points G4 and G3 but shows more pronounced underpredictions at the farther points G1 and G2.

Fig. 13 contrasts the flood inundation maps predicted by the four solvers to the reference map after the 10-hour simulations. Dynamic MWDG2 adaptivity yields the closest predictions to the reference extent with the highest C of 0.93. Static DG2 adaptivity leads to slight underpredictions, at the southern areas of the domain, leading to a close agreement with the reference extent with a C of 0.91. The FV1-based solvers predict the reference extent less accurately with a C of 0.89 achieved by static FV1 adaptivity and a lower C of 0.82 achieved by dynamic HWFV1 adaptivity, suggesting the latter predicts a narrower extent (i.e. in the south and westernmost regions, also confirmed by its lowest H of 0.89).

For this test case, static DG2 adaptivity is able to provide accurate enough predictions for water level and flood extent, but can fall short in predicting velocities, leading to increasingly deviated predictions with higher velocities. Dynamic MWDG2 adaptivity, however, provides the best quality predictions to those of the reference uniform DG2 solver, for the velocities as well as the water levels and flood extent. The opposite is observed for the predictions made by the FV1-based solvers: static FV1 adaptivity predicts flood extents, water level and velocity time series that are closer to the reference predictions than those predicted by dynamic HWFV1 adaptivity. The latter, given its relatively more refined

grid (analysed next in Fig. 16), leads to accumulation of thresholding errors that can impact its predictions over time to make them less accurate than those of static FV1 adaptivity. This suggests favouring static FV1 adaptivity over dynamic HWFV1 adaptivity for modelling slowly propagating floods over a real DEM for a long duration simulation, and dynamic MWDG2 adaptivity over static DG2 adaptivity to acquire the most accurate predictions. The solvers are next compared for a test case involving rapid to gradual flood flow propagation.

3.4. Flood wave along a valley

The performance of the dynamic MWDG2 and HWFV1 adaptivity versus static DG2 and FV1 adaptivity is finally compared for reproducing a scenario that involves a rapidly propagating flow turning to a gradually propagating flow. The flood wave arises from an inflow that enters the upstream of a $17.0 \text{ km} \times 0.8 \text{ km}$ valley and propagates downstream. **Fig. 14** shows the terrain of the valley including 7 sampling points that are typically used to compare water level and velocity time series. As shown in **Fig. 14**, the inflow hydrograph has a high peak discharge of $3000 \text{ m}^3 \text{ s}^{-1}$, lasting for less than 2 hr. During the first 3.5 hr, the flood wave undergoes a rapidly propagating flow stage to then become in a stage of gradually rapidly propagating flow as the water flows downhill, filling in the topographic depressions to reach the larger pond located downstream. The latter stage begins at about 5 hr to last until 30 hr, during which the flow slows down by the friction to approach a steady-state.

Fig. 15 includes the water level and velocity time series recorded for the four solvers with dynamic and static adaptivity at the sampling points and those produced by the reference uniform DG2 solver. Following other studies (cited in Table 1), the time series are analysed for points 1, 3 and 5 located at the upstream, middle, and downstream of the topographic depressions. The RMSE values from the predicted time series with respect to those of the reference solution are included in Table 3.

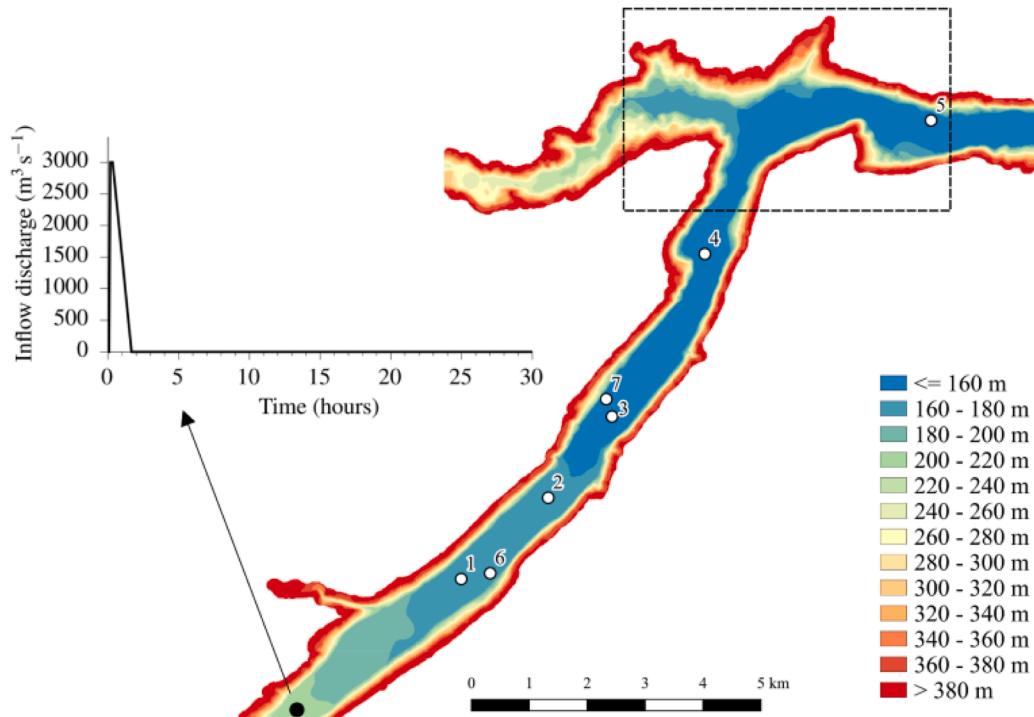


Fig. 14. Flood wave along a valley (Section 3.4). The study areas including the topography of the valley and the positions of the sampling points; along with the inflow hydrograph imposed at the vicinity of the southwestern end.

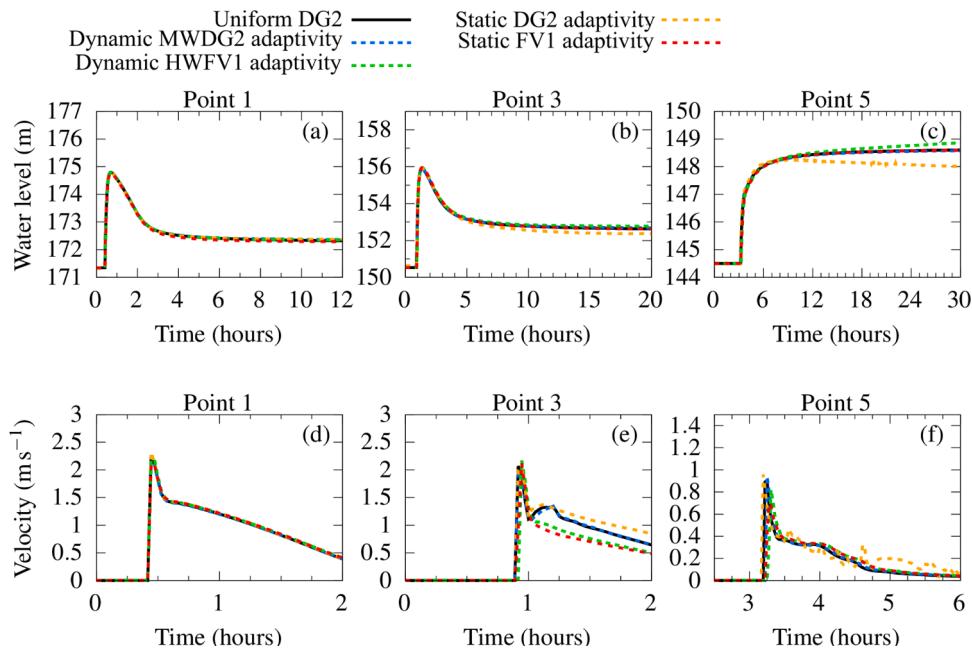


Fig. 15. Flood wave along a valley (Section 3.4). Water level and velocity time-series predicted the solvers with static and dynamic adaptivity compared those predicted by uniform DG2 at sampling points 1, 3 and 5.

Table 3

Flood wave along a valley (Section 3.4). RMSEs for the predicted water level (m) and velocity ($m s^{-1}$) time-series measured against the predictions made by the uniform DG2 solver.

		Dynamic adaptivity		Static adaptivity	
		MWDG2	HWFV1	DG2	FV1
Water level	Point 1	0.037	0.024	0.036	0.052
	Point 3	0.082	0.107	0.295	0.054
	Point 5	0.020	0.142	0.353	0.050
Velocity	Point 1	0.005	0.020	0.013	0.021
	Point 3	0.023	0.286	0.104	0.237
	Point 5	0.061	0.095	0.084	0.072

The velocity time series (Figs. 15d-15f) are recorded for up to 5 hr and are discussed first to analyse the performance of the solvers during the rapidly propagating flow stage. Dynamic MWDG2 adaptivity predicted the reference velocities perfectly, leading to the smallest RMSEs limited to 0.06. This is notably observed at point 3, where the solver with dynamic MWDG2 adaptivity is the only one that perfectly trails the small transient variations, appearing after 1 hr. These velocity variations were detected with the uniform FV2 solver simulation in Ayog et al. (2021) on the finest DEM resolution, but dynamic MWDG2 adaptivity could capture them despite the presence of coarser resolution portions on its grid. Static DG2 adaptivity performs equally well at point 1 (Fig. 15d), where the earliest arrival time is recorded. At point 5 (Fig. 15e), it is able to partly capture the small transient variations, but overly predicts the reference velocity leading to an RMSE of 0.104. At Point 5 (Fig. 15f), where the flood is driven by the topography and friction effects, static DG2 adaptivity predicts comparable velocities up to 3.5 hr, but its predictions are affected by numerical disturbances afterwards. In line with the past observations, made for Fig. 12, these spurious disturbances can be attributed to the heuristic nature of averaging slope coefficients (Section 2.2.1). The FV1-based solvers lead to good predictions at points 1 and 5 in good agreement with the reference velocities. Both fail to capture the small-scale velocity variations at point 3, leading to RMSEs up to 0.28, but dynamic HWFV1 adaptivity shows a

better performance in capturing velocity peaks, suggesting that it can still be a good choice for short duration simulations of highly dynamic flows.

Water level time series are compared in Figs. 15a-15c to analyse the performance of the solvers during the gradually propagating flow stage. Again, dynamic MWDG2 adaptivity leads to the best agreement with the reference water levels at all three points with RMSEs less than 0.08. Static DG2 adaptivity only closely predicts the reference water level at point 1 but tends to underpredict it at point 3, leading to the highest RMSE of 0.295. This underprediction is intensified at point 5, which is likely due to the impact of the spurious numerical disturbances arising in the velocity predictions. The FV1-based solvers predict closely the reference water levels at point 1, but dynamic HWFV1 adaptivity shows an overprediction tendency at points 3 and 5 (by 0.2m). This tendency seems to only occur in the long run after the flow stagnates to steady-state, where the accumulation of thresholding errors from the deeper traversals across resolution levels becomes irrelevant.

When modelling flows with rapid to gradual transitions, this test case suggests that static DG2 adaptivity is not the best choice. As observed in Section 3.3, its velocity predictions can be disturbed by averaging errors that tend to grow when the flow decelerates, causing underpredicted water levels. Dynamic MWDG2 adaptivity comes in as a viable alternative to preserve the highest quality possible of the reference predictions no matter if the flow is gradual or rapid. Static FV1 adaptivity, although not as good on velocity predictions, is still a good option to get acceptable water level predictions during the gradually propagating flow stage. Dynamic HWFV1 adaptivity could only perform better over the short period of rapidly propagating flow; in the long run, it can be affected by the accumulation of thresholding errors, inflicting less accurate water levels, thus may not be a better alternative to static FV1 adaptivity for long duration simulation of decelerating floods.

4. Efficiency analysis

The efficiency of the solvers is analysed in terms of reduction in the number of elements over that on the uniform grid at the finest resolution R , speedups from the runtime achieved on a single CPU core to identify the difference between static and dynamic adaptivity costs, and the

extent of runtime cost reduction with dynamic adaptivity for the GPU implementation. The percentages of reduction in the number of elements are evaluated using Eq. (12).

$$\text{Reduction of elements (\%)} = \frac{\text{Elements No. on the uniform grid} - \text{Elements No. on the adaptive grid}}{\text{Elements No. on the uniform grid}} \times 100 \quad (12)$$

The reductions are first evaluated for the fixed grid used with static DG2 and FV1 adaptivity and then for the 2D adaptive grids predicted by dynamic MWDG2 and HWFV1 adaptivity during the simulations. These evaluations are shown in Fig. 16 for the three simulated scenarios: “Carlisle 2005 flooding”, the “Hypothetical flood in Thamesmead” and the “Flood wave along a valley” (Section 3.2-3.4). In the first two scenarios, the DEMs are more complex, including urban features, than that of the third scenario. Because of this, less reduction in the number of elements is achieved on the fixed grid in the first two scenarios, 13 % and 9 %, respectively, compared to the third scenario where the reduction is around 47 %. This results in lower speedups with DG2 and FV1 static adaptivity with reference to the uniform DG2 solver, informed by the analysis of CPU runtimes listed in Table 4. Namely, for the scenario with 9 % reduction, speedup ratios of 5.4 and 1.7 were achieved, respectively, which increased to 7.5 and 2.8 for the scenario with 13 % reduction and to 25.7 and 6.5 for the scenario with 47 % reduction. The enhancement in the speedups by moving from static DG2 and FV1 adaptivity to dynamic MWDG2 and HWFV1 adaptivity are found to be 0.9 and 2.6 for the scenario with 9 % reduction, 1.8 and 1.9 for the scenario with 13 % reduction, 2.7 and 5.7 for the scenario with 47 %

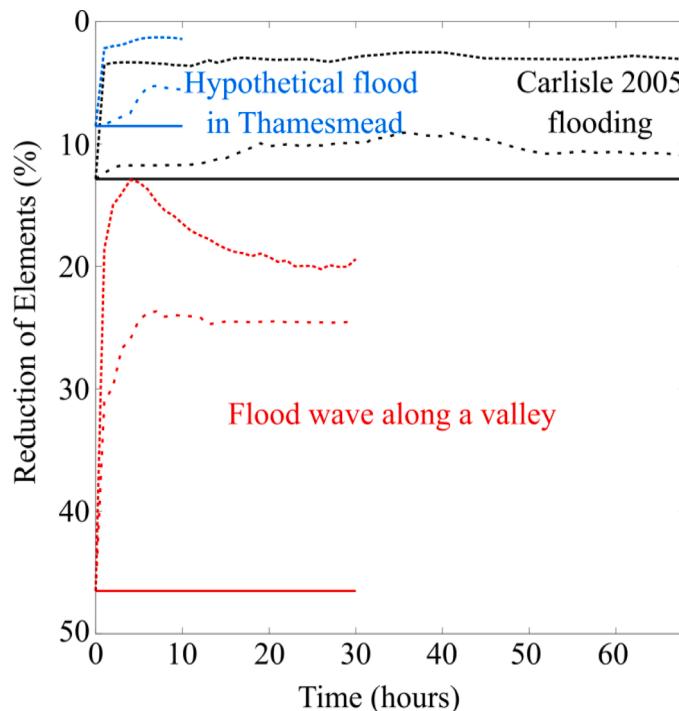


Fig. 16. Reduction in the number of grid elements by dynamic MWDG2 adaptivity (long dashed lines), dynamic HWFV1 adaptivity (short dashed lines) and the fixed grids (solid lines) used with static DG2 and FV1 adaptivity with respect to the uniform grids using the finest resolution. These are shown for the three test cases of “Carlisle 2005 flooding” (black), “Hypothetical flood propagation and inundation in Thamesmead” (blue) and “Flood wave along a valley” (red).

reduction, respectively. This speedup analysis indicates that the larger the reduction in elements the more the speedup gain with static adaptivity and more enhancement in speedup would be gained by moving to

dynamic adaptivity. As dynamic adaptivity started from the fixed grids (used with static adaptivity), this analysis implies that the key parameter to look at when weighing the expected, or baseline, speedup is the reduction in elements with respect to the finest uniform grid. The higher the reduction, the more the merit from deploying dynamic adaptivity to reduce the runtime cost. Therefore, making the choice of when to use static or dynamic adaptivity depends on the properties of the test, first in relation to the features of the DEM and then to the nature of the flood flow. This is discussed next in particular for dynamic MWDG2 adaptivity to identify when it would be a fast enough alternative to uniform DG2 in spite of running the GPU implementation.

Over time, the number of elements on the dynamic HWFV1 and MWDG2 adaptivity increases leading to an expected reduction in their efficiency. Dynamic HWFV1 adaptivity uses a much higher number of elements than dynamic MWDG2 adaptivity in all the scenarios. In the first two, it leads to grids that have less than 5 % reduction in elements over the fine uniform grid, confirming its impracticality for simulating slow to gradually propagating floods over a long duration. In the third scenario, HWFV1’s grid reaches 13 to 20 % reduction in elements. In all the scenarios, dynamic MWDG2 adaptivity offers a greater reduction in elements while leading to more accurate predictions (Sections 2.3 and 2.4) while remaining faster to run than the uniform DG2 simulations (Table 4). Dynamic MWDG2 adaptivity’s speedup ratios for the CPU simulations are around 1.5 for the first two scenarios, and higher, around 2.3, for the third one that has more reduction in the number of elements on the initial grid. Nonetheless, the costs for the MWDG2 runs on a single CPU core are unfeasibly large for practical simulations over largely dry areas, hindered by its need to regenerate the 2D adaptive grid in serial every time step. With the proposed GPU implementation for dynamic MWDG2 and HWFV1 adaptivity, their runtimes are reduced significantly and remain smaller than the runtimes required by uniform DG2 simulation on the GPU (Table 4). However, dynamic HWFV1 adaptivity leads to predictions that are somewhat less competitive than those made by static FV1 adaptivity (Sections 2.3 and 2.4) and is not as valid as an option compared to dynamic MWDG2 adaptivity. Compared to the single core CPU runs, MWDG2 on the GPU achieved gains in speedup of around 41, 30, and 53.5 times for the three scenarios, respectively. Dynamic MWDG2 adaptivity also leads to runtimes that are much less than the clock time and the runtimes of uniform DG2 on the GPU, namely achieving speedups of 1.05, 1.75 and 3.67 respectively. Less speedups are expected for the first two scenarios given the relatively less reduction on their initial grids compared to the third scenario. Consequently, using dynamic MWDG2 adaptivity on the GPU can still be more efficient than uniform DG2 on the GPU for flood simulations; however, the extent of its efficiency gain primarily depends on how much reduction in the number of elements is achieved on the initial grid with respect to the finest uniform grid of the DG2 solver, and secondarily on the dynamic transients involved in the flood in question. The more the reduction and the stronger the flow’s dynamics, the more the efficiency gain from using dynamic MWDG2 adaptivity.

5. Summary, conclusions and recommendations

Real-world flood simulators often use first-order finite volume (FV1) solvers of the shallow water equations with efficiency enhancements to

Table 4

The runtime cost (in hours) of running the three test cases with respective solvers on CPU/GPU.

	Static adaptivity		Dynamic adaptivity				Uniform DG2	
	FV1	DG2	HWFV1		MWDG2		Uniform DG2	
	CPU	CPU	CPU	GPU	CPU	GPU	CPU	GPU
Carlisle 2005 flooding	136	362	265	1.7	682	16.7	1024	17.5
Hypothetical flood in Thamesmead	2.8	8.7	7.4	0.04	9.6	0.32	15.1	0.56
Flood wave along a valley	5.6	22	32	0.26	61	1.14	144	4.19

include parallelisation on Graphical Processing Units (GPUs) and static adaptivity on a fixed grid. A second-order discontinuous Galerkin (DG2) solver greatly increases the accuracy in the predictions but is costly to run on uniform grids and its potential for use as an alternative to finite volume based solvers with static adaptivity was newly investigated. Alternative solvers with (multi)wavelet-based dynamic adaptivity have been developed that adapt and scale local DG2 and FV1 numerical solutions, while applying the multiresolution analysis (MRA), to select the 2D adaptive grid over time. The MRA of Multiwavelets (MWs) with DG2 piecewise-planar solutions leads to dynamic MWDG2 adaptivity, and that of the Haar wavelet (HW) with FV1 piecewise-constant solutions to dynamic HWFV1 adaptivity. The performance of dynamic adaptivity was also newly explored for practical real-world simulations, to find out when they yield more accuracy than static adaptivity, and how far the runtimes of dynamic adaptivity become affordable when the MRA of MWs and HW is re-worked to run on the GPU.

Therefore, the predictive capabilities of dynamic MWDG2 and HWFV1 adaptivity were compared to static DG2 and FV1 adaptivity for complex, real-world flooding scenarios. The static DG2 and FV1 adaptivity used a fixed grid achieved by applying the MRA of MWs to the (time-invariant) features of digital elevation model (DEM). The fixed grid had to be graded to be able to adapt the DG2 and FV1 flow solutions across the interfaces with adjacent, heterogeneously-sized grid elements. The graded, fixed grid was also the initial grid used to run dynamic adaptivity for a fair assessment of performance. Dynamic MWDG2 and HWFV1 adaptivity and static DG2 and FV1 adaptivity were applied to reproduce three field-scale flood simulation case studies, involving real topographies and different flows from slow to rapidly propagating. Their predictive accuracy was evaluated by measuring the closeness of their predictions to the reference predictions made by uniform grid DG2 solver simulations using the finest DEM resolution (uniform DG2). The evaluations included qualitative and quantitative analysis of maximum water levels, water levels/velocities time-series, spatial flood maps, and an efficiency analysis considering reduction in the number of elements during a simulation, and runtime costs on a single CPU core to complete the simulation, and the speedup gained with dynamic MWDG2 and HWFV1 adaptivity by using the GPU parallelised versions.

The evaluations show that static adaptivity is a better choice than dynamic adaptivity to simulate slow to gradually propagating flood flows. For such flows, the static adaptivity yields closer predictions to the reference uniform DG2 predictions and is faster to run than dynamic adaptivity that is hindered by the overhead costs of the MRA. Amongst static DG2 and FV1 adaptivity, that of DG2 seems to be the best choice to capture small-scale time variations in the velocities and more accurate flood maps. Static FV1 adaptivity, though not as accurate as static DG2 adaptivity, is still a valid alternative to acquire similar water level predictions at reduced runtimes. As for dynamic MWDG2 and HWFV1 adaptivity, it remains less expensive to run than the reference uniform DG2 solver on a single CPU core but gain major speedups on the GPU, where their simulation times become less than the clock times. The piecewise-constant representations within dynamic HWFV1 adaptivity yields to more refinement on the 2D adaptive grid, causing accumulation of thresholding errors from the MRA of HW across a deeper tree traversal. In the long run, this manifests in overly predicted water levels,

making dynamic HWFV1 adaptivity, or even a second-order finite volume variant that uses the MRA of HW with piecewise-constant representations, not ideal for long duration simulations with slowly propagating flows.

Dynamic MWDG2 adaptivity yields the best quality predictions, for both water levels and velocities, and its 2D adaptive grid involves more sensible resolution coarsening as the flow remains in a state of gradual to slow propagation. For a rapidly propagating flow, dynamic MWDG2 adaptivity seems to excel in closely reproducing the predictive accuracy of the reference, uniform DG2 predictions where any other choice for the adaptivity underperform: static DG2 adaptivity became affected by spurious disturbances whereas static FV1 adaptivity could not closely replicate the reference velocity predictions. In terms of speedup over the reference uniform DG2 simulations on the GPU, dynamic MWDG2 adaptivity always remained faster but the speedup rate of dynamic adaptivity is mainly dependant on the reduction in the number of elements on the fixed grid relative to the finest uniform grid. This suggests that GPU parallelisation with dynamic MWDG2 adaptivity is useful to maximise accuracy and efficiency when simulating flood events driven by a fast flow propagation (e.g. tsunamis or dam-breaks) for uniform resolution DEMs leading to at least 15 % reduction in the number of elements on its initial, fixed grids.

CRediT authorship contribution statement

Georges Kesserwani: Conceptualization, Methodology, Formal analysis, Supervision, Project administration, Funding acquisition, Writing – original draft, Writing – review & editing. **Mohammad Kazem Sharifian:** Conceptualization, Methodology, Software, Visualization, Investigation, Data curation, Writing – review & editing.

Declaration of Competing Interest

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

Data availability

The data is already openly available on: 10.5281/zenodo.5047565 or in Sharifian and Kesserwani (2021).

Acknowledgements, data and software accessibility

We wish to thank Paul Bates (University of Bristol) for sharing the validation data for the Carlisle 2005 test case and Mr. Alovya A Chowdhury (University of Sheffield) for the technical support to setup and run dynamic MWDG2 and HWFV1 adaptivity on the GPU. This work was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/R007349/1. It is part of the SEAMLESS-WAVE project (Software infrAstructure for Multi-purpose flood modElling at various scaleS based on WAVElets, <https://www.seamlesswave.com>). The simulation results, reported in Section 3, are available on Zenodo (Sharifian and Kesserwani, 2021), and the DEMs

can be accessed by contacting the UK Environment Agency. The CPU codes for static DG2 and FV1 adaptivity can be shared upon request, and those of dynamic MWDG2 and HWFV1 adaptivity are available on Zenodo (Sharifian and Kesserwani, 2020). The GPU implementation of the fixed grid generator and of dynamic MWDG2 and HWFV1 adaptivity have been incorporated into the open-source LISFLOOD-FP8.1 hydraulic modelling packages and will be available in the next version release (10.5281/zenodo.4073011). For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

References

- Ayog, J.L., Kesserwani, G., Shaw, J., Sharifian, M.K., Bau, D., 2021. Second-order discontinuous Galerkin flood model: comparison with industry-standard finite volume models. *J. Hydrol. (Amst)* 594, 125924.
- Bader, M., 2013. Space-Filling Curves: An Introduction With Applications in Scientific Computing, 1st ed. Springer, Berlin, Heidelberg.
- Baert, J., Lagae, A., Dutré, P., 2013. Out-of-core construction of sparse voxel octrees. In: Proceedings of the 5th high-performance graphics conference, pp. 27–32.
- BMT. (2018). TUFLOW Classic/HPC User Manual, Build 2018-03-AD. WBM Oceanics Australia.
- Bellou, V., Papageorgaki, I., Kourtis, I., Vangelis, H., Kalogiros, I., Tsakiris, G., 2020. Reconstruction of a flash flood event using a 2D hydrodynamic model under spatial and temporal variability of storm. *Adv. Clim. Changes, Global Warming, Biol. Probl. Nat. Hazards, 3rd WSEAS Int. Conf. Clim. Changes, Global Warming, Biol. Probl. (CGB '10) 3rd WSEAS Int. Conf. Nat. Hazards (NAHA '10) 101 (3), 711–726.*
- Billeter, M., Olsson, O., Assarsson, U., 2009. Efficient stream compaction on wide SIMD many-core architectures. In: Proceedings of the conference on high performance graphics 2009, pp. 159–166.
- Borthwick, A.G.L., Marchant, R.D., Copeland, G.J.M., 2000. Adaptive hierarchical grid model of water-borne pollutant dispersion. *Adv. Water Resour.* 23 (8), 849–865.
- Chowdhury, A., Kesserwani, G., Rougé, & Richmond, P. (2022). Wavelet-based adaptive mesh refinement on the GPU for shallow water modelling. Preprint <https://arxiv.org/abs/2206.05761> submitted to the *Journal of Hydroinformatics*, in review.
- Caviedes-Voullième, D., Gerhard, N., Sikstel, A., Müller, S., 2020. Multiwavelet-based mesh adaptivity with Discontinuous Galerkin schemes: exploring 2D shallow water problems. *Adv. Water Resour.* 138 (March).
- Caviedes-Voullième, D., Kesserwani, G., 2015. Benchmarking a multiresolution discontinuous Galerkin shallow water model: implications for computational hydraulics. *Adv. Water Resour.* 86, 14–31.
- Cockburn, B., Shu, C.-W., 2001. Runge–Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Comput.* 16 (3), 173–261.
- Donat, R., Martí, M.C., Martínez-Gavarra, A., Mulet, P., 2014. Well-Balanced Adaptive Mesh Refinement for shallow water flows. *J. Comput. Phys.* 257, 937–953.
- Dunning, D., Marts, W., Robey, R.W., Bridges, P., 2020. Adaptive mesh refinement in the fast lane. *J. Comput. Phys.* 406, 109193.
- Echeverriabar, I., Morales-Hernández, M., Brufau, P., García-Navarro, P., 2019. 2D numerical simulation of unsteady flows for large scale floods prediction in real time. *Adv. Water Resour.* 134 (June), 103444.
- Fewtrell, T.J., Neal, J.C., Bates, P.D., Harrison, P.J., 2011. Geometric and structural river channel complexity and the prediction of urban inundation. *Hydrol. Process.* 25 (20), 3173–3186.
- Gerhard, N., Caviedes-Voullième, D., Müller, S., Kesserwani, G., 2015. Multiwavelet-based grid adaptation with discontinuous Galerkin schemes for shallow water equations. *J. Comput. Phys.* 301, 265–288.
- Ghazizadeh, M.A., Mohammadi, A., Kurganova, A., 2020. An adaptive well-balanced positivity preserving central-upwind scheme on quadtree grids for shallow water equations. *Comput. Fluids* 208, 104633.
- Guo, K., Guan, M., Yu, D., 2020. Urban surface water flood modelling – a comprehensive review of current models and future challenges. *Hydrol. Earth Syst. Sci. Discuss.* 1–27.
- Haleem, D.A., Kesserwani, G., Caviedes-Voullième, D., 2015. Haar wavelet-based adaptive finite volume shallow water solver. *J. Hydroinf.* 17 (6), 857–873.
- Hoch, J.M., Trigg, M.A., 2019. Advancing global flood hazard simulations by improving comparability, benchmarking, and integration of global flood models. *Environ. Res. Lett.* 14 (3).
- Horritt, M.S., Bates, P.D., Fewtrell, T.J., Mason, D.C., Wilson, M.D., 2010. Modelling the hydraulics of the carlisle 2005 flood event. *Proc. Inst. Civ. Eng. Water Manage.* 163 (6), 273–281.
- Hou, J., Wang, R., Liang, Q., Li, Z., Huang, M.S., Hinkelmann, R., 2018. Efficient surface water flow simulation on static Cartesian grid with local refinement according to key topographic features. *Comput. Fluids* 176, 117–134.
- Hunter, N.M., Bates, P.D., Neelz, S., Pender, G., Villanueva, I., Wright, N.G., Liang, D., Falconer, R.A., Lin, B., Waller, S., Crossley, A.J., Mason, D.C., 2008. Benchmarking 2D hydraulic models for urban flooding. *Proc. Inst. Civ. Eng. Water Manage.* 161 (1), 13–30.
- Karras, T. (2012). Thinking parallel, part ii: tree traversals on the GPU. <https://devblogs.nvidia.com/parallelforall/thinkingparallel-part-ii-tree-traversals-gpu/>.
- Kesserwani, G., Ayog, J.L., Bau, D., 2018. Discontinuous Galerkin formulation for 2D hydrodynamic modelling: trade-offs between theoretical complexity and practical convenience. *Comput. Methods Appl. Mech. Eng.* 342, 710–741.
- Kesserwani, G., Ghostine, R., Vazquez, J., Ghénaïm, A., Mosé, R., 2008. Riemann solvers with Runge–Kutta discontinuous Galerkin schemes for the 1D shallow water equations. *J. Hydraul. Eng.* 134 (2), 243–255.
- Kesserwani, G., Liang, Q., 2012a. Dynamically adaptive grid based discontinuous Galerkin shallow water model. *Adv. Water Resour.* 37, 23–39.
- Kesserwani, G., Liang, Q., 2012b. Locally limited and fully conserved RKDG2 shallow water solutions with wetting and drying. *J. Sci. Comput.* 50 (1), 120–144.
- Kesserwani, G., Caviedes-Voullième, D., Gerhard, N., Müller, S., 2015. Multiwavelet discontinuous Galerkin h-adaptive shallow water model. *Comput. Methods Appl. Mech. Eng.* 294, 56–71.
- Kesserwani, G., Sharifian, M.K., 2020. (Multi)wavelets increase both accuracy and efficiency of standard Godunov-type hydrodynamic models: robust 2D approaches. *Adv. Water Resour.* 144.
- Kesserwani, G., Shaw, J., Sharifian, M.K., Bau, D., Keylock, C.J., Bates, P.D., Ryan, J.K., 2019. (Multi)wavelets increase both accuracy and efficiency of standard Godunov-type hydrodynamic models. *Adv. Water Resour.* 129.
- Liang, Q., Du, G., Hall, J.W., Borthwick, A.G., 2008. Flood Inundation Modeling with an Adaptive Quadtree Grid Shallow Water Equation Solver. *J. Hydraul. Eng.* 134 (11), 1603–1610.
- 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.
- Liu, Y., Pender, G., 2013. Carlisle 2005 urban flood event simulation using cellular automata-based rapid flood spreading model. *Soft Comput.* 17 (1), 29–37.
- Neal, J.C., Bates, P.D., Fewtrell, T.J., Hunter, N.M., Wilson, M.D., Horritt, M.S., 2009. Distributed whole city water level measurements from the Carlisle 2005 urban flood event and comparison with hydraulic model simulations. *J. Hydrol. (Amst)* 368 (1–4), 42–55.
- Néelz, S., Pender, G., 2013. Benchmarking the Latest Generation of 2D Hydraulic Modelling Packages. Environment Agency, Bristol, UK.
- Nguyen, P., Thorstensen, A., Sorooshian, S., Hsu, K., AghaKouchak, A., Sanders, B., Koren, V., Cui, Z., Smith, M., 2016. A high resolution coupled hydrologic-hydraulic model (HiResFlood-UCI) for flash flood modeling. *J. Hydrol. (Amst)* 541, 401–420.
- Nkwunonwo, U.C., Whitworth, M., Baily, B., 2020. A review of the current status of flood modelling for urban flood risk management in the developing countries. *Scient. Africain* 7, e00269.
- NVIDIA (2021). CUDA C++ Programming Guide.
- Özgen-Xian, I., Kesserwani, G., Caviedes-Voullième, D., Molins, S., Xu, Z., Dwivedi, D., David Moulton, J., Steefel, C.I., 2020. Wavelet-based local mesh refinement for rainfall-runoff simulations. *J. Hydroinf.* 22 (5), 1059–1077.
- Popinet, S., 2003. Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *J. Comput. Phys.* 190 (2), 572–600.
- Saksena, S., Dey, S., Merwade, V., Singhofen, P.J., 2020. A Computationally Efficient and Physically Based Approach for Urban Flood Modeling Using a Flexible Spatiotemporal Structure. *Water Resour. Res.* 56 (1), 1–22.
- Sagan, H., 1994. Space-Filling Curves, 1st ed. Springer New York, NY, New York.
- Sharifian, M.K., Kesserwani, G., 2020. Two-dimensional Fortran 2003 finite volume and discontinuous Galerkin hydrodynamic models with (multi)wavelet-based adaptivity (Version 1). Zenodo. <https://doi.org/10.5281/zenodo.3941723>.
- Sharifian, M.K., Kesserwani, G., 2021. Benchmarking (multi)wavelet-based dynamic and static non-uniform grid solvers for flood inundation modelling (Simulation results) (Version 1.0) [Data set]. Zenodo. <https://doi.org/10.5281/zenodo.5047565>.
- Shaw, J., Kesserwani, G., Pettersson, P., 2020. Probabilistic Godunov-type hydrodynamic modelling under multiple uncertainties: robust wavelet-based formulations. *Adv. Water Resour.* 137, 103526.
- Shaw, J., Kesserwani, G., Neal, J., Bates, P., Sharifian, M.K., 2021. LISFLOOD-FP 8.0: the new discontinuous Galerkin shallow-water solver for multi-core CPUs and GPUs. *Geosci. Model Dev.* 14 (6), 3577–3602.
- Spiekermann, R., Kienberger, S., Norton, J., Briones, F., Weichselgartner, J., 2015. The Disaster-Knowledge Matrix - Reframing and evaluating the knowledge challenges in disaster risk reduction. *Int. J. Disaster Risk Reduct.* 13, 96–108.
- Teng, J., Jakeman, A.J., Vaze, J., Croke, B.F.W., Dutta, D., Kim, S., 2017. Flood inundation modelling: a review of methods, recent advances and uncertainty analysis. *Environ. Monit. Assess.* 90, 201–216.
- Vaccondio, R., Dal Palù, A., Ferrari, A., Mignosa, P., Aureli, F., Dazzi, S., 2017. A non-uniform efficient grid type for GPU-parallel Shallow Water Equations models. *Environ. Monit. Assess.* 88, 119–137.
- Wang, J.P., Liang, Q., 2011. Testing a new adaptive grid-based shallow flow model for different types of flood simulations. *J. Flood Risk Manage.* 4 (2), 96–103.
- Wing, O.E.J., Bates, P.D., Sampson, C.C., Smith, A.M., Johnson, K.A., Erickson, T.A., 2017. Validation of a 30 m resolution flood hazard model of the conterminous United States. *Water Resour. Res.* 53 (9), 7968–7986.
- Xia, X., Liang, Q., Ming, X., 2019. A full-scale fluvial flood modelling framework based on a high-performance integrated hydrodynamic modelling system (HiPIMS). *Adv. Water Resour.* 132, 103392.
- Zhao, J., Liang, Q., 2022. Novel variable reconstruction and friction term discretisation schemes for hydrodynamic modelling of overland flow and surface water flooding. *Adv. Water Resour.* 163, 104187.