Article:
Kesserwani, G and Liang, Q (2012) Dynamically adaptive grid based discontinuous Galerkin shallow water model. ADVANCES IN WATER RESOURCES, 37. 23 - 39. ISSN 0309-1708

http://dx.doi.org/10.1016/j.advwatres.2011.11.006

Dynamically adaptive grid based discontinuous Galerkin shallow water model

Georges Kesserwani[a], Qiuhua Liang[b,*]

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

[b] *School of Civil Engineering and Geosciences, Newcastle University, Newcastle upon Tyne NE1 7RU, UK*

Abstract: A Godunov-type numerical model, which is based on the local planar Runge-Kutta discontinuous Galerkin (RKDG2) solutions to the two dimensional (2D) shallow water equations (SWEs) on a dynamically adaptive quadrilateral grid system, is developed in this work for shallow water wave simulations, with particular application to flood inundation modelling. To be consistent with the dynamic grid adaptation, the well-balanced RKDG2 framework is reformulated to facilitate realistic flood modelling. Grid adaptation and redistribution of flow data are automated based on simple measures of local flow properties. One analytical and two diagnostic test cases are used to validate the performance of the dynamically adaptive RKDG2 model against an alternative RKDG2 code based on uniform quadrilateral meshes. The adaptive model is then assessed by further applying it to reproduce a laboratory-scale tsunami benchmark case and the historical Malpasset dam-break event. Numerical evidence indicates that the new algorithm is able to resolve the moving wave features adequately and at much less computational cost than the refined uniform grid-based counterpart.

Keywords: discontinuous Galerkin method; shallow water equations; quadrilateral mesh; dynamic grid adaptation; computational efficiency; inundation modelling.

[*]Corresponding author:
E-mails: g.kesserwani@sheffield.ac.uk; (G. Kesserwani) Qiuhua.Liang@ncl.ac.uk (Q. Liang)

1. Introduction

Godunov-type schemes solving the 2D depth-averaged shallow water equations (SWEs) are attractive to hydrodynamic modellers because of their intrinsic capability to represent complex flow features accurately [33,64]. Godunov-type numerical models solving the SWEs have undergone significant theoretical and numerical advancement in recent years. As a whole, a well-established Godunov-type shallow water wave model should be stable and consistent when (a) strong convective transport is involved [63], (b) steep topographic gradients are present [12,17,24,36,39,47,48,73], (c) flooding and drying processes take place [3,7,9,13,18,26,53,50,52,75] and, (d) high roughness values are combined with extremely low water depths [15,16,50,71]. Such models are superior to other flow modelling tools when coping with a situation that demands accurate prediction of violent flood hydrodynamics including the extent of inundation [34;66].

In the last decade, a number of capable Godunov-type shallow flow models have been reported to be successful in real-time flood forecasts. Most of them are based on the first- or second-order non-oscillatory Total Variation Diminishing (TVD) finite volume (FV) approximation (*e.g.*, [3,4,8,13,25,26,27,49,50,65]). However, the utility of these models in more general real-time flood modelling is still questionable. One major issue, which significantly hampers the applicability of these models, is the requirement for high-resolution spatial representation of the detailed domain features in large-scale flow simulations. This inevitably requires the use of highly refined computational meshes that lead to a long run-time, which could become prohibitive even for the most advanced computers [6]. One viable means to improve the the run-time efficiency but not compromise the representation of complex topography is to use dynamic mesh adaptation so that the refined mesh only covers those regions with complex flow or topographic features. For instance, Nikolos and Delis [56] attempted to use a locally refined but static grid to represent more accurately the areas

where the topographic slopes are relatively steep. Although domain topography may significantly impact upon the flow direction, the optimal grid resolution is generally unknown in advance, given the transient nature of most of the numerical solutions [27,44]. This therefore motivates the use of *dynamic grid refinement and coarsening* to optimize the performance of an unsteady shallow water model, which may be realized by utilizing either quadrilateral cells [26,27,39,46,49,60,61] or triangular elements [9,57].

In recent years, the Runge-Kutta Discontinuous Galerkin (RKDG) methods have become popular in solving PDEs [45]. For the hyperbolic conservation laws, RKDG methods can be viewed as a genuine generalization to the conservative formulation initially introduced by Godunov [19,20]. They are attractive due to the fact that they employ local element-oriented solutions of arbitrary order of accuracy and only exchange information between computational cells via numerical fluxes. They also excel in delivering converged solutions (*e.g.*, [42,48,72,74]) and are highly suited to local adaptation of the computational mesh (referred to as "*h-adaptation*") or local adjustment of the order of accuracy (referred to as "*p-adaptation*"). In terms of computational cost, RKDG models are generally much more expensive than other numerical schemes (*e.g.*, those based on the traditional continuous Galerkin finite element method or non-oscillatory FV Godunov-type approach) and their overhead cost obviously increases with the scheme's order of accuracy. As a consequence, employing a second-order RKDG method (RKDG2) provides a reasonable tradeoff between computational efficiency and numerical accuracy. RKDG2 can also substantially reduce the overall algorithm complexity in the implementation of effective slope limiting processes [41], consideration of uneven topography under a well-balanced framework [36,68], inclusion of accurate tracking of a wet/dry front [76] and stable discretization of friction source terms [37].

In solving the conservation laws of the SWEs, the RKDG methods have undergone notable improvements in recent years and they can now provide stability properties similar to robust finite volume Godunov-type schemes (*e.g.*, [1,2,10,4,21,22,23 ,28,29,30,31,35,38,43,44,55,58,59,62,63]. Particular innovations include the work of Krivodonova et al. [40] who introduced a user-parameter-free local slope limiting process to improve the scheme's accuracy and efficiency. Xing and Shu [68,69] reported some useful techniques for accurately handling irregular domain topographies. More recently, much effort has been devoted to the design of wetting and drying conditions [11,21,14,31,35,70] and stable discretization of bed friction terms [37,38].

Irrespective of the aforementioned progress made in recent years, the RKDG models have not, surprisingly, become common in practical flood applications. One of the main reasons may be, debatably, because of the enormous operational costs associated with this group of models. To surmount the barrier of high runtime cost, a number of attempts have been made to increase computational efficiency by devising dynamically adaptive RKDG schemes. Kubatko et al. [44] systematically studied the impacts of global *hp*-adaptation on the computational efficiency and solution reliability of RKDG shallow water solutions on structured uniform meshes. Based on their conclusions, the authors later introduced dynamic *p*-adaption to the RKDG shallow water solutions [43]. With regard to local dynamic *h*-adaptation, Remacle et al. [57] demonstrated its advantages with high-order RKDG models but no information was provided on the computational cost. Most recently, Bader et al. [5] reported a new dynamically *h*-adaptive mesh generator in the context of a RKDG shallow flow solver, with a particular focus on minimizing memory demand. All of these studies have been implemented on grids with triangular elements and involve rather idealized hydraulic test-cases. Interestingly, a recent study by Wirasaet et al. [67] suggested that the use of quadrilateral meshes with low-order RKDG methods is preferable to triangular meshes in

terms of cost effectiveness. Consequently, the payoff in deploying dynamic *h*-adaptation with an RKDG2 scheme that solves the SWEs on structured quadrilateral grids may be quite substantial especially when targeting water wave simulations with moving local features and large-scale flood applications.

The main novel aspect of this work is the installation of dynamic *h*-adaptivity that was initially developed in the context of a finite volume scheme [76] to a well-established RKDG2 2D SWE numerical solver on a quadrilateral mesh [36-38], aiming to enhance its effectiveness. The RKDG2 scheme is therefore reformulated to cope with the (spatial) non-uniformity and (transient) adaptive nature of the mesh while retaining the latest techniques for local slope limiting, handling complex domain topographies and wetting and drying processes. Dynamical grid adaptation is configured by sensing the water wave gradients and the topographic data are stored with the highest resolution attainable. A set of analytical shallow flow tests are first used to comprehensively test the performance of the new dynamically *h*-adaptive RKDG2 model versus an RKDG2 alternative based on uniform quadrilateral meshes of different sizes. The code is then assessed for practical flood modelling by applying it to reproduce a laboratory-scale tsunami case and a field-scale dam-break flood event. Applying an RKDG method to such realistic flood cases constitutes another contribution of this work. Therefore, the present RKDG2 model may be among the very few existing RKDG 2D SWE codes that are suitable for applications to realistic flood modelling.

2. Shallow water equations (SWEs)

Based on mass and momentum conservation, the 2D depth-averaged nonlinear SWEs may be written in a matrix form as

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

Herein t is the time, (x, y) represent the Cartesian coordinates, $\mathbf{U} = [\eta, q_x, q_y]^{\mathrm{T}}$, $\mathbf{F} = [q_x, q_x^2 h^{-1} + \frac{1}{2} g(\eta^2 - 2\eta z),\ q_x q_y h^{-1}]^{\mathrm{T}}$, $\mathbf{G} = [q_y, q_x q_y h^{-1}, q_y^2 h^{-1} + \frac{1}{2} g(\eta^2 - 2\eta z)]^{\mathrm{T}}$, and $\mathbf{S} = [0,\ g\eta S_{0x} - S_{fx}, g\eta S_{0y} - S_{fy}]^{\mathrm{T}}$ are the vectors containing the flow variables, x-direction fluxes, y-direction fluxes and source terms, respectively, g denotes the acceleration due to gravity, $\eta(x, y, t)$ is the free-surface elevation, $h(x, y, t)$ gives the water depth, $z(x, y)$ represents the ground level where $\eta = h + z$, $q_x(x, y, t)$ and $q_y(x, y, t)$ are respectively the x- and y-components of the unit-width discharge. The two Cartesian components of the depth-averaged velocity are calculated by $u = q_x / h$ and $v = q_y / h$. The source term vector may be expressed as $\mathbf{S} = \mathbf{S_b} + \mathbf{S_f}$ with $\mathbf{S_b} = [0, -g\eta\partial_x z, -g\eta\partial_y z]^{\mathrm{T}}$ and $\mathbf{S_f} = [0, S_{fx}, S_{fy}]^{\mathrm{T}}$. $S_{fx} = -\frac{g n_M^2}{h^{1/3}} u\sqrt{u^2 + v^2}$ and $S_{fy} = -\frac{g n_M^2}{h^{1/3}} v\sqrt{u^2 + v^2}$ are the two components of friction source terms with n_M denoting the Manning coefficient.

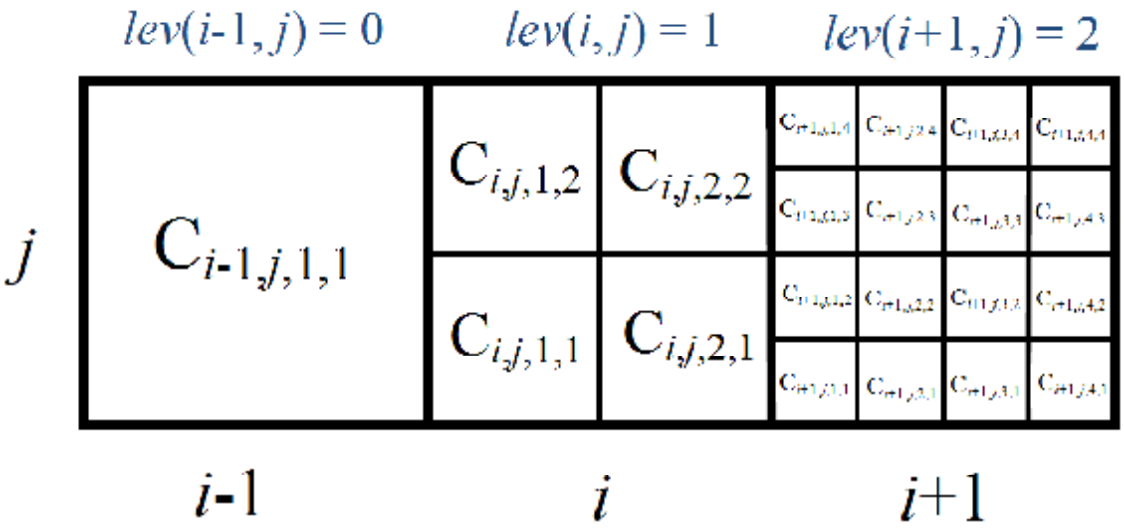


Fig. 1: Regularized non-uniform quadrilateral mesh with genuine indexing.

3. Non-uniform but structured quadrilateral mesh

In order to perform dynamic grid adaptation, a non-uniform but structured quadrilateral mesh is first generated, as described in [51]. To summarise, the problem domain is first discretized using a coarse uniform mesh consisting of $M \times N$ quadrilaterals or cells. This is called "*background mesh*", on which a cell is termed "*background cell*" and has a dimension of $\Delta x \times \Delta y$. Secondly, user-defined seeding points [32] are scattered over those areas where higher resolution mesh is desired (closed seeding point sets may be also distributed to approximate domain geometry). Thirdly, a background cell containing one or more seeding points is detected and refined by specifying its subdivision level. The refinement is performed in a fractal sense, *i.e.* the cell size reduces by a factor of two whenever the refinement level increases by one. By default, a background cell containing one or more seeding points will be given a user-determined maximum subdivision level, denoted by "$lmax$" (where $lmax \in \mathbb{N}$). The cell is consequently subdivided uniformly into $2^{lmax} \times 2^{lmax}$ quadrilateral child cells of dimension $\frac{\Delta x}{2^{lmax}} \times \frac{\Delta y}{2^{lmax}}$. Finally, the mesh is regularized so that no background cell is adjacent to another background neighbour that differs in subdivision level by more than '1', as seen in Fig. 1.

Since the above procedure generates a local uniform mesh inside each background cell, each computational (child) cell "*C*" can be recognized by an index system that consists of four entries (i, j, i_s, j_s) where $1 \le i \le M$, $1 \le j \le N$ and $1 \le i_s, j_s \le 2^{lev(i,j)}$ with $0 \le lev(i, j) \le lmax$ storing the subdivision level of background cell (i, j). Associating (i, j, i_s, j_s) with a reference ID number "i_c", *i.e.* $(i, j, i_s, j_s) \mapsto i_c$, the 2D domain ($\Omega$) subdivision may be expressed as

$$\Omega = \left\{ \bigcup_{i_c} \overline{C}_{i_c} \text{ and } C_{i_{c1}} \cap C_{i_{c2}} = \varnothing \text{ if } i_{c1} \neq i_{c2} \right\} \tag{2}$$

Cell C_{i_c} has an area of $\Delta x_c \times \Delta y_c$ ($\Delta x_c = \Delta x / 2^{lev(i_c)}$ and $\Delta y_c = \Delta y / 2^{lev(i_c)}$) and centres at (x_c, y_c), which is consequently defined as $C_{i_c} = \left[x_c - \frac{\Delta x_c}{2};\ x_c + \frac{\Delta x_c}{2}\right] \times \left[y_c - \frac{\Delta y_c}{2};\ y_c + \frac{\Delta y_c}{2}\right]$. The neighbouring information of any child cell can be easily obtained and stored, without the need for a hierarchical data structure [51].

4. Local RKDG2 adaptive flow solution

The RKDG method adopted in this paper originates from the formulation proposed in Cockburn and Shu [19,20]. In this section, the RKDG2 scheme is constructed for shallow flow simulations, with a key-focus on its implementation on dynamically adaptive quadrilateral grids.

4.1 *Well-balanced framework in 2D*

The RKDG2 algorithm solves for a local planar solution $\mathbf{U^h}(x,y,t) = \left[\eta^h, q_x^h, q_y^h\right]^{\mathrm{T}}$ to (1), which is a second-order approximation in space and time [37]. Over a cell C_{i_c}, the local solution $\mathbf{U^h}(x,y,t)|_{C_{i_c}}$ is spanned by two perpendicular lines intersecting at (x_c, y_c) and is determined by average coefficients $\mathbf{U}_{i_c}^0(t)$ and slope coefficients $\mathbf{U}_{i_c}^{1x}(t)$ and $\mathbf{U}_{i_c}^{1y}(t)$ according to the following local plane equation

$$\mathbf{U^h}(x,y,t)\Big|_{C_{i_c}} = \mathbf{U}_{i_c}^0(t) + \mathbf{U}_{i_c}^{1x}(t)\frac{(x-x_c)}{\Delta x_c/2} + \mathbf{U}_{i_c}^{1y}(t)\frac{(y-y_c)}{\Delta y_c/2} \quad \left(\forall (x,y) \in C_{i_c}\right) \tag{3}$$

From a set of given initial conditions, *i.e.* $\mathbf{U_0}(x,y) = \mathbf{U}(x,y,0)$, the local approximate solution $\mathbf{U^h}(x,y,t)|_{C_{i_c}}$ is initialized based on a local planar projection to $\mathbf{U_0}(x,y)$ onto each C_{i_c}. The average and slope coefficients of $\mathbf{U^h}(x,y,0)|_{C_{i_c}}$ can be approximated by [36]

$$\begin{aligned}
\mathbf{U}_{i_c}^{0}(0) &= \tfrac{1}{4}\left[\mathbf{U_0}\left(x_c+\tfrac{\Delta x_c}{2},y_c\right)+\mathbf{U_0}\left(x_c-\tfrac{\Delta x_c}{2},y_c\right)+\mathbf{U_0}\left(x_c,y_c+\tfrac{\Delta y_c}{2}\right)+\mathbf{U_0}\left(x_c,y_c-\tfrac{\Delta y_c}{2}\right)\right]\\
\mathbf{U}_{i_c}^{1x}(0) &= \tfrac{1}{2}\left[\mathbf{U_0}\left(x_c+\tfrac{\Delta x_c}{2},y_c\right)-\mathbf{U_0}\left(x_c-\tfrac{\Delta x_c}{2},y_c\right)\right]\\
\mathbf{U}_{i_c}^{1y}(0) &= \tfrac{1}{2}\left[\mathbf{U_0}\left(x_c,y_c+\tfrac{\Delta y_c}{2}\right)-\mathbf{U_0}\left(x_c,y_c-\tfrac{\Delta y_c}{2}\right)\right]
\end{aligned} \tag{4}$$

The well-balanced property can be automatically satisfied if a similar local planar approximation is also applied to the topography function as in (3), denoted by $z^h(x,y)|_{C_{i_c}}$, which must be continuous across the cell faces of C_{i_c} [68]

$$z^h(x,y)\Big|_{C_{i_c}} = z_{i_c}^0 + z_{i_c}^{1x}\frac{(x-x_c)}{\Delta x_c/2} + z_{i_c}^{1y}\frac{(y-y_c)}{\Delta y_c/2} \qquad \left(\forall (x,y)\in C_{i_c}\right) \tag{5}$$

in which $z_{i_c}^0$, $z_{i_c}^{1x}$ and $z_{i_c}^{1y}$ are the topography-associated (constant scalar) coefficients, which are obtained by applying a similar mathematical manipulation to the topography function $z(x,y)$ as in (4). From (5), local approximations to the topography gradients in $\mathbf{S_b}$ can be obtained by partial differentiation with respect to *x* and *y*, respectively. For instance, over a cell C_{i_c} the topography gradient $\partial_x z$ is discretized as

$$\partial_x z(x,y) \approx -\partial_x\left[z^h(x,y)\Big|_{C_{i_c}}\right] = -\partial_x\left[z_{i_c}^0 + z_{i_c}^{1x}\frac{(x-x_c)}{\Delta x_c/2} + z_{i_c}^{1y}\frac{(y-y_c)}{\Delta y_c/2}\right] = -\frac{z_{i_c}^{1x}}{\Delta x_c/2} \tag{6}$$

After initialization, the local approximate solution $\mathbf{U^h}(x,y,t)\big|_{C_{i_c}}$ is solved in space and time with its coefficients updated by solving locally the following ODEs

$$\begin{aligned}
\partial_t \mathbf{U}_{i_c}^{0} &= \mathbf{L}_{i_c}^{0}\\
\partial_t \mathbf{U}_{i_c}^{1x} &= \mathbf{L}_{i_c}^{1x}\\
\partial_t \mathbf{U}_{i_c}^{1y} &= \mathbf{L}_{i_c}^{1y}
\end{aligned} \tag{7}$$

$\mathbf{L}_{i_c}^{0}$, $\mathbf{L}_{i_c}^{1x}$ and $\mathbf{L}_{i_c}^{1y}$ are nonlinear vectors of space-functions that are locally derived from the system of conservation laws (1). They can be explicitly expressed as

$$\mathbf{L}_{i_c}^{0} = -\frac{\tilde{\mathbf{F}}_{i_c}^{\mathrm{E}} - \tilde{\mathbf{F}}_{i_c}^{\mathrm{W}}}{\Delta x_c} - \frac{\tilde{\mathbf{G}}_{i_c}^{\mathrm{N}} - \tilde{\mathbf{G}}_{i_c}^{\mathrm{S}}}{\Delta y_c} + \mathbf{S_b}\left(\mathbf{U}_{i_c}^{0}, z_{i_c}^{1x}, z_{i_c}^{1y}\right) \tag{8}$$

$$\mathbf{L}_{i_c}^{1x} = -\frac{3}{\Delta x_c}\left\{\tilde{\mathbf{F}}_{i_c}^{\mathrm{E}} + \tilde{\mathbf{F}}_{i_c}^{\mathrm{W}} - \mathbf{F}\left(\mathbf{U}_{i_c}^{0} + \frac{\hat{\mathbf{U}}_{i_c}^{1x}}{\sqrt{3}}, z_{i_c}^{0} + \frac{z_{i_c}^{1x}}{\sqrt{3}}\right) - \mathbf{F}\left(\mathbf{U}_{i_c}^{0} - \frac{\hat{\mathbf{U}}_{i_c}^{1x}}{\sqrt{3}}, z_{i_c}^{0} - \frac{z_{i_c}^{1x}}{\sqrt{3}}\right) - \frac{\Delta x_c\sqrt{3}}{6}\left[\mathbf{S_b}\left(\mathbf{U}_{i_c}^{0} + \frac{\hat{\mathbf{U}}_{i_c}^{1x}}{\sqrt{3}}, z_{i_c}^{1x}, z_{i_c}^{1y}\right) - \mathbf{S_b}\left(\mathbf{U}_{i_c}^{0} - \frac{\hat{\mathbf{U}}_{i_c}^{1x}}{\sqrt{3}}, z_{i_c}^{1x}, z_{i_c}^{1y}\right)\right]\right\} \tag{9}$$

$$\mathbf{L}_{i_c}^{1y} = -\frac{3}{\Delta y_c}\left\{\tilde{\mathbf{G}}_{i_c}^{\mathrm{N}} + \tilde{\mathbf{G}}_{i_c}^{\mathrm{S}} - \mathbf{G}\left(\mathbf{U}_{i_c}^{0} + \frac{\hat{\mathbf{U}}_{i_c}^{1y}}{\sqrt{3}}, z_{i_c}^{0} + \frac{z_{i_c}^{1y}}{\sqrt{3}}\right) - \mathbf{G}\left(\mathbf{U}_{i_c}^{0} - \frac{\hat{\mathbf{U}}_{i_c}^{1y}}{\sqrt{3}}, z_{i_c}^{0} - \frac{z_{i_c}^{1y}}{\sqrt{3}}\right) - \frac{\Delta y_c\sqrt{3}}{6}\left[\mathbf{S_b}\left(\mathbf{U}_{i_c}^{0} + \frac{\hat{\mathbf{U}}_{i_c}^{1y}}{\sqrt{3}}, z_{i_c}^{1x}, z_{i_c}^{1y}\right) - \mathbf{S_b}\left(\mathbf{U}_{i_c}^{0} - \frac{\hat{\mathbf{U}}_{i_c}^{1y}}{\sqrt{3}}, z_{i_c}^{1x}, z_{i_c}^{1y}\right)\right]\right\} \tag{10}$$

The approximate solution $\mathbf{U^h}$ is generally not continuous across the edges of C_{i_c}. Hence, numerical fluxes $\tilde{\mathbf{F}}_{i_c}^{\mathrm{E}}$, $\tilde{\mathbf{F}}_{i_c}^{\mathrm{W}}$, $\tilde{\mathbf{G}}_{i_c}^{\mathrm{N}}$ and $\tilde{\mathbf{G}}_{i_c}^{\mathrm{S}}$ across the eastern, western, northern and southern faces of C_{i_c} should be calculated by means of two-argument numerical flux functions $\tilde{\mathbf{F}}$ and $\tilde{\mathbf{G}}$, which are here evaluated by an HLLC approximate Riemann solver [64]. For instance, the numerical flux ($\tilde{\mathbf{F}}_{i_c}^{\mathrm{E}}$) across the eastern face of C_{i_c}, *i.e.* $x_c + \frac{\Delta x_c}{2}$, shared by the neighbouring cell $C_{i_{nei}}$ can be estimated by

$$\tilde{\mathbf{F}}_{i_c}^{\mathrm{E}} = \tilde{\mathbf{F}}\left(\mathbf{U^h}\left(x_c + \tfrac{\Delta x_c}{2}, y\right)\Big|_{C_{i_c}}, \mathbf{U^h}\left(x_c + \tfrac{\Delta x_c}{2}, y\right)\Big|_{C_{i_{nei}}}\right) \tag{11}$$

in which

$$\begin{aligned}
\mathbf{U^h}\left(x_c + \tfrac{\Delta x_c}{2}, y\right)\Big|_{C_{i_c}} &= \mathbf{U}_{i_c}^{0}(t) + \hat{\mathbf{U}}_{i_c}^{1x}(t) + \hat{\mathbf{U}}_{i_c}^{1y}(t)\frac{(y - y_c)}{\Delta y_c / 2} \\
\mathbf{U^h}\left(x_c + \tfrac{\Delta x_c}{2}, y\right)\Big|_{C_{i_{nei}}} &= \mathbf{U}_{i_{nei}}^{0}(t) + \hat{\mathbf{U}}_{i_{nei}}^{1x}(t)\frac{\left(x_c + \frac{\Delta x_c}{2} - x_{nei}\right)}{\Delta x_{nei} / 2} + \hat{\mathbf{U}}_{i_{nei}}^{1y}(t)\frac{(y - y_{nei})}{\Delta y_{nei} / 2}
\end{aligned} \tag{12}$$

On the current grid system, cells C_{i_c} and $C_{i_{nei}}$ can be of different sizes and the value of y must be carefully selected so that the flux $\tilde{\mathbf{F}}_{i_c}^{\mathrm{E}}$ is evaluated at the face-centre of the smaller cell between C_{i_c} and $C_{i_{nei}}$. The other fluxes $\tilde{\mathbf{F}}_{i_c}^{\mathrm{W}}$, $\tilde{\mathbf{G}}_{i_c}^{\mathrm{N}}$ and $\tilde{\mathbf{G}}_{i_c}^{\mathrm{S}}$ can be calculated similarly.

Conservative computation of these interface fluxes will be discussed in more details in Subsections 4.2 and 4.3.

The time derivative terms in (7) are discretized by a two-stage Runge-Kutta method, which leads to the following temporal update scheme for the average and slope coefficients

$$\begin{aligned}\left(\mathbf{U}_{i_c}^{0,1x,1y}\right)^{n+1/2} &= \left(\mathbf{U}_{i_c}^{0,1x,1y}\right)^{n} + \Delta t\ \left(\mathbf{L}_{i_c}^{0,1x,1y}\right)^{n} \\ \left(\mathbf{U}_{i_c}^{0,1x,1y}\right)^{n+1} &= \frac{1}{2}\left[\left(\mathbf{U}_{i_c}^{0,1x,1y}\right)^{n} + \left(\mathbf{U}_{i_c}^{0,1x,1y}\right)^{n+1/2} + \Delta t\ \left(\mathbf{L}_{i_c}^{0,1x,1y}\right)^{n+1/2}\right]\end{aligned} \tag{13}$$

Controlled by the CFL condition, the time step is restricted to a Courant number no bigger than 0.3 [19].

It should be noted that only the topography-associated coefficients are included in (8) – (10) and the friction source terms ($\mathbf{S}_f$) are separately integrated, before each RK stage, using an implicit discretization scheme to ensure stable numerical simulation at cells near the wet/dry interfaces (see [37,38] for details).

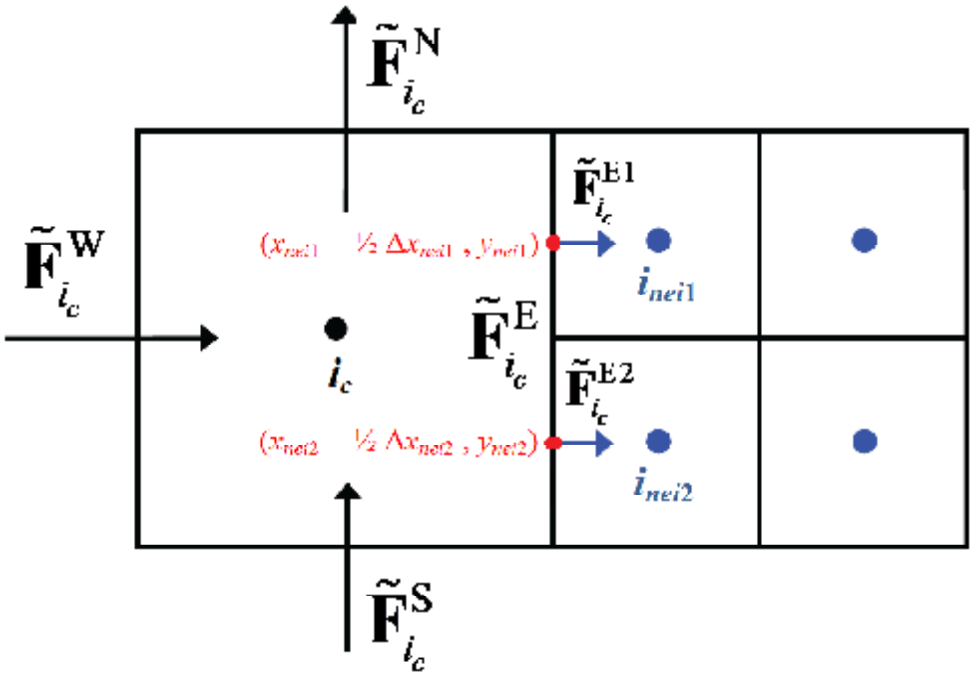


Fig. 2: Conservative flux calculation at a cell with a finer eastern neighbour.

4.2 *Conservative flux calculation*

As shown in Fig. 2, when cell C_{i_c} shares an edge with two finer cells, the flux $\tilde{\mathbf{F}}_{i_c}^{\mathrm{E}}$ leaving cell C_{i_c} must be equal to the sum of the fluxes entering the subdivided cells $C_{i_{nei1}}$ and $C_{i_{nei2}}$ to ensure mass and momentum conservation. Therefore,

$$\tilde{\mathbf{F}}_{i_c}^{\mathrm{E}} = \tfrac{1}{2}\left(\tilde{\mathbf{F}}_{i_c}^{\mathrm{E1}} + \tilde{\mathbf{F}}_{i_c}^{\mathrm{E2}}\right) \tag{14}$$

where $\tilde{\mathbf{F}}_{i_c}^{\mathrm{E1}}$ and $\tilde{\mathbf{F}}_{i_c}^{\mathrm{E2}}$ contain the fluxes across the western faces of cells $C_{i_{nei1}}$ and $C_{i_{nei2}}$, respectively. To calculate $\tilde{\mathbf{F}}_{i_c}^{\mathrm{E1}}$, the values of flow variables are first found on either side of western face of cells $C_{i_{nei1}}$

$$\begin{aligned}
\mathbf{U}_{\mathrm{E1}}^{-} &= \mathbf{U}^{\mathbf{h}}\left(x_{nei1} - \tfrac{\Delta x_{nei1}}{2}, y_{nei1}\right)\Big|_{C_{i_c}} = \left[\eta_{\mathrm{E1}}^{-}, q_{x,\mathrm{E1}}^{-}, q_{y,\mathrm{E1}}^{-}\right]^{\mathrm{T}} \\
\mathbf{U}_{\mathrm{E1}}^{+} &= \mathbf{U}^{\mathbf{h}}\left(x_{nei1} - \tfrac{\Delta x_{nei1}}{2}, y_{nei1}\right)\Big|_{C_{i_{nei1}}} = \left[\eta_{\mathrm{E1}}^{+}, q_{x,\mathrm{E1}}^{+}, q_{y,\mathrm{E1}}^{+}\right]^{\mathrm{T}}
\end{aligned} \tag{15}$$

to derive the associated bed elevation and "depth-positivity-preserving" Riemann states, *i.e.* $\left\{z_{\mathrm{E1}}^{*}, \mathbf{U}_{\mathrm{E1}}^{\pm,*}\right\}$ (which will be described in more details in Subsection 4.3). The topography level and Riemann states $\left\{z_{\mathrm{E2}}^{*}, \mathbf{U}_{\mathrm{E2}}^{\pm,*}\right\}$ relative to the western face-centre of cell $C_{i_{nei2}}$ can be found similarly. Finally, when they are needed (*i.e.*, to complete the local and temporal correction step (23) relative to the wetting and drying condition, which will be detailed in Subsection 4.3), the flow and topography data at the centre of the eastern face of cell C_{i_c} $(x_c + \frac{\Delta x_c}{2}, y_c)$ are gathered by interpolation

$$\begin{aligned}
\mathbf{U}_{\mathrm{E}}^{-,*} &= \tfrac{1}{2}\left(\mathbf{U}_{\mathrm{E1}}^{-,*} + \mathbf{U}_{\mathrm{E2}}^{-,*}\right) \\
z_{\mathrm{E}}^{*} &= \tfrac{1}{2}\left(z_{\mathrm{E1}}^{*} + z_{\mathrm{E2}}^{*}\right)
\end{aligned} \tag{16}$$

4.3 *Wetting and drying condition*

In order to have a numerical scheme free of negative water depth for applications involving wetting and drying over complex topography, the Riemann states must be constructed to be "*depth-positivity-preserving*" before calling a proper Riemann solver to compute interface fluxes. Subsequently, the associated "depth-positivity-preserving" planar approximations must also be regenerated for evaluating other flux and source terms within the space operator functions (8) – (10). Detailed derivation of this wetting and drying condition can be found in Kesserwani and Liang [37, 38]. Herein, the technique is modified for implementation in the current adapted mesh RKDG2 scheme.

Firstly, depth-positivity-preserving face values and subsequently Riemann states at the centre of an interface of C_{i_c} are reconstructed for calculating numerical fluxes. Considering the eastern interface of cell C_{i_c} shared by a similar size neighbour $C_{i_{nei}}$, the face values of the flow variables at $(x_c + \frac{\Delta x_c}{2}, y_c)$ can be procured by evaluating (12) with $y = y_c$

$$\begin{aligned} \mathbf{U}_{\mathrm{E}}^{-} &= \mathbf{U}^{\mathbf{h}}\left(x_c + \tfrac{\Delta x_c}{2}, y_c\right)\Big|_{C_{i_c}} = \left[\eta_{\mathrm{E}}^{-}, q_{x,\mathrm{E}}^{-}, q_{y,\mathrm{E}}^{-}\right]^{\mathrm{T}} \\ \mathbf{U}_{\mathrm{E}}^{+} &= \mathbf{U}^{\mathbf{h}}\left(x_c + \tfrac{\Delta x_c}{2}, y_c\right)\Big|_{C_{i_{nei}}} = \left[\eta_{\mathrm{E}}^{+}, q_{x,\mathrm{E}}^{+}, q_{y,\mathrm{E}}^{+}\right]^{\mathrm{T}} \end{aligned} \tag{17}$$

At the same point, the face values of the water depth, h_{E}^{-} and h_{E}^{+} must be also evaluated. This is done by using a formula similar to (12) that acts on the coefficients corresponding to local depth approximate solutions defined by $(\eta - z)^h = \eta^h - z^h$ (*i.e.*, $h_{\mathrm{E}}^{-} = (\eta - z)^h (x_c + \frac{\Delta x_c}{2}, y_c)\Big|_{C_{i_c}}$ and $h_{\mathrm{E}}^{+} = (\eta - z)^h (x_c + \frac{\Delta x_c}{2}, y_c)\Big|_{C_{i_{nei}}}$. The associated velocity components ($u_{\mathrm{E}}^{\pm}$ and $v_{\mathrm{E}}^{\pm}$) are then calculated by

$$u_{\mathrm{E}}^{\pm} = q_{x,\mathrm{E}}^{\pm} / h_{\mathrm{E}}^{\pm} \quad \text{and} \quad v_{\mathrm{E}}^{\pm} = q_{y,\mathrm{E}}^{\pm} / h_{\mathrm{E}}^{\pm} \tag{18}$$

Rather than being calculated by (18), velocities are directly set to zero if C_{i_c} is classified as a dry cell with $\min(h^h)|_{C_{i_c}} \le 10^{-8}$. $\min(h^h)|_{C_{i_c}}$ returns the minimum value of the depth components that are required for computing the local fluxes in the space operators (9) and (10), *i.e.*

$$\min(h^h)|_{C_{i_c}} = \min\left[(\eta_{i_c}^0 - z_{i_c}^0) \pm (\eta_{i_c}^{1x} - z_{i_c}^{1x}), (\eta_{i_c}^0 - z_{i_c}^0), (\eta_{i_c}^0 - z_{i_c}^0) \pm (\eta_{i_c}^{1y} - z_{i_c}^0)\right] \quad (19)$$

To ensure a consistent set of reconstructed flow data [54], the corresponding topography evaluations ($z_{\mathrm{E}}^{\pm}$) must be *numerically deduced* from previously reconstructed face values of water surface elevation and water depth, *i.e.*

$$z_{\mathrm{E}}^{\pm} = \eta_{\mathrm{E}}^{\pm} - h_{\mathrm{E}}^{\pm} \quad (20)$$

However, (20) may lead to a discontinuous bed topography across the cell face, *i.e.* $z_{\mathrm{E}}^{-} \neq z_{\mathrm{E}}^{+}$, which breaks the assumption of a well-balanced DG scheme [68]. Therefore, a single face value of the topography should be redefined via $z_{\mathrm{E}}^{*} = \max(z_{\mathrm{E}}^{-}, z_{\mathrm{E}}^{+})$, based on which the positivity-preserving Riemann states of water depth, and consequently other Riemann states, are defined

$$\begin{aligned} h_{\mathrm{E}}^{\pm,*} &= \max\left(0, \eta_{\mathrm{E}}^{\pm} - z_{\mathrm{E}}^{*}\right) \\ \eta_{\mathrm{E}}^{\pm,*} &= h_{\mathrm{E}}^{\pm,*} + z_{\mathrm{E}}^{*} \\ q_{x,\mathrm{E}}^{\pm,*} &= h_{\mathrm{E}}^{\pm,*} u_{\mathrm{E}}^{\pm} \\ q_{y,\mathrm{E}}^{\pm,*} &= h_{\mathrm{E}}^{\pm,*} v_{\mathrm{E}}^{\pm} \end{aligned} \quad (21)$$

According to [52], the Riemann states produced in (21) may still be inadequate to properly handle the situation where cell C_{i_c} is close to the wet/dry front, *e.g.* C_{i_c} is wet but $C_{i_{nei}}$ is dry with specifically $\eta_{\mathrm{E}}^{-} < z_{\mathrm{E}}^{*}$. In this case, the actual free-surface η_{E}^{-} at the eastern face-centre of cell C_{i_c} is overwritten after (21) to become the same as the ground elevation z_{E}^{*} defined at the common cell face, *i.e.* $\eta_{\mathrm{E}}^{-,*} = z_{\mathrm{E}}^{*} \neq \eta_{\mathrm{E}}^{-}$. To ensure that the actual free-surface elevation is

conserved at any wet cell C_{i_c}, the following local and temporal adjustment must be appended to (22)

$$\begin{aligned}
&\Delta\eta_{\mathrm{E}}^{-} = \max\left[0, -\left(\eta_{\mathrm{E}}^{-} - z_{\mathrm{E}}^{*}\right)\right] \\
&\eta_{\mathrm{E}}^{\pm,*} \leftarrow \eta_{\mathrm{E}}^{\pm,*} - \Delta\eta_{\mathrm{E}}^{-} \\
&z_{\mathrm{E}}^{*} \leftarrow z_{\mathrm{E}}^{*} - \Delta\eta_{\mathrm{E}}^{-}
\end{aligned} \tag{22}$$

Hence (22) brings back the continuous motionless states for the free-surface elevation while adjusting the single value of the topography to ensure zero momentum fluxes across the eastern face. The reconstruction procedure thus must be in place to ensure a well-balanced scheme. The flux $\tilde{\mathbf{F}}_{i_c}^{\mathrm{E}}$ is then calculated by inputting the reconstructed bed elevation and Riemann states $\left\{z_{\mathrm{E}}^{*}, \eta_{\mathrm{E}}^{\pm,*}, q_{x,\mathrm{E}}^{\pm,*}, q_{y,\mathrm{E}}^{\pm,*}\right\}$ to the HLLC numerical flux function. Similarly, fluxes $\tilde{\mathbf{F}}_{i_c}^{\mathrm{W}}$, $\tilde{\mathbf{F}}_{i_c}^{\mathrm{N}}$ and $\tilde{\mathbf{F}}_{i_c}^{\mathrm{S}}$ are also evaluated.

Secondly, the local average and slope coefficients of both flow variables and bed elevation must be regenerated to be consistent with the depth-positivity-preserving reconstructed data in (21) and (22) for evaluating the source terms and local fluxes in (8) – (10). This is an essential step for a "well-balanced" and "depth-positivity-preserving" RKDG2 scheme [38]. By denoting $\mathbf{U}_{\mathrm{E}}^{\pm,*} = \left[\eta_{\mathrm{E}}^{\pm,*}, q_{x,\mathrm{E}}^{\pm,*}, q_{y,\mathrm{E}}^{\pm,*}\right]^{\mathrm{T}}$, $\mathbf{U}_{\mathrm{W}}^{\pm,*} = \left[\eta_{\mathrm{W}}^{\pm,*}, q_{x,\mathrm{W}}^{\pm,*}, q_{y,\mathrm{W}}^{\pm,*}\right]^{\mathrm{T}}$, $\mathbf{U}_{\mathrm{N}}^{\pm,*} = \left[\eta_{\mathrm{N}}^{\pm,*}, q_{x,\mathrm{N}}^{\pm,*}, q_{y,\mathrm{N}}^{\pm,*}\right]^{\mathrm{T}}$ and $\mathbf{U}_{\mathrm{S}}^{\pm,*} = \left[\eta_{\mathrm{S}}^{\pm,*}, q_{x,\mathrm{S}}^{\pm,*}, q_{y,\mathrm{S}}^{\pm,*}\right]^{\mathrm{T}}$, the corresponding local average and slope coefficients are found by re-implementing the relationship (4), *i.e.*

$$\begin{aligned}
&\overline{\mathbf{U}}_{i_c}^{0} = \tfrac{1}{4}\left[\mathbf{U}_{\mathrm{E}}^{-,*} + \mathbf{U}_{\mathrm{W}}^{+,*} + \mathbf{U}_{\mathrm{N}}^{-,*} + \mathbf{U}_{\mathrm{S}}^{+,*}\right] && \overline{z}_{i_c}^{0} = \tfrac{1}{4}\left[z_{\mathrm{E}}^{*} + z_{\mathrm{W}}^{*} + z_{\mathrm{N}}^{*} + z_{\mathrm{S}}^{*}\right] \\
&\overline{\mathbf{U}}_{i_c}^{1x} = \tfrac{1}{2}\left[\mathbf{U}_{\mathrm{E}}^{-,*} - \mathbf{U}_{\mathrm{W}}^{+,*}\right] && \text{and} \quad \overline{z}_{i_c}^{1x} = \tfrac{1}{2}\left[z_{\mathrm{E}}^{*} - z_{\mathrm{W}}^{*}\right] \\
&\overline{\mathbf{U}}_{i_c}^{1y} = \tfrac{1}{2}\left[\mathbf{U}_{\mathrm{N}}^{-,*} - \mathbf{U}_{\mathrm{S}}^{+,*}\right] && \overline{z}_{i_c}^{1y} = \tfrac{1}{2}\left[z_{\mathrm{N}}^{*} - z_{\mathrm{S}}^{*}\right]
\end{aligned} \tag{23}$$

It should be stressed that (23) is only applied locally and temporally to evaluate the space operators (8) – (10), but not used to change the global coefficient values related to flow variables and bed elevation.

The numerical fluxes ($\tilde{\mathbf{F}}_{i_c}^{\mathrm{E}}$, $\tilde{\mathbf{F}}_{i_c}^{\mathrm{W}}$, $\tilde{\mathbf{F}}_{i_c}^{\mathrm{N}}$ and $\tilde{\mathbf{F}}_{i_c}^{\mathrm{S}}$) and the modified coefficients in (23) are hence employed to compute the local spatial operators $\mathbf{L}_{i_c}^{0}$, $\mathbf{L}_{i_c}^{1x}$ and $\mathbf{L}_{i_c}^{1y}$ in (8) – (10), which are in turn used to complete the updating process at each RK stage in (13).

4.4 *Local slope limiting*

To eliminate spurious oscillations in the numerical solution near sharp gradients, the variation of slope coefficients (*i.e.* $\mathbf{U}_{i_c}^{1x}$ and $\mathbf{U}_{i_c}^{1y}$) must be restricted by utilizing TVD FV slope limiters. For DG methods, localized slope limiting has been accepted to be very effective (*e.g.* [38]). In implementing a localized slope limiting process, local slope coefficients are measured and those "*troubled-slopes*" are identified according to the criterion derived by Krivodonova et al. [40]. Then only the variation of those troubled-slopes is controlled using a TVD-*minmod* slope limiter [19].

On the current non-uniform grid system, an arbitrary cell C_{i_c} may be adjacent to a neighbour $C_{i_{nei}}$ of different size but the "troubled-slope detection and limiting" process is to be carried out on a uniform grid template, similar to the aforementioned flow calculation. Fig. 2 illustrates an example where the eastern neighbours are of one level finer than C_{i_c} and other cases may be derived by analogy. In order to form a local uniform grid template, an eastern ghost neighbour $C_{i_{nei}}$ is imposed, which should be at the same level as C_{i_c}. At the centre of the ghost cell, the flow data are obtained via bi-linear interpolation that is consistent to the current planar RKDG solutions. After that, based on the similar expressions to (12) but replacing the limited slope coefficients with the original ones (*i.e.*, the ones without a "*hat*") and noting that $\Delta x_{nei} = \Delta x_c$ and $\Delta y_{nei} = \Delta y_c$, a discontinuity detector at the eastern face centre of C_{i_c} can be implemented as

$$\mathbf{DS}_{i_c}^{\mathrm{E}} = \frac{\left| \mathbf{U}^{\mathbf{h}}\left(x_c + \frac{\Delta x_c}{2}, y_c\right)\Big|_{C_{i_{nei}}} - \mathbf{U}^{\mathbf{h}}\left(x_c + \frac{\Delta x_c}{2}, y_c\right)\Big|_{C_{i_c}} \right|}{\left|\Delta x_c / 2\right| \max\left(\left|\mathbf{U}_{i_c}^{0} - \mathbf{U}_{i_c}^{1x}/\sqrt{3}\right|, \left|\mathbf{U}_{i_c}^{0} + \mathbf{U}_{i_c}^{1y}/\sqrt{3}\right|\right)} \tag{24}$$

The discontinuity detector $\mathbf{DS}_{i_c}^{\mathrm{W}}$ at the western face-centre of cell C_{i_c} can be evaluated in a similar manner. The slope coefficient $\mathbf{U}_{i_c}^{1x}$ is classified as 'troubled' if the discontinuity detector at either eastern or western face is greater than the unity, *i.e.*

$$\max\left(\mathbf{DS}_{i_c}^{\mathrm{W}}, \mathbf{DS}_{i_c}^{\mathrm{E}}\right) > 1 \tag{25}$$

Under such a situation, the TVD-*minmod* slope limiter is applied to control the variation of the slope coefficient $\mathbf{U}_{i_c}^{1x}$ and therefore produce the controlled slope coefficient $\hat{\mathbf{U}}_{i_c}^{1x}$ via

$$\hat{\mathbf{U}}_{i_c}^{1x} = minmod\left(\mathbf{U}_{i_c}^{1x}, \mathbf{U}_{i_{nei}}^{0} - \mathbf{U}_{i_c}^{0}, \mathbf{U}_{i_c}^{0} - \mathbf{U}_{i_w}^{0}\right) \tag{26}$$

$\mathbf{U}_{i_w}^{0}$ stands for the averaged coefficient associated with the local solution at the western neighbour C_{iw} that has the same size as C_{i_c} (if not, bi-linear interpolation formulae will be used). The *minmod* function produces a zero output if the sign of any entry is different from others. Otherwise, $\hat{\mathbf{U}}_{i_c}^{1x}$ equals the slope with smallest magnitude. If (25) is fault, $\mathbf{U}_{i_c}^{1x}$ is not a 'troubled-slope' and (26) is not activated. The original slope coefficient is retained, *i.e.* $\hat{\mathbf{U}}_{i_c}^{1x} = \mathbf{U}_{i_c}^{1x}$.

It should be pointed out that (24) – (26) are applicable to cell C_{i_c} only when its eight neighbours are flood cells. This indicates that the "troubled-slope detection and limiting" is deactivated at a wet cell C_{i_c} that is adjacent to one or more dry cells. At such a cell, $\hat{\mathbf{U}}_{i_c}^{1x} = \mathbf{U}_{i_c}^{1x}$ and $\hat{\mathbf{U}}_{i_c}^{1y} = \mathbf{U}_{i_c}^{1y}$ are set to retain high resolution wet/dry front tracking (a detailed investigation can be found in [76]).

4.5 *Grid adaptation and data repartition*

The current model performs *dynamic h*-adaptation at every time step in line with a *gradient-based sensor* motivated by its simplicity and convenience to water wave type problems. In the context of an RKDG2 scheme, the mesh refinement, or coarsening, is triggered by sensing the magnitude of the *local* water surface gradient coefficients, *i.e.*

$$E_{i_c} = \sqrt{\left(\frac{\eta_{i_c}^{1x}}{\Delta x_c / 2}\right)^2 + \left(\frac{\eta_{i_c}^{1y}}{\Delta y_c / 2}\right)^2} \tag{27}$$

The local gradient magnitudes E_{ic} are evaluated at all of the flow cells and included in a vector **E**, which is then sorted to return the q^{th}-quantile $P_q(\mathbf{E})$. Herein $q = 1 - s_a$ with s_a representing the sensitivity of the grid adaptation to the gradient sensor (27). In this work, we select s_a = 25%, which means that 25% of the flow cells are subjected to grid adaptation. The gradients are finally normalized by $\Theta_{i_c} = E_{i_c} / P_q(\mathbf{E})$.

If over a flow cell C_{i_c} the gradient magnitude has $\Theta_{i_c} > 1$, the associated background cell will be marked for refinement and its subdivision level will increase by '1' (unless it has already been subdivided to the maximum level). Independent of this, those cells defining the wet/dry front are also marked for mesh refinement. For the grid coarsening, if all of the child cells have $\Theta < \Theta_{coar}$, the corresponding background cell will be flagged for coarsening and its subdivision level will decrease by '1' (unless it has a subdivision level of 0). Herein, Θ_{coar} is a user-specified tolerance and $\Theta_{coar} = 0.4$ is used in this work. When a background cell is marked for refining and coarsening, its neighbours are also checked to ensure the final mesh is '*regularized*'.

Grid adaption creates new cells at each time step (either finer or coarser) and flow data must be allocated. The current RKDG2 scheme facilitates data allocation for the newly created cells. For mesh refinement, consider cell C_{i_c} has multiplied into four finer cells $C_{i_{c1}}$,

$C_{i_{c2}}$, $C_{i_{c3}}$ and $C_{i_{c4}}$ that centre at (x_{c1}, y_{c1}), (x_{c2}, y_{c2}), (x_{c3}, y_{c3}) and (x_{c4}, y_{c4}). The average coefficients for the new cells $\left(C_{i_{cl}}\right)_{l=1,2,3,4}$ are obtained from a direct evaluation of the local planar solution $\mathbf{U}^{\mathbf{h}}(x,y)\big|_{C_{i_c}}$ at points $\left(x_{cl}, y_{cl}\right)_{l=1,2,3,4}$, where $\mathbf{U}^{\mathbf{h}}(x,y)\big|_{C_{i_c}}$ is already available over C_{i_c}. The slope coefficients associated to the local planar solution over $\left(C_{i_{cl}}\right)_{l=1,2,3,4}$, respectively, are obtained by conserving the local slope coefficients of the solution over the parent cell C_{i_c}. Concerning grid coarsening, if four finer cells $\left(C_{i_{cl}}\right)_{l=1,2,3,4}$ are coalesced into a coarser cell C_{i_c}, the coefficients defining the new local solution over C_{i_c} are obtained by aggregating the corresponding elementary values from the four finer cells $\left(C_{i_{cl}}\right)_{l=1,2,3,4}$.

4.5.1 Treatment of the topographic data

Regardless of the mesh on which the flow data are solved, the topography function is firstly discretized, or projected, onto a fine uniform grid, which is either user-selected if an analytical topography shape is available, or otherwise, on the basis of the amount of available field data represented by a DEM. In either case, local planar P^1-projection to the topography is applied to form a *background topographic mesh* (*i.e.*, produced as in Eq. (5) with the associated average and slopes coefficients estimated by applying Eq. (4) to the originally available topographic data). Secondly, after dynamic grid adaptation performs on the *actual computational mesh*, the associated topographic data are collected from the *background topographic mesh*. For instance, the average coefficient at the centre 'P' of a cell 'C_{actual}' belonging to the *actual computational mesh* is obtained by directly evaluating its values from the local planar topography projection at cell 'C_{topo}' that belongs to the *background topographic mesh* and particularly contains 'P'. The slope's coefficients over 'C_{actual}' are produced by conserving (i.e., expanding when 'C_{actual}' > 'C_{topo}') the original topographic

slopes coefficients available at 'C_{topo}'. Evidently, $C_{\text{topo}} = C_{\text{actual}}$ when the *actual* calculation grid has (*i.e.*, locally or globally) the same resolution as the one for *background topographic mesh*.

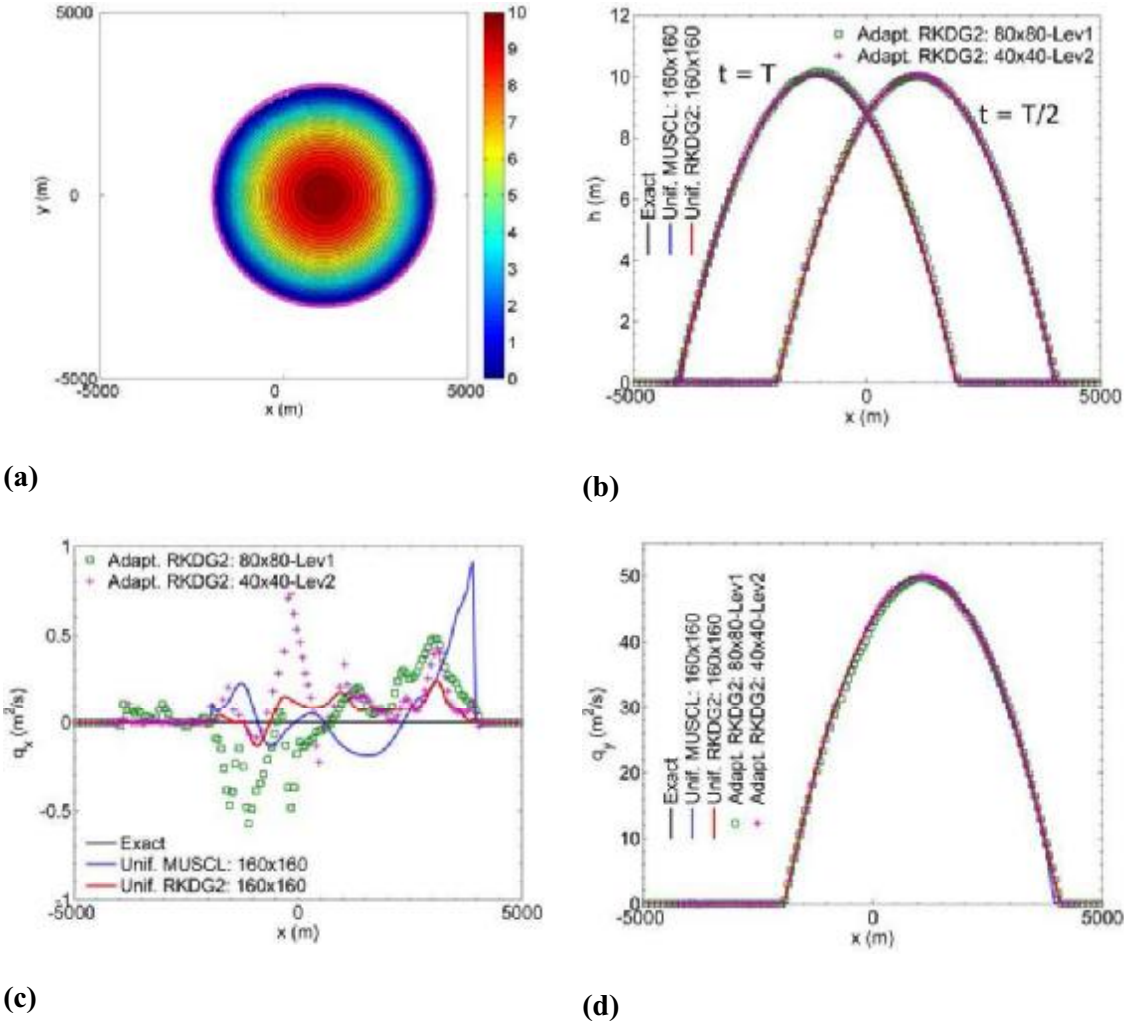


Fig. 3: Oscillatory flow in a parabolic bowl. **(a)** 2D view of the water depth (i.e. 30 contour-lines) produced by the RKDG2 scheme (on the uniform grid 160 × 160) at $t = T/2$, where the surrounding curve represents the analytical shoreline. **(b)** Profiles of water depth along $y = 0$ produced by different numerical schemes at $t = T/2$ and $t = T$. **(c)** Profiles of q_x along $y = 0$ produced by different numerical schemes at $t = T/2$. **(d)** Profiles of q_y along $y = 0$ produced by different numerical schemes at $t = T/2$.

5. Numerical tests and results

Three analytical cases and one laboratory-scaled tsunami test are first investigated to demonstrate the performance of the dynamically *h*-adaptive RKDG2 model by comparing it

to uniform grid based counterparts. The new adaptive RKDG2 model is then applied to reproduce the historical Mapasset dam break. All the simulations are run on a Toshiba Tecra M10-10i laptop with Core 2 Duo T9400 2.53 GHz and 3GB of RAM. Quantitative comparison is evaluated in terms of runtime cost and (when possible) the following relative L^1-Error (%)

$$Error(\%) = \frac{\left\| h^{numerical} - h^{reference} \right\|_{L^1}}{N^{cells} \times \left\| h^{reference} \right\|_{L^1}} \times 100 \tag{28}$$

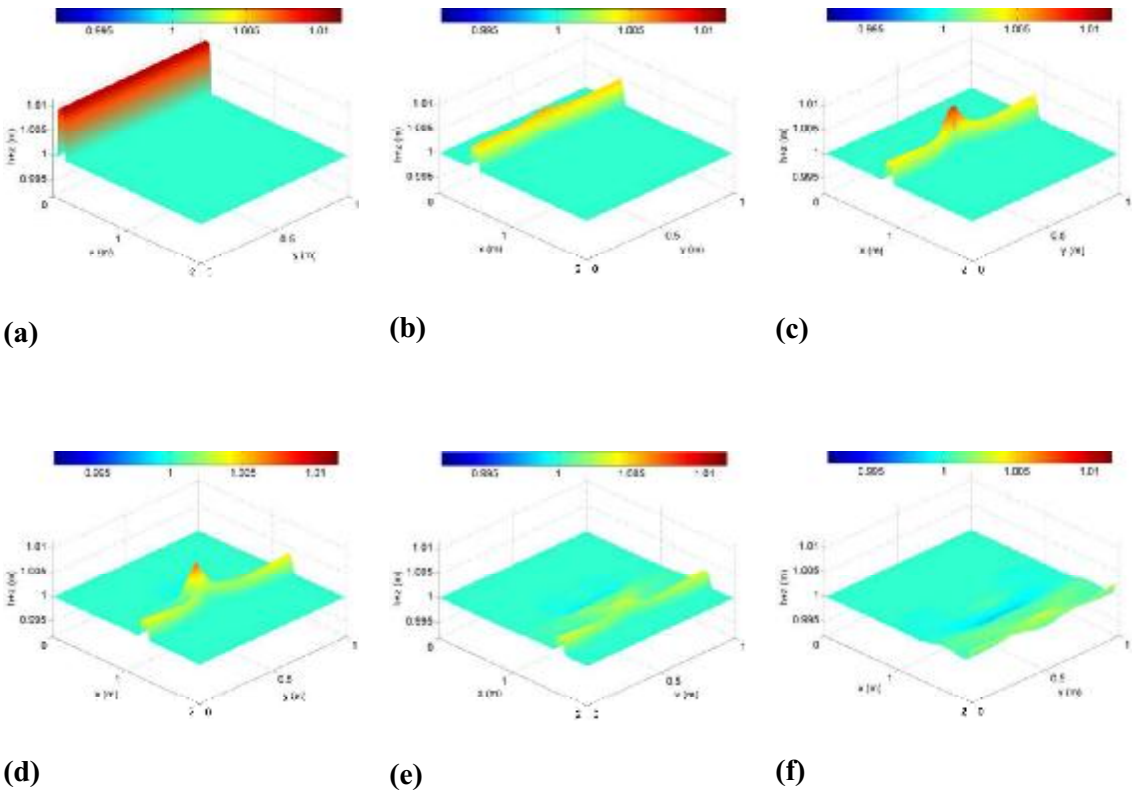


Fig. 4: Small perturbation travelling over an elliptical-shaped topography. 3D water surface produced by the adaptive RKDG2 model (i.e., 40×20-Lev. 2) at **(a)** t = 0s, **(b)** t = 0.12s, **(c)** t = 0.24s, **(d)** t = 0.36s, **(e)** t = 0.48s and **(f)** t = 0.60s.

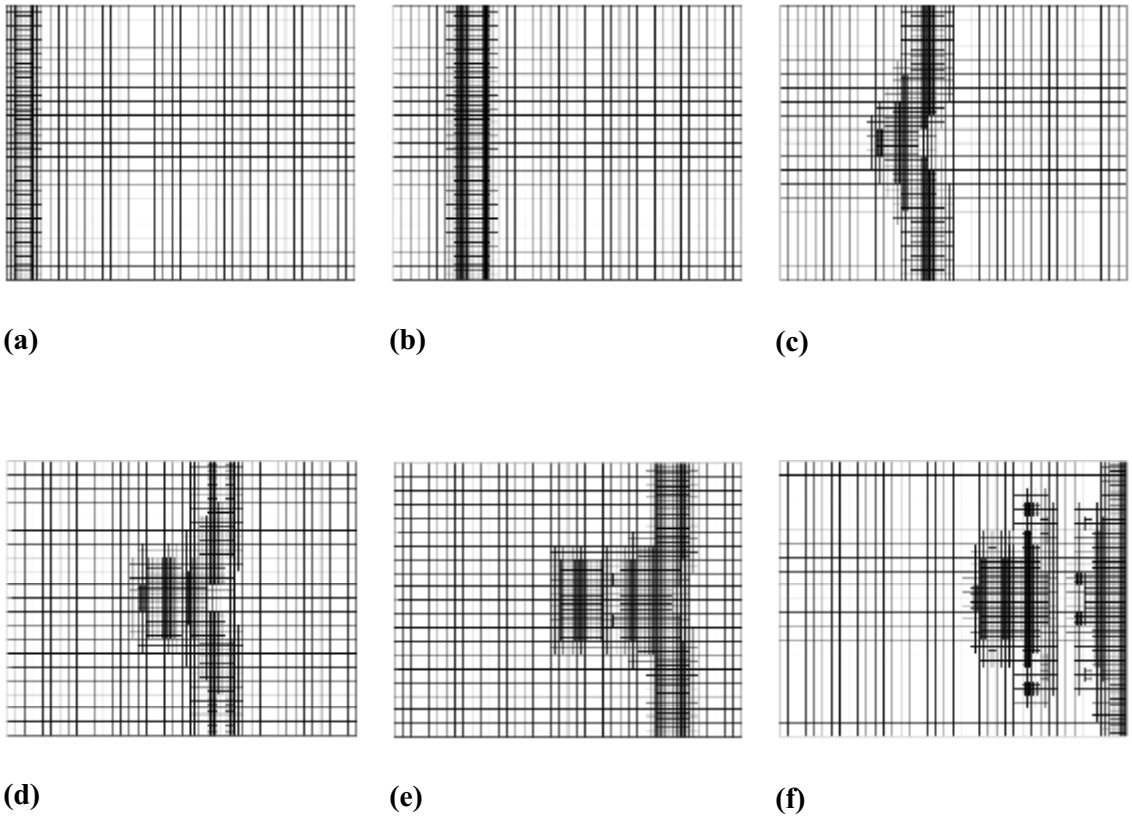
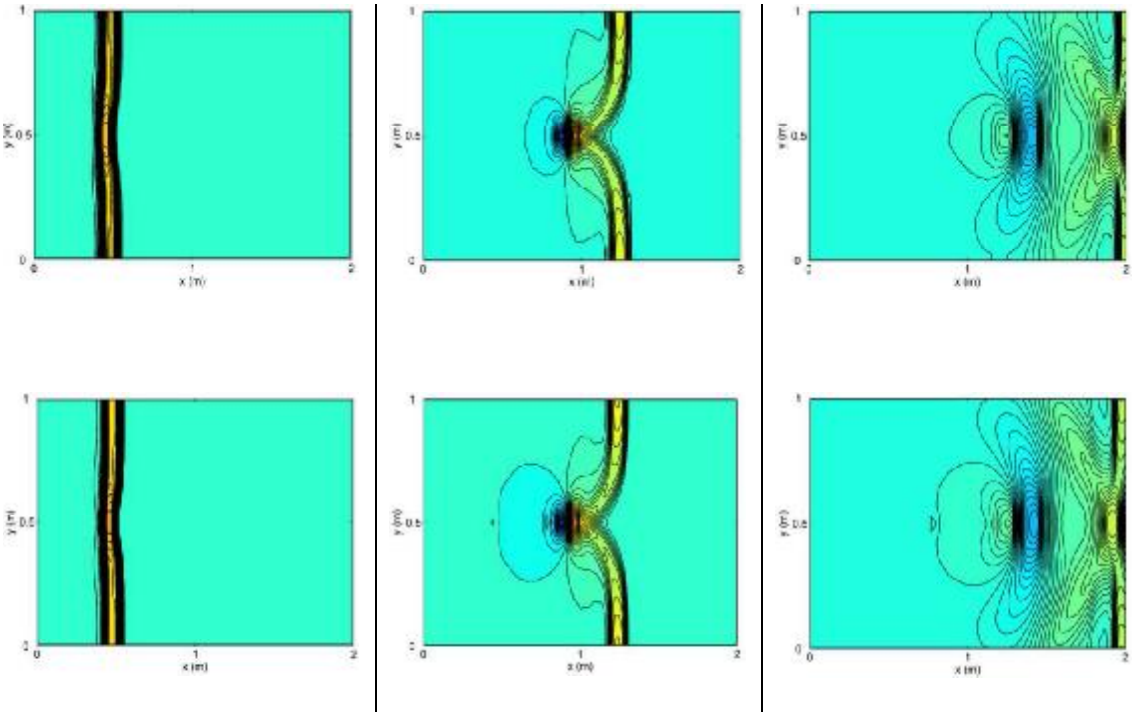


Fig. 5: Small perturbation travelling over an elliptical-shaped topography. Dynamically adaptive mesh (i.e., 40×20—Lev. 2) at **(a)** t = 0s, **(b)** t = 0.12s, **(c)** t = 0.24s, **(d)** t = 0.36s, **(e)** t = 0.48s and **(f)** t = 0.60s.



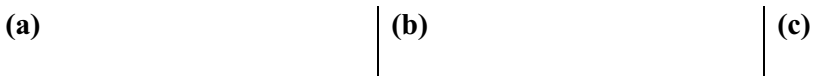


Fig. 6: Small perturbation travelling over an elliptical-shaped topography. RKDG2 free-surface contours obtained on dynamically adaptive mesh (upper panel) and uniform mesh (lower panel) at **(a)** t = 0.12s, **(b)** t = 0.36s and **(c)** t =0.60s.

5.1. Oscillatory flow in a parabolic bowl

The classic 2D analytical test due to Thacker [77] assesses the capability of the numerical model in tracking continually moving wet/dry fronts. The case involves a planar water surface running up and down the beach in a domain with a bowl like topography. Bottom friction is not included and the fluid motion is therefore oscillatory periodically as there is no energy dissipation. In a computational domain Ω = [-5000; +5000] × [-5000; +5000], the bottom topography is defined by

$$z(x,y) = h_0\left(\frac{x^2+y^2}{a^2}\right) \tag{29}$$

where h_0 and a are both constants. The exact solution writes

$$\begin{aligned} &\eta(x,y,t) = \left(h_0 - \frac{1}{2g}B^2\right) + \frac{Bw}{g}\left[-x\cos(wt) + y\sin(wt)\right] \\ &u(t) = B\sin(wt) \quad \text{and} \quad v(t) = B\cos(wt) \end{aligned} \tag{30}$$

where B is a velocity-related constant, $w = \sqrt{\frac{2gh_0}{a^2}}$ is the peak magnitude and $T = \frac{2\pi}{w}$ is the period of flow. Denoting by $K = \frac{wBa^2}{gh_0}$, $X_1 = -\frac{K}{2}\cos(wt)$, $Y_1 = \frac{K}{2}\cos(wt)$ and $C_1 = a^2\left(\frac{B^2}{2gh_0} - 1\right)$, the analytical shoreline is a circle centred at (X_1, Y_1) with the radius $R_1 = \sqrt{X_1^2 + Y_1^2 - C_1}$. Herein, the constants are set to h_0 = 10m, a = 3000m, B = 5ms^{-1} and a simulation is run up to one period (*i.e.*, $t = T$). First, uniform mesh RKDG2 simulation are run using coarse, intermediate and fine uniform resolution grids with 40×40, 80×80 and 160×160 cells, respectively. Two adaptive mesh RKDG2 simulations are then performed by taking the

coarse and intermediate uniform grids as a background mesh and allowing, respectively, two- and one- level of refinement. The associated pattern of the grid was found to refine in the vicinity of the shoreline (*i.e.*, at $t = T/2$ and $t = T$) but are not illustrated for this example as there are no perturbed flow features).

The RKDG2 predictions produced on all the meshes mentioned previously are compared with the output obtained by an alternative MUSCL FV scheme on the fine uniform grid together with the analytical solution in Fig. 3, in which Fig. 3a shows the 2D contour lines of water depth (produced by the RKDG2 scheme on the fine uniform grid). The numerical results produced by different schemes are observed to be qualitatively very similar, for the water depth and non-zero discharge variables as illustrate the plots in Fig. 3b and Fig. 3d. The MUSCL scheme is found to be slightly less accurate in tracing the wet/dry front at $t = T/2$. Remarkably, Fig. 3c presents less accurate approximations to the q_x-discharge profile for all the simulations, where small discrepancies are present at the zone where the water depth is vanishing. This unsatisfactory prediction has also been reported in several previous works for MUSCL-type FV schemes (*e.g.* most recently in [78]). Herein, the RKDG2 scheme appears to calculate less discrepancy than the MUSCL scheme on the same fine uniform mesh (*i.e.*, Fig. 3c). The dynamically adaptive RKDG2 schemes are less successful in replicating the dry zone in view of the spurious momentum that is apparent in Fig. 3c. This is probably caused by the method of enforcing flux conservation along the non-uniform wet/dry moving boundary. This may suggest the need to design a more advanced adaptation technique to better represent moving wet/dry fronts in the frame of an *h*-adaptive RKDG scheme.

Table 1: Oscillatory flow in a parabolic bowl: L^1-Error for free-surface elevation and CPU time costs caused by different simulations at $t = T/2$.

Mesh/scheme		Error (%)	CPU time (s)
	40×40	1.2301E-001	28.5
	80×80	2.9995E-002	197.0

Uni. RKDG2	160×160	7.2021E-003	882.15
Uni. MUSCL	160×160	1.9110E-001	102.14
Adapt. RKDG2	40×40-Lev. 2	3.2934E-001	30.06
	80×80-Lev. 1	2.9006E-001	221.16

Table 1 presents the quantitative analysis for this case, where the L^1-Error and CPU time induced by different simulations are presented. Regarding the results generated by uniform-mesh simulations, despite being much less efficient than the MUSCL scheme for the same fine mesh, the RKDG2 scheme predicts smaller error even on the coarse mesh. Therefore, it may be commented that a uniform mesh based RKDG2 scheme tends to have a better convergence property and, hence, may achieve similar or even better solution accuracy using coarse grids in comparison with a MUSCL scheme. This further supports earlier findings by Shu and co-workers [72,74] who compared systematically RKDG methods with WENO-type finite volume schemes. In our case, both adaptive mesh RKDG2 simulations are less accurate than the RKDG2 computation on the fixed coarse grid. This is likely due to that fact that no perturbed flow feature are considered in this test.

5.2. Propagation of slightly perturbed flow over topography

This synthetic example is credited to LeVeque [48] and is typically used to assess the well-balanced property of a SWEs numerical solver as well as its capability in handling the propagation of a slightly perturbed steady state over a non-flat topography (*e.g.* [17]). The elliptical-shaped bottom topography in a [0; 2] × [0; 1] domain is defined as follows

$$z(x,y) = 0.8\,\mathrm{e}^{-50\left[(x-0.5)^2+(y-0.5)^2\right]} \tag{31}$$

The initial conditions characterize a stationary pool with a water surface elevation of η = 1m except for a 0.01m upward perturbation in the upstream part at 0.05m $\leq x \leq$ 0.15m. RKDG2 simulations are run on different uniform meshes with 40×20, 80×40 and 160×80 cells, respectively. Adaptive grid simulations are performed using either coarse or intermediate

uniform mesh as a background mesh and allowing dynamic local *h*-adaption up to either two or one subdivision level.

Fig. 4 and Fig. 5 depict the RKDG2 predictions at different output times in terms of a 3D view of the free-surface elevations and the corresponding adapted grids for the simulation corresponding to two levels of refinement. As observed in Fig. 4a and Fig. 5a, the initial non-uniform grid is refined at $t = 0$ to represent the sharp surface gradient caused by the original surface perturbation. The dynamically adaptive mesh evolves to capture effectively the details of the moving perturbation with the highest grid resolution while discretizing the unperturbed zone with coarse background mesh, as shown in Fig. 5. Fig. 6 compares the depth contours predicted on the adaptive mesh (upper panel) with those produced on the uniform grid (lower panel), all of which use 30 contour lines between 0.992m and 1.0115m. Apart from certain parts of slightly more diffusive solution in those areas covered by the coarse mesh due to comparatively gentle surface gradient, the adaptive RKDG2 scheme captures most of the small-scale features of the perturbed flow to a similar resolution as those obtained on the fine uniform grid.

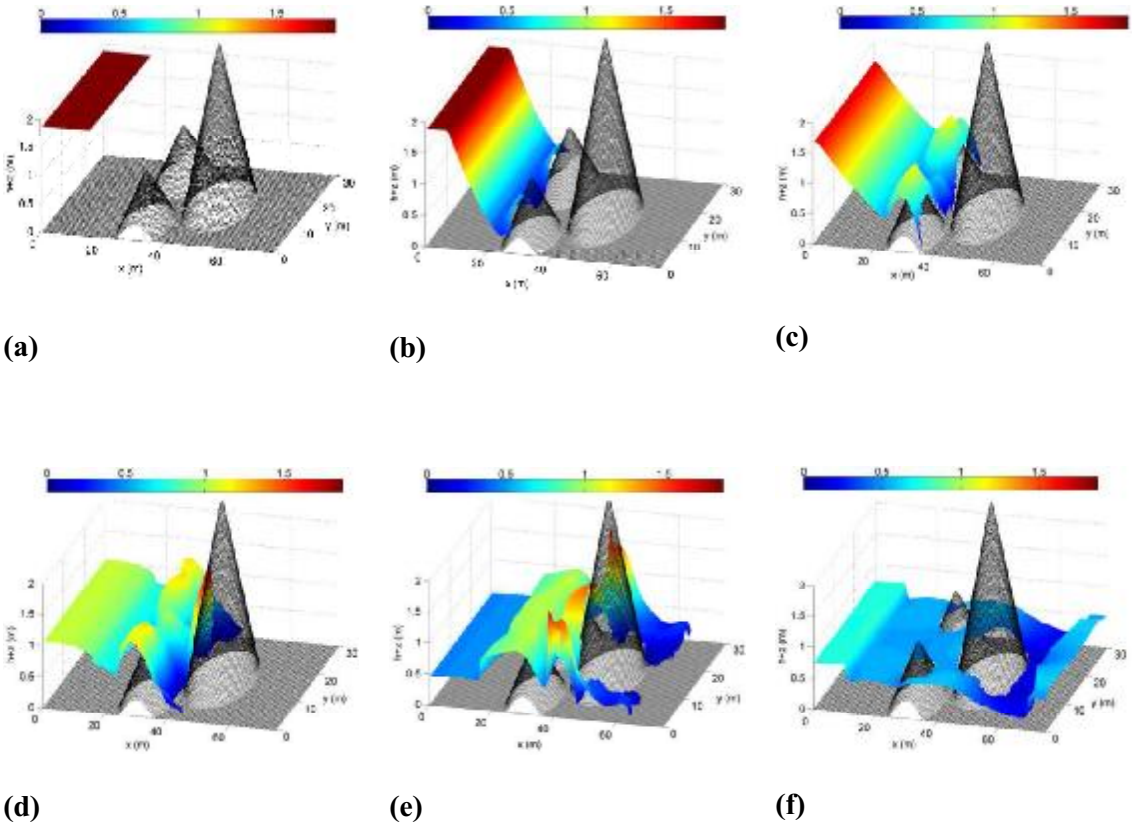
Since no analytical solution is available, the RKDG2 prediction obtained on a 320×160 fine mesh is treated as a reference solution to allow a quantitative assessment. Table 2 lists the errors and the CPU times corresponding to the RKDG2 simulations.

Table 2: Perturbed flow over topography: L^1-Error for free-surface elevation and CPU time costs caused by different simulations at $t = 0.60$s.

Mesh/scheme		**Error (%)**	**CPU time (s)**
Uni. RKDG2	40×20	15.81	4.5
	80×40	11.92	30.2
	160×80	5.39	238.0
Adapt. RKDG2	40×20-Lev. 2	7.44	53.6
	80×40-Lev. 1	6.85	91.6

Both of the adaptive mesh RKDG2 simulations lead to similar numerical error. The simulation on the 40×20 background grid with 2 levels of refinement is found to be less time-

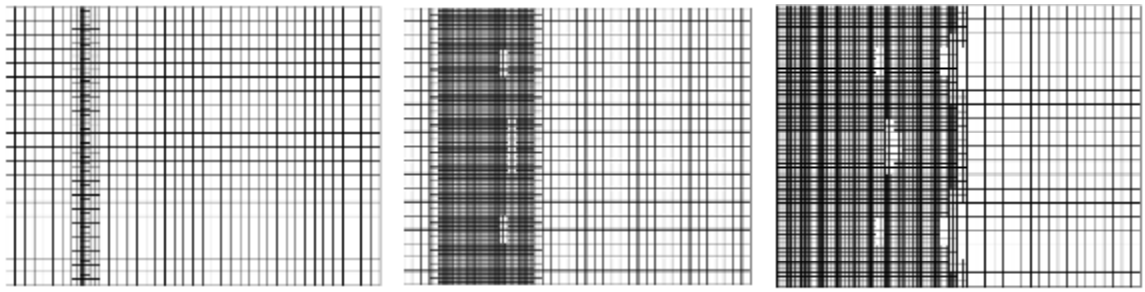
consuming than the one on an 80×40 background grid allowing 1 level refinement. The two runs are, respectively, 4.4 and 2.6 times more efficient than the fine uniform grid RKDG2 simulation.



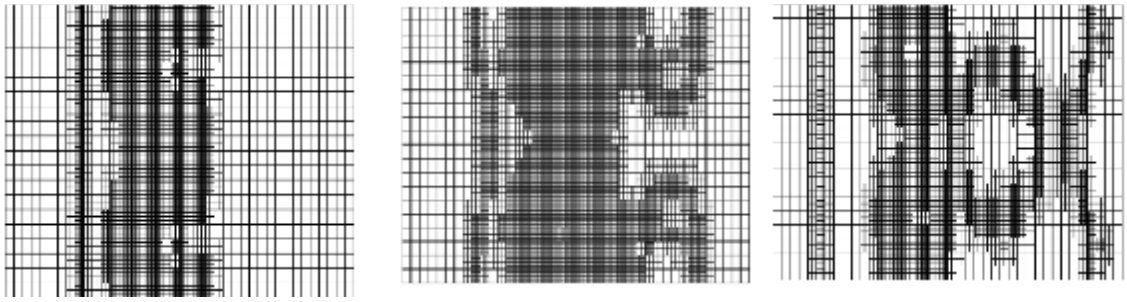
(a) **(b)** **(c)**

(d) **(e)** **(f)**

Fig. 7: Dam-break wave moving over an initial dry floodplain. 3D water surface predicted by the adaptive RKDG2 scheme at **(a)** t = 0s, **(b)** t = 2s, **(c)** t = 4s, **(d)** t = 6s, **(e)** t = 12s and **(f)** t = 24s.

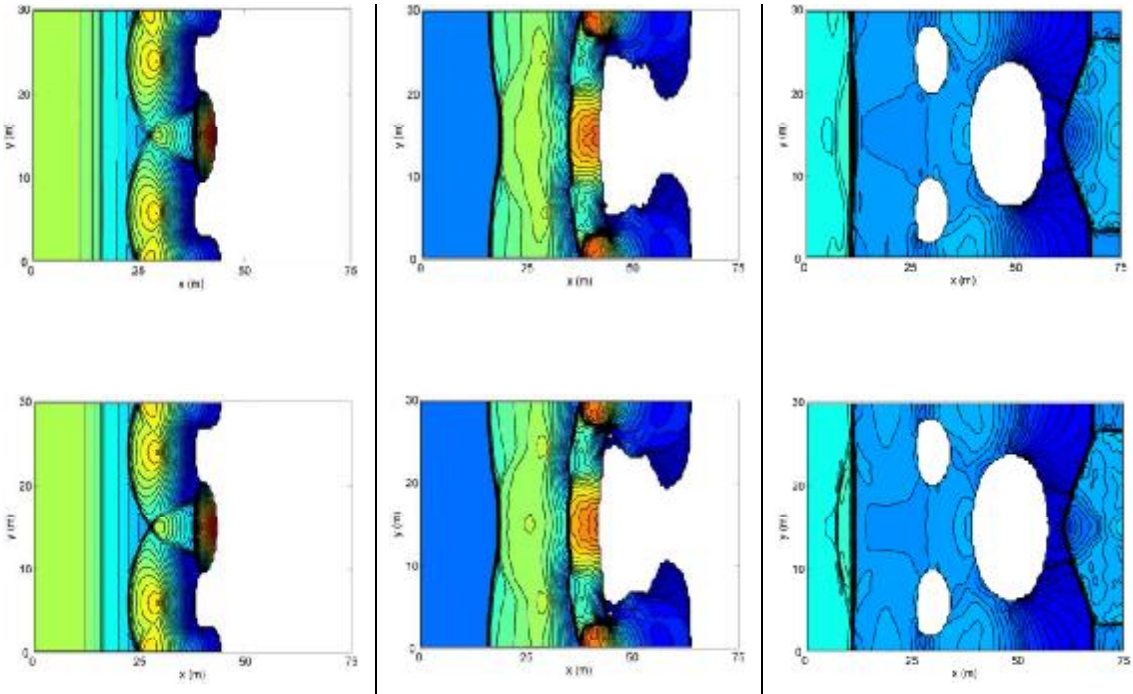


(a) **(b)** **(c)**



(d) **(e)** **(f)**

Fig. 8: Dam-break wave moving over an initial dry floodplain. Associated dynamically adaptive mesh at **(a)** t = 0s, **(b)** t = 2s, **(c)** t = 4s, **(d)** t = 6s, **(e)** t = 12s and **(f)** t = 24s.



(a) **(b)** **(c)**

Fig. 9: Dam-break wave moving over an initial dry floodplain. Comparing the free-surface contours obtained on dynamically adaptive mesh (upper panel) with those produced on uniform mesh (lower panel) at **(a)** t = 2s, **(b)** t = 12s, **(c)** t = 24s.

5.3. Dam-break wave interacting with hillocks

This test of a 2D dam-break wave travelling over an initially dry and rough floodplain with three hillocks has been widely accepted as a standard benchmark to assess the adequacy of a numerical model for realistic flood modelling applications (*e.g.* [9,56]). The mathematical expression of the topography in the [0; 75m] × [0; 30m] close domain is

$$z(x,y) = \max\left[0, 1-\tfrac{1}{8}\sqrt{(x-30)^2+(y-6)^2}, 1-\tfrac{1}{8}\sqrt{(x-30)^2+(y-24)^2}, 3-\tfrac{3}{10}\sqrt{(x-47.5)^2+(y-15)^2}\right] \tag{32}$$

The dam is initially placed at x = 16m to hold a tranquil water body with a surface elevation of 1.875m. The rest of the domain is dry and a global Manning coefficient is set to be n_M = 0.0185. Identical uniform and *h*-adaptive grids as in the previous example are employed to run the RKDG2 simulations. For adaptive meshes, as shown in Fig. 7a and Fig. 8a, initial grid refinement is performed at the dam location to represent the original sharp water surface gradient. Fig. 7 illustrates the 3D view of free-surface elevation at t = 0s, 2s, 4s, 6s, 12s and 24s, predicted by the adaptive RKDG2 solver on the 40×20 background grid allowing 2 levels of refinement. The corresponding dynamically adaptive grids are displayed in Fig. 8. As shown in Fig. 7b and Fig. 7c for the first few seconds after the instantaneous collapse of the dam, while the fast-moving wave front running downstream towards the two small hillocks, a depression wave is formed and travels upstream. The grid dynamically evolves according to the dam-break hydrodynamics, with a refined mesh generated to cover nearly half of the domain. This is due to the steep water surface gradient caused by the dam-break wave, as indicated in Fig. 8b and Fig. 8c. As illustrated in Fig. 7d for t = 6s, while the violent dam-break flow continues on moving downstream and interacts with the three hills, a reflected shock is developed after the depression wave hitting the western boundary wall and propagating downstream. Apart from successfully capturing the advancing wave front and the

complex hydrodynamic structures formed by wave-topography interactions, the mesh is also refined to resolve the reflected shock, as evidenced in Fig. 8d. With the increasing simulation time as demonstrated in Fig. 7e and Fig. 7f for t = 12s and 24s, the dam-break flow grows to be more complex due to further, wave-wave, wave-topography and wave-boundary interactions. Nevertheless, as depicted in Fig. 8e and Fig. 8f, all of the complex flow features are automatically captured by the adaptive grid with high-resolution refinements. In order to reveal the merit of the current dynamic grid system, the water surface contours produced on the fine uniform and the adaptive grid are compared in Fig. 9 at the output times t = 6s, 12s and 24s, where the upper panel presents the adaptive grid result, while the lower panel shows the uniform grid solution. All of the plots use 30 contour lines between 0 and 1.875m. It is evident that the adaptive grid simulation resolves the complex dam-break hydrodynamics to a similar resolution achieved by the fine uniform grid solver.

Table 3: Dam-break wave moving over an initial dry floodplain: L^1-Error for free-surface elevation and CPU time costs caused by different simulations at t = 24s.

Mesh/scheme		**Error(%)**	**CPU time (s)**
Uni. RKDG2	40×20	16.58	6.4
	80×40	12.46	47.3
	160×80	6.68	345.2
Adapt. RKDG2	40×20-Lev. 2	7.85	215.4
	80×40-Lev. 1	8.76	214.2

Similar to the previous case, a reference RKDG2 solution is constructed on a fine uniform grid with 320×160 cells to facilitate quantitative analysis and the results are summarized in Table 3. The numerical accuracy of both adaptive grid based simulations is quite close to that achieved on the fine uniform mesh. In this case, the runtime cost of the two adaptive mesh simulations is very similar and corresponds to a 1.6 times saving compared to the RKDG2 solver on the fine uniform mesh. This is quite reasonable in this case because the grid is expected to be refining and coarsening frequently over a large part of the computational

domain due to the rapid-varying flow hydrodynamics and repeated wetting and drying process. Therefore the computational efficiency gained by the adaptive grid is not outstanding.

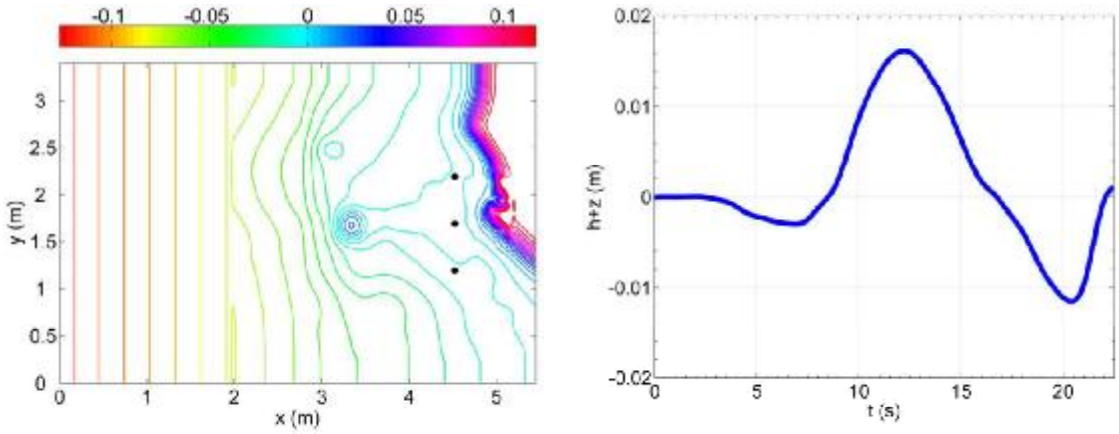
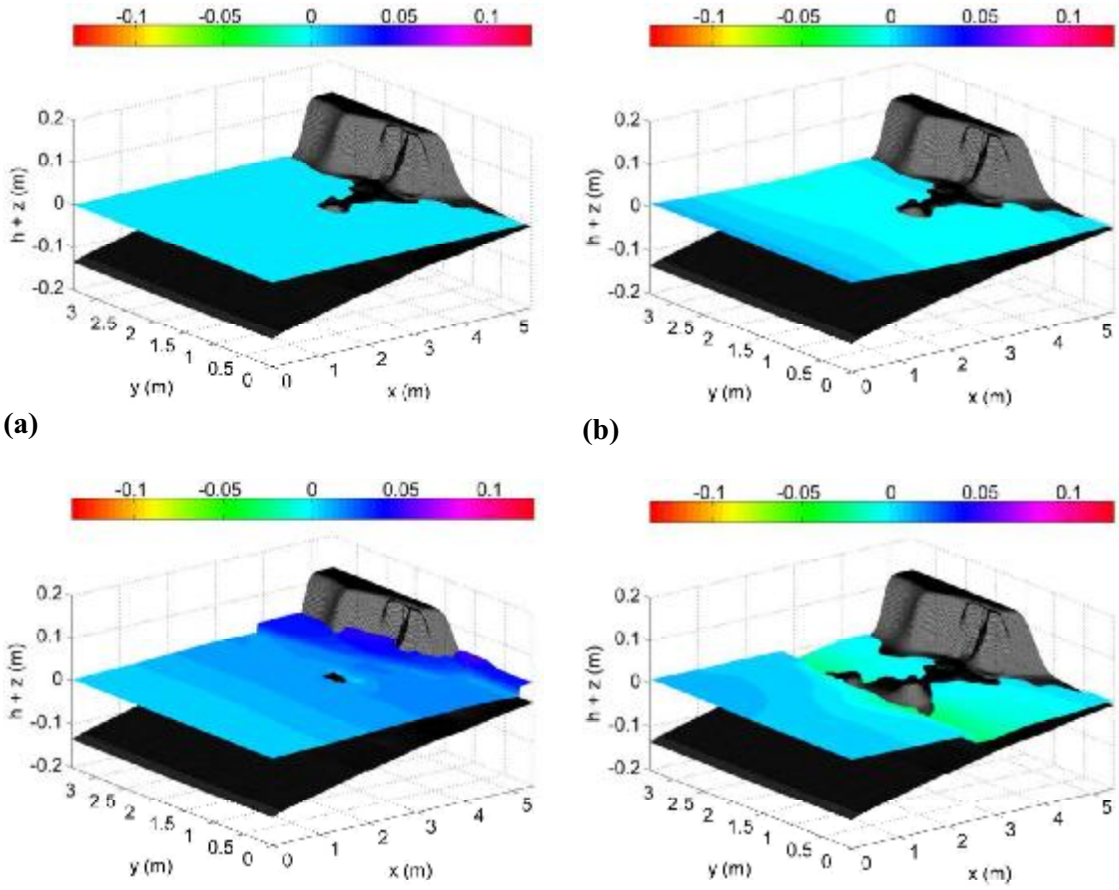
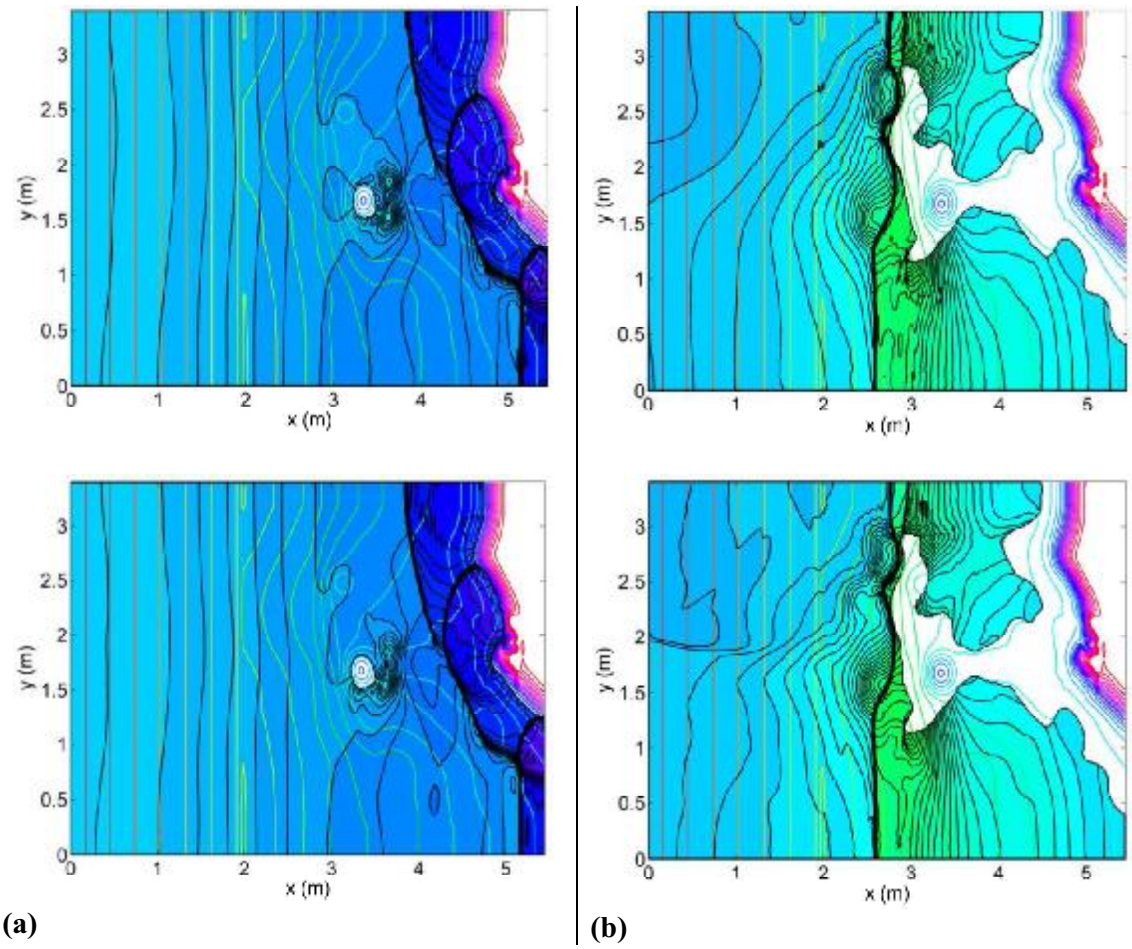


Fig. 10: Tsunami run-up: **(a)** bathymetric contours and **(b)** incoming wave profile through the western boundary.



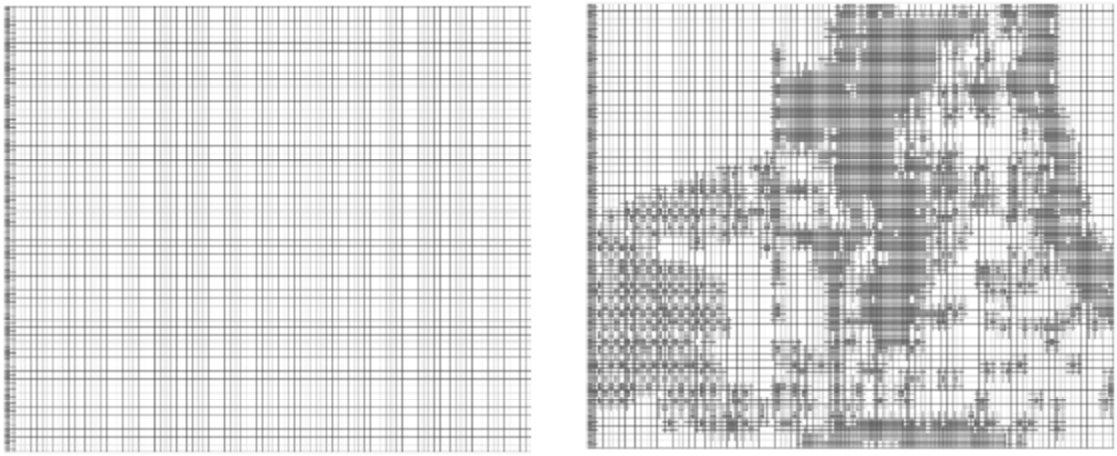
(c) **(d)**

Fig. 11: Tsunami run-up: 3D view of water surface predicted on dynamically adaptive mesh at **(a)** t = 0, **(b)** t = 10s, **(c)** t = 17s and **(d)** t = 25s.



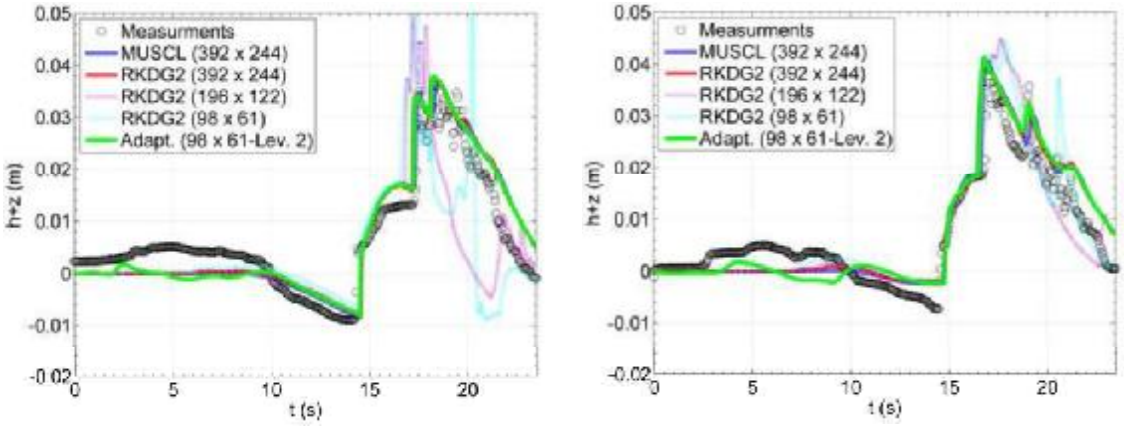
(a) **(b)**

Fig. 12: Tsunami run-up: comparing the free-surface contours obtained on dynamically adaptive mesh (upper panel) with those produced on uniform mesh (lower panel) at **(a)** t = 17s and **(b)** t = 25s.

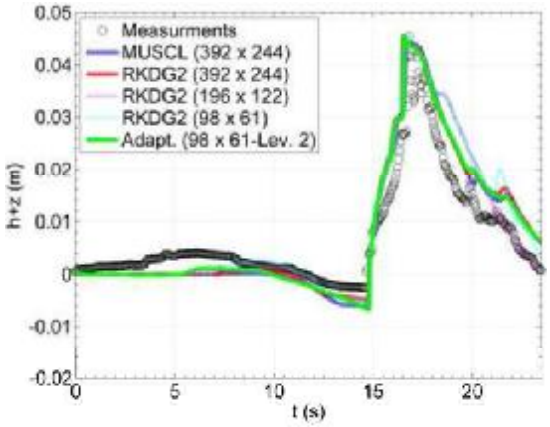


(a) **(b)**

Fig. 13: Tsunami run-up: (a) initial (t = 0), and (b) final (t = 25s) adaptive meshes.



(a) **(b)**



(c)

Fig. 14: Tsunami run-up: comparing numerical time histories of the water surface elevation with experimental data **(a)** G5, **(b)** G7, and **(c)** G9.

5.4. Tsunami run-up onto a complex seashore

The laboratory data for this test were produced from a wave tank experiment and presented at the Third International Workshop on Long Wave Run-up Models in 2004 (http://nctr.pmel.noaa.gov/benchmark/). The 1:400 physical model was built in a wave tank to represent the coastal bathymetry near Monai in Japan. This region suffered severe flood inundation due to the Okushiri tsunami in 1993. The tsunami triggered a maximum run-up of 31.7m observed at Monai valley, which opens onto a small beach. This run-up was not uniform along the coast and is considered to be caused by particular topographic effects.

The 4.588m × 3.402m wave tank was installed with a bathymetric profile described in Fig. 10a, which is represented by 392 × 244 data points on a uniform grid. The initial motionless water body submerges the majority of the domain as shown in Fig. 11a. Along the western boundary at $x = 0$, an incoming wave profile, as described in Fig. 10b was imposed for 22.5s. As indicated in Fig 10a, three gauge points, denoted “G5”, “G7” and “G9”, were used to measure the temporal change of water surface elevation. The points were located at (4.521m; 1.196m), (4.521m; 1.696m) and (4.521m; 2.196m), respectively. The Manning coefficient is assumed to be 0.01 throughout the entire computational domain. The initial non-uniform grid for the 25s simulation is generated on an 98 × 61 coarse background mesh with a two-level refinement at the western boundary inlet. For comparison purposes, a simulation is run on a fine uniform gird with 392 × 244 cells, which has the same resolution as the available topographic data. Two other RKDG2 simulations are also carried out on two coarser uniform meshes with 196 × 122 cells and 98 × 61 cells, respectively.

Fig. 11 shows 3D views of the free water surface of the adaptive grid based RKDG2 prediction at $t = 0$s, 10s, 17s and 25s, respectively. After a small tidal retreat at the beginning as shown in Fig. 11a and Fig. 11b, a tsunami wave with a peak of 0.016m rushes into the

domain and onto the shore. The incident wave interacts with the island and creates wave reflections and diffractions. The reflected and diffracted wave reaches the coast and inundates a large area, as shown in Fig. 11c. After reaching the maximum run-up, the wave retreats gradually as presented in Fig. 11d for t = 25s. The contours of the water elevation at t = 17s and t = 25s achieved by the RKDG2 solvers on the dynamically adaptive and the fine uniform grids are plotted side by side in Fig. 12. Fig. 13 presents the corresponding initial and final dynamically adaptive grid. Compared with the uniform grid based RKDG2 outputs, the results obtained on the adaptive grid are able to locate correctly the wet/dry front and capture most of the detailed wave structures as compared to the fine uniform grid-based model. The number of cells used in the adaptive grid based simulation varies between 7076 and 41843, which is clearly much less than the 95648 cells required by the fine uniform grid simulation.

Table 4: Tsunami run-up: CPU time costs.

Mesh/scheme	**Uni. RKDG2**			**Uni. MUSCL**	**Adapt. RKDG2**
	98 × 61	196 × 122	392 × 244	392 × 244	98 × 61-Lev. 2
CPU time (hrs)	0.25	3.7	8.6	0.75	1.52

Fig. 14 compares the time-histories of water surface elevation predicted by the adaptive mesh RKDG2 solver with the experimental data at the three gauge points. In the same figure is also plotted the outcome of three other RKDG2 simulations on different uniform grids and the results produced by a MUSCL scheme on the fine uniform grid. The numerical predictions resulting from the adaptive and the two fine uniform mesh simulations agree satisfactorily with the experimental measurements.

For this specific case, the wave characteristics are highly affected by the steepness of the domain topography and the issue of scale is found to be dominant. The two RKDG2 simulations on the coarser uniform grids dramatically failed to reproduce the time histories of water surface elevation at gauge points G5 and G7 (i.e., in Fig. 14a and Fig. 14b, where

spikes exceeded 0.05m). This is because the topography-associated coefficients over the two coarse uniform meshes are collected from the refined topographic mesh (as explained in Subsection 4.5.1). While this practice is helpful to avoid losing the scale data, it seems that it may have a local side effect on the *C*-property, which is exactly fulfilled (i.e., in a spectral sense [36,38]) when the *actual mesh* and the *topographic mesh* overlap. This explains why the unperturbed zone at $t < 10$s is better captured with the RKDG2 scheme on the uniform topographic grid (i.e., 392 × 244 cells) than with the RKDG2 model on the dynamically adaptive grid (although no substantial disturbances to the overall solution are observed for this case). In practice, this side effect will be eliminated for the dynamically adaptive RKDG2 scheme in the limit of the portions of the actual grid with the highest resolution coinciding with the topographic grid. It can also be commented, supported further by previous examples, that the side effect of the *C*-property is minor at zones where the mesh is coarse but the topography is mild.

This in turn indicates that the adaptive grid based simulation produces results with similar resolution to the fine uniform grid outcomes. Concerning the computational efficiency, the use of a dynamic *h*-adaptive grid based RKDG2 solver enables up to 6.7 times of saving in computational cost compared with its uniform mesh based counterpart, as indicated in Table 4. Despite the significant saving in the RKDG2 model due to mesh adaptation, the refined FV MUSCL simulation remains 5-, 11.5- and 2-times more time-efficient than the RKDG2 schemes on the intermediate, fine and dynamically adaptive meshes, respectively.

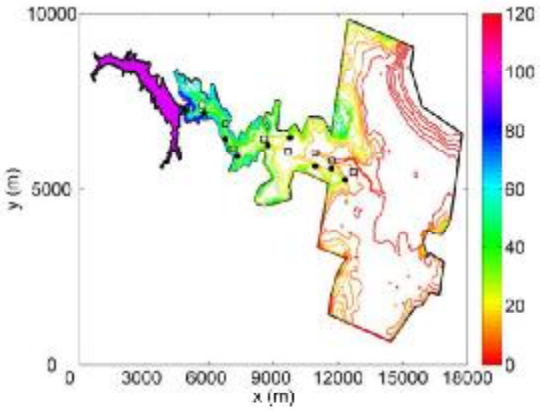


Fig. 15: Malpasset dam-break: computational domain and locations of the Gauge "**G**" and Police survey "**P**" points, which are marked by squares and circles, respectively.

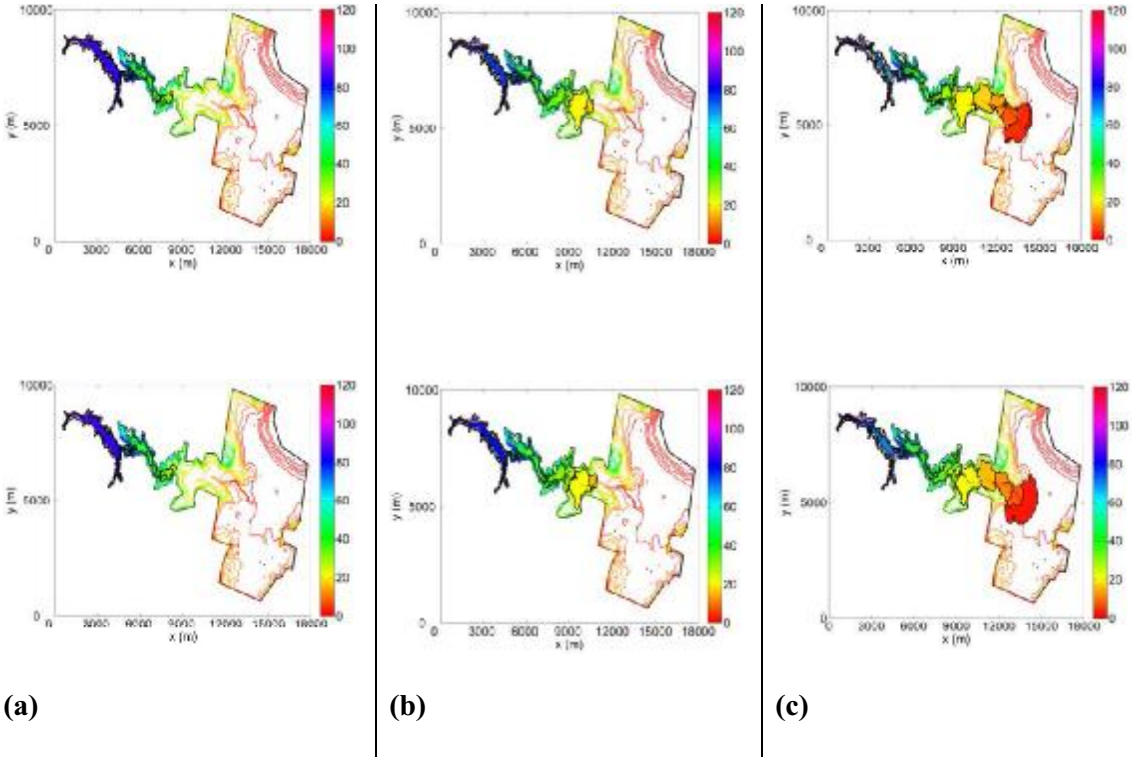
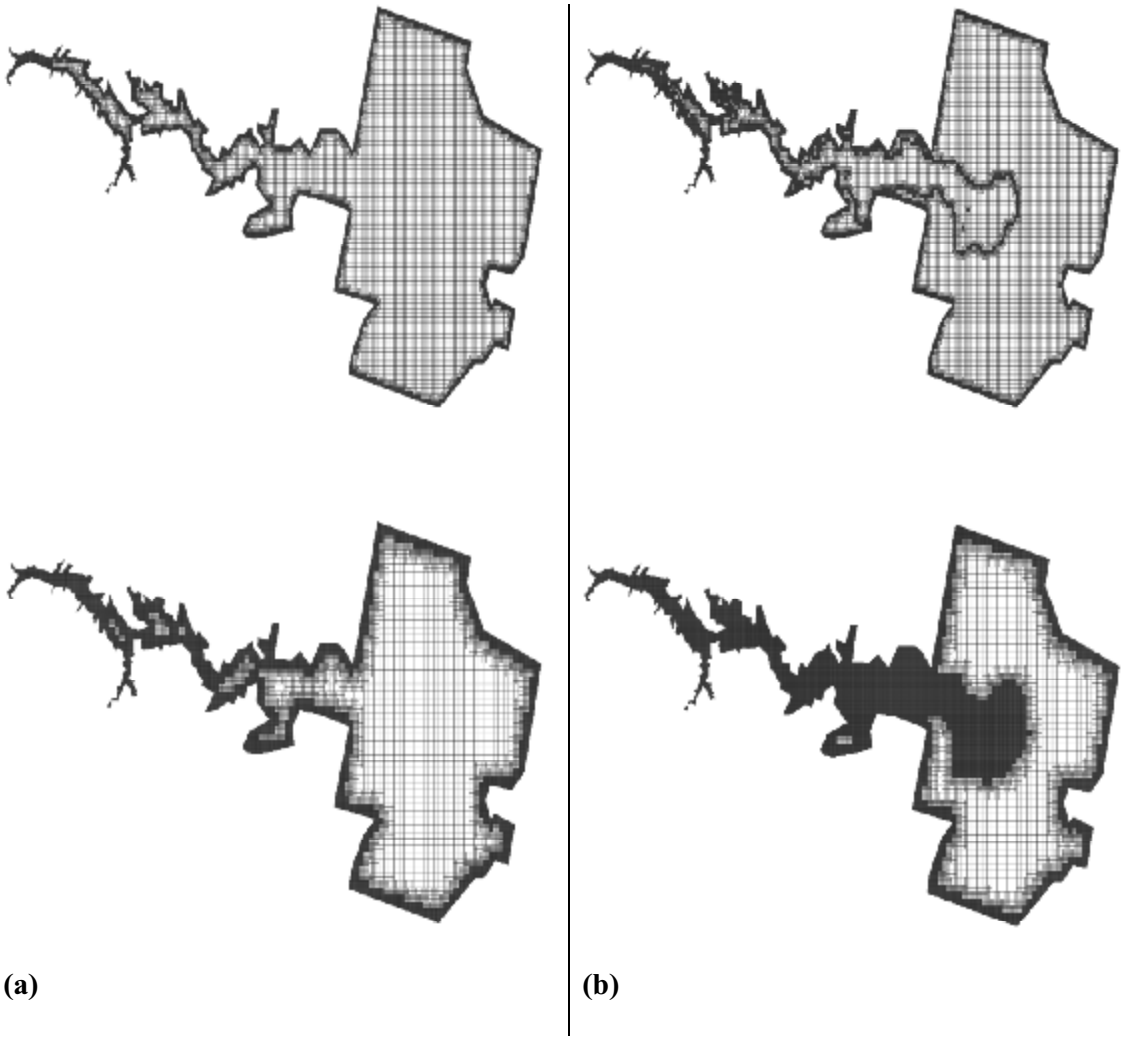
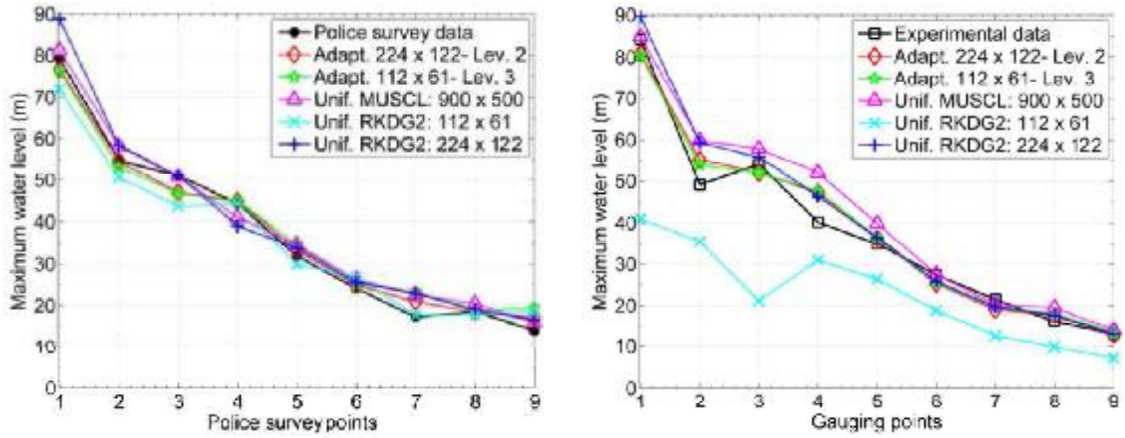


Fig. 16: Malpasset dam-break: flood maps predicted on two adaptive grids (**upper panel**: 224×122 background mesh with 2 levels of subdivision; **lower panel**: 112×61 background mesh with 3 levels of subdivision) at **(a)** t = 10 min, **(b)** t = 20 min, **(c)** t = 40 min.



(a) **(b)**

Fig. 17: Malpasset dam-break: dynamically adaptive grids (**upper panel**: 224×122 background mesh with 2 levels of subdivision; **lower panel**: 112×61 background mesh with 3 levels of subdivision) at **(a)** t = 0min and **(b)** t = 40min.



(a) **(b)**

Fig. 18: Malpasset dam-break: comparing numerical predictions with measurements **(a)** Maximum water level at police survey points and **(b)** Maximum water level at experimental gauge points.

5.5. Malpasset dam break

The Malpasset dam on the Reyran River in southern France breached in December 1959 and created a 40m high wall of water moving rapidly downstream with a speed of 70km/h. The dam-break wave destroyed villages and other obstacles on its way. More than 400 victims were reported and the infrastructure damages were amounted to a total of US$68 million. After the disastrous accident, a police survey was undertaken to estimate the maximum water level at certain locations ("**P**") by tracing water marks. Moreover, in 1964, laboratory studies were also carried out at Electicité de France (EDF) to measure the arrival time and maximum water level at a number of gauge points ("**G**") close to the police survey locations. Police survey points and experimental gauges are indicated in Fig. 15 by filled circles and hollow squares.

The problem domain is shown in Fig. 15 and the dam is assumed to be a straight line between (4701m; 7143m) and (4655m; 7392m). The free-surface level (*i.e.*, $\eta = h + z$) upstream of the dam is set to 100m and the rest of the valley is initially dry. The Manning coefficient is chosen to be 0.033 over the whole domain. The topography data, available originally on an unstructured triangular mesh, have been interpolated to form a uniform mesh of 20m × 20m resolution, resulting in a 900×500 raster grid for storing the topography data. Two dynamically adaptive grid based simulations are set up. The first one is based on a 224 × 122 background grid and allows 2 levels of refinement while the second one relates to a background mesh of 112×61 cells and allows up to 3 levels of subdivision. Both meshes are initially refined to represent the dam and domain boundaries. Due to strong domain irregularity, it was also found important to refine the mesh along external boundaries. Cells outside the boundaries are excluded from the flow calculations. The initial number of cells

(excluding those outside the boundaries) on the two non-uniform grids is 29666 and 43601, respectively.

Simulations are executed for 40min after the dam breaks on the two adaptive grids as mentioned previously. Fig. 16 illustrates the flood maps obtained on both meshes at 10min, 20min and 40min. Both sets of flood maps represent the catastrophic dam-break wave rushing along the narrow valley to the open floodplain and agree closely with those reported in the literature (*e.g.* [13]). Fig. 17 displays the dynamically adaptive grids at $t = 0$ and $t =$ 40min. The grids obviously capture the moving wave front effectively as indicated in Fig. 16, which is as expected. Fig. 18 compares the simulated and measured maximum free-surface levels collected at the gauges and the police survey points, respectively. Also plotted are the predictions produced by a MUSCL model [51] on a refined uniform mesh of 900 × 500 cells and the RKDG2 model on much coarser uniform meshes (i.e., consisting of 112×61 cells and 224×122 cells—relative to the two considered background meshes). A quantitative error analysis with respect to the reference data can be found in Table 5, where the resulting runtime costs are also listed. All of the simulations are able to reproduce maximum water depths that closely agree with the laboratory measurement and police survey, except an obvious underestimation delivered by the RKDG2 solver on the very coarse uniform mesh with 112×61 cells (as indicated in Fig. 18b). It is interesting to note for this case that the dynamically *h*-adaptive RKDG2 simulations generates results that agree most closely with the reference data. Since both adaptive grid simulations predict nearly similar results, it seems to suggest that the adaptive grid based prediction is not sensitive to the resolution of the background grid, which is obviously a desirable property. Generally, the present adaptive grid based RKDG2 predictions also compare reasonably well with those numerical predictions provided by other finite volume Godunov-type schemes documented in literature (*e.g.*, [13,27,65,71,78]).

In terms of runtime cost, the adaptive mesh based simulations consume 8.1hrs and 6.3hrs for the first and second mesh setups, respectively, where the number of computational cells increases to a maximum of 37103 and 65240. Both of these numbers are significantly smaller than 900 × 500 cells if a uniform grid based RKDG2 simulation is run to a similar resolution, which will result in significant savings in terms of computational cost.

Table 5: Malpasset dam-break: computational errors evaluated against the experimental data and CPU costs, resulting from different simulations.

Mesh/scheme		**Error(%) at (P)**	**L^1-Error(%) at (G)**	**CPU time (hrs)**
Uni. RKDG2	112×61	0.976	4.468	0.0036
	224×122	1.008	0.985	0.063
Uni. MUSCL	900×500	0.760	1.237	0.72
Adapt. RKDG2	112×61-Lev. 3	0.712	0.842	6.3
	224×122-Lev.2	0.543	0.846	8.1

6. Summary and conclusions

This work has introduced dynamic *h*-adaptation to an RKDG2 shallow flow model, which is based on local planar solutions to the depth-averaged SWEs on a quadrilateral grid system. The computational grid is essentially non-uniform and involves quadrilateral cells, or panels, of different sizes but regularized in a manner that ensures neighbouring background cells do not differ by more than one level of refinement. The explicit RKDG2 scheme, which evolves in time the coefficients defining a planar solution over each quadrilateral element, while exchanging inter-elemental spatial fluxes, is formulated to comply with the non-uniform grid system. Meanwhile, advanced numerical techniques relevant to practical flood simulation have been implemented (i.e., for controlling slope coefficients, handling complex domain topography and wetting and drying). Dynamic grid adaptation is performed according to an indicator based on the magnitude of the local gradients of the RKDG2 free-surface solution. User-determined constraints are employed to automate grid refinement and coarsening. Three analytical benchmark tests are first employed to systematically investigate the performance of

the dynamically *h*-adaptive RKDG2 model against its uniform mesh equivalent. The new RKDG2 adaptive mesh model is then applied, and further assessed, for reproducing two documented flood tests.

Compared with the corresponding uniform mesh based RKDG2 solver, the dynamically adaptive grid based RKDG2 model is able to deliver numerical predictions of comparable resolution and capture equally well those small-scale flow features without introducing noticeable distortions or noises at the mesh interface, but at much less computational cost. Furthermore, the RKDG2 solutions generally have a better convergence property and higher numerical accuracy compared with those predicted by the traditional finite volume scheme. This implies that numerical solutions with similar accuracy may be obtained on a much coarser mesh. Overall, with the enhanced computational efficiency provided by the dynamically adaptive grid, the new RKDG2 model becomes much more likely applicable to real-scale shallow flow modelling. Local time stepping is currently being investigated to further improve the performance of the current adaptive grid based RKDG2 model.

Acknowledgement

This work is funded by the UK Engineering and Physical Sciences Research Council (EPSRC) through grant: EP/F030177/1. The authors thank the reviewers for the valuable comments and insightful suggestions which led to a significant improvement in the substance of the paper. Lastly, the authors wish to thank Dr. Christopher Keylock for proofreading the paper.

References

1. Aizinger V, Dawson C. A discontinuous Galerkin method for two-dimensional flow and transport in shallow water. Adv Wat Res 2002; 25(1):67–84.
2. Ambati VR, Bokhove O. Space–time discontinuous Galerkin discretization of rotating shallow water equations. J Comput Phys 2007; 225(2):1233–1261
3. Audusse E, Bouchut F, Bristeau MO, Klein R, Perthame B. A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. SIAM J Sci Comput 2004; 25(6):2050–2065.
4. Aureli F, Maranzoni A, Mignosa P, Ziveri C. A weighted surface-depth gradient method for the numerical integration of the 2D shallow water equations with topography. Adv Wat Res 2008; 31(7):962–974.
5. Bader M, Bock C, Schwaiger J, Scaba B. Dynamically adaptive simulations with minimal memory requirement—solving the shallow water equation using sierpinski curves. Siam J Sci Comput 2010; 32(1):212–228.
6. Begnudelli L, Sanders BF, Bradford SF. Adaptive Godunov-Based Model for Flood Simulation. J Hydr Engrg 2008; 134(6):714–725.
7. Begnudelli L, Sanders BF. Conservative wetting and drying methodology for quadrilateral grid finite-volume models. J. Hydr Engrg 2007; 133(3):312–322.
8. Begnudelli L, Sanders BF. Simulation of the St. Francis dam-break flood. J Engrg Mech 2007; 133(11):1200–1212.
9. Benkhaldoun F, Elmahi I, Seaid M. A new finite volume method for flux-gradient and sour-term balancing in shallow water equations. Comput Meth App Mech Eng 2010; 199:3324–3335.
10. Bernard PE, Remacle JF, Legat V. Boundary discretization for high order discontinuous

Galerkin computations of tidal flows around shallow water islands. Int J Numer Meth Fluids 2009; 59(5):535–557.

11. Bokhove O. Flooding and drying in discontinuous Galerkin finite element method for shallow water flows. J Sci Comput 2005; 22-23(1):47–82.
12. Bradford SF, Sanders BF. Finite volume model for shallow water flooding of arbitrary topography. J Hydr Engrg 2002; 128(3):289–298.
13. Brufau P, García-Navarro P, Vázquez-Cendón ME. Zero mass error using unsteady wetting-drying conditions in shallow flows over dry irregular topography. Int J Numer Meth Fluids 2004; 45(10):1047–1082.
14. Bunya S, Kubatko EJ, Westerink JJ, Dawson C. A wetting and drying treatment for the Runge-Kutta discontinuous Galerkin solution to the shallow water equations. Comput Meth Appl Mech Engrg 2009; 198(17-20):1548–1562.
15. Burguete J, García-Navarro P, Murillo J, García-Palacín I. Analysis of the friction term in the one-dimensional shallow water model. J Hydr Engrg 2007; 133(9):1048–1063.
16. Burguete J, García-Navarro P, Murillo J. Friction term discretization and limitation to preserve stability and conservation in the 1D shallow-water model: Application to unsteady irrigation and river flow. Int J Numer Meth Fluids 2008; 58(4):403–425.
17. Caleffi V, Valiani A, Bernini A. Fourth-order balanced source term treatment in central WENO schemes for shallow water equations. J Comput Phys 2007; 218: 228–245.
18. Casulli V. A high-resolution wetting and drying algorithm for free-surface hydrodynamics. Int J Numer Meth Fluids 2009; 60(4):391–408.
19. Cockburn B, Shu CW. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. J Sci Comput 2001; 16(3):173–261.
20. Cockburn B, Shu CW. The Runge-Kutta local projection P1-discontinuous Galerkin method for scalar conservation laws. RAIRO Modél Math Anal Numér 1991; 25(3):337–

361.

21. Ern A, Piperno S, Djadel K. A well-balanced Runge-Kutta discontinuous Galerkin method for the shallow-water equations with flooding and drying. Int J Numer Meth Fluids 2008; 58(1):1–25.
22. Eskilsson C, Sherwin SJ. A triangular spectral/hp discontinuous Galerkin method for modelling 2D shallow water equations. Int J Numer Meth Fluids 2004; 45(6):605–623.
23. Fagherazzi S, Rasetarinera P, Hussaini YM, Furbish DJ. Numerical solution of the dam-break problem with a discontinuous Galerkin method. J Hydr Engrg 2004; 130(6):532–539.
24. Gallardo JM, Parés C, Castro M. On a well-balanced high-order finite volume scheme for shallow water equations with topography and dry areas. J Comput Phys 2007; 227(1):574–601.
25. Gallegos HA, Schubert JE, Sanders BF. Two-dimensional, high-resolution modeling of urban dam-break flooding: A case study of Baldwin Hills, California. Avd Wat Res 2009; 32(8):1323–1335.
26. George DL, LeVeque R. Finite volume methods and adaptive refinement for global tsunami propagation and local inundation. Science of Tsunami Hazards 2006; 24(5):319–328.
27. George DL. Adaptive finite volume methods with well-balanced Riemann solvers for modeling floods in rugged terrain: Application to the Malpasset dam-break flood (France, 1959). Int J Numer Meth Fluids 2011; DOI: 10.1002/fld.2298.
28. Giraldo FX, Restelli M. High-order semi-implicit time-integrators for a triangular discontinuous Galerkin oceanic shallow water model. Int J Numer Meth Fluids 2010; 63(9):1077–1102.
29. Giraldo FX, Warburton T. A high-order triangular discontinuous Galerkin oceanic

shallow water model. Int J Numer Meth Fluids 2008; 56(7):899–925.

30. Giraldo FX. High-order triangle-based discontinuous Galerkin methods for hyperbolic equations on a rotating sphere. J Comput Phys 2006; 214(2):447–465.
31. Gourgue O, Comblen R, Lambrechts J, Kärnä T, Legat V, Deleersnijder E. A flux-limiting wetting–drying for finite element shallow-water models, with application to Scheldt Estuary. Adv Wat Res 2009; 32(12):1726–1739.
32. Greaves DM, Borthwick AGL: Hierarchical tree-based finite element mesh generation. Int J Numer Meth Engrg 1999; 45:447–471.
33. Guinot V. Godunov-type schemes: an introduction for engineers. Elsevier: Amsterdam, 2003.
34. Hunter NM, Bates PD, Neelz S, Pender G, Villanueva I, Wright NG, et al. Benchmarking 2D hydraulic models for urban flood simulations. Proc Inst Civ Eng-Wat Mgmt 2008; 161(1):13–30.
35. Kärnä T, de Brye B, Gourgue O, Lambrechts J, Comblen R, Legat V, Deleersnijder E. A fully implicit wetting–drying method for DG-FEM shallow water models, with an application to the Scheldt Estuary. Comput Methods Appl Mech Engrg 2011; 200(5-8):509–524.
36. Kesserwani G, Liang Q, Vazquez J, Mosé R. Well-balancing issues related to the RKDG2 scheme for the shallow water equations. Int J Numer Meth Fluids 2010; 62(4)428–448.
37. Kesserwani G, Liang Q. A discontinuous Galerkin algorithm for the two-dimensional shallow water equations. Comput Meth App Meth Eng 2010; 199(49-52):3356–3368.
38. Kesserwani G, Liang Q. Locally limited and fully conserved RKDG2 shallow water solutions with wetting and drying. J Sci Comput 2011; DOI 10.1007/s10915-011-9476-4.
39. Krámer T, Józsa J. Solution-adaptivity in modelling complex shallow flows. Comput

Fluids 2007; 36(3):562-577.

40. Krivodonova L, Xin J, Remacle JF, Chevaugeon N, Flaherty JE. Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws. Appl Numer Math 2004; 48(3-4):323–338.
41. Krivodonova L. Limiters for high-order discontinuous Galerkin methods. J Comput Phys 2007; 226(1):879–896
42. Kubatko EJ, Bunya S, Dawson C, Westerink JJ, Mirabito C. A performance comparison of continuous and discontinuous finite element shallow water models. J Sci Comput 2009; 40(1-3):315-339.
43. Kubatko EJ, Bunya S, Dawson C, Westerink JJ. Dynamic p-adaptive Runge–Kutta discontinuous Galerkin methods for the shallow water equations. Comput Meth App Mech Engrg 2009; 198(21-26):1766–1774.
44. Kubatko EJ, Westerink JJ, Dawson C. Hp Discontinuous Galerkin methods for advection dominated problems in shallow water flow. Comput Meth App Meth Eng 2006; 96(1-3):437–451.
45. Xu Y, Shu CW. Local discontinuous Galerkin methods for high-order time dependent partial differentiable equations. Commun Comput Phys 2010; 7(1):1–46.
46. Lamby P., Müller S., Stiriba Y., Solution of shallow water equations using fully adaptive multiscale schemes. Int J Numer Meth Fluids 2005; 49:417–437.
47. Lee SH, Wright NG. A simple and efficient solution of the shallow water equations with source terms. Int J Numer Meth Fluids 2010; 63(3):313–340.
48. LeVeque RJ. Balancing source terms and flux gradients in high-resolution Godunov methods: The quasi-steady wave-propagation algorithm. J Comput Phys 1998; 146(1): 346–365.
49. Liang Q, Du G, Hall JW, Borthwick AGL. Flood inundation modelling with an adaptive

quadtree grid shallow water equation solver. J Hydr Engrg 2008; 134(11):1603–10.

50. Liang Q, Marche F. Numerical resolution of well-balanced shallow water equations with complex source terms. Adv Wat Res 2009; 32: 873–884.
51. Liang Q. A structured but non-uniform Cartesian grid-based model for the shallow water equations. Int J Numer Meth Fluids 2010; DOI: 10.1002/fld.2266.
52. Liang Q. Flood simulation using a well-balanced shallow flow model. J Hydr Engrg 2010; 136(9):69–675.
53. Marche F, Bonneton P, Fabrie P, Seguin N. Evaluation of well-balanced bore-capturing schemes for 2D wetting and drying processes. Int J Numer Meth Fluids 2007; 53(5):867–894.
54. Mungkasi S, Roberts SG. On the best quantity reconstructions for a well-balanced finite volume method used to solve the shallow water wave equations with a wet/dry interface. ANZIAM J. 51 (EMAC2009) 2010; C48–C65.
55. Nair RD, Thomas SJ, Loft RD. A discontinuous Galerkin global shallow water model. Mon Weather Rev 2005; 133(4):876–888.
56. Nikolos IK, Delis AI. An unstructured node-centred finite volume scheme for shallow water flows with wet/dry fronts over complex topography. Comput Meth App Mech Eng 2010; 198:3723–3750.
57. Remacle JF, Soares-Frazão S, Li X, Shephard MS. An adaptive discretization of the shallow water equations based on discontinuous Galerkin methods. Int J Numer Meth Fluids 2006; 52(8):903–923.
58. Schwanenberg D, Harms M. Discontinuous Galerkin finite-element method for tanscritical two-dimensional shallow water flow. J Hydr Engrg 2004; 130(5):412–421.
59. Schwanenberg D. Die Runge-Kutta-Discontinuous-Galerkin-Methode zur Lösung konvektionsdominierter tiefengemittelter Flachwasserprobleme. PhD thesis; Fakultät für

Bauingenieurwesen, 2003.

60. Skoula ZD, Borthwick AGL., Moutzouris CI, Godunov-type solution of the shallow water equations on adaptive unstructured triangular grids, Int J Comput Fluid Dynam, 2006; 20(9):621-636.
61. Tang H, Solution of the shallow-water equations using an adaptive moving mesh method. Int J Numer Meth Fluids 2004; 44:789–810.
62. Tassi PA, Bokhove O, Vionnet CA. Space discontinuous Galerkin method for shallow water flows—kinetic and HLLC flux, and potential vorticity generation. Adv Wat Res 2007; 30(4):998–1015
63. Toro EF, García-Navarro P. Godunov-type methods for free-surface shallow flows: A review. J Hydr Res 2007; 45(6):737–751.
64. Toro EF. Shock-capturing methods for free-surface shallow flows. John Wiley & Sons, Ltd, 2001.
65. Valiani A, Caleffi V, Zanni A. Case Study: Malpasset dam-break simulation using a two-dimensional finite volume method. J Hydr Engrg 2002; 128(5):460–472.
66. Wang Y, Liang Q, Kesserwani G, Hall WJ. A positivity-preserving zero-inertia model for flood simulation. Comput Fluids 2011; doi:10.1016/j.compfluid.2011.01.026
67. Wirasaet D, Tanaka S, Kubatko EJ, Westerink JJ, Dawson C. A performance comparison of nodal discontinuous Galerkin methods on triangles and quadrilaterals. Int J Numer Meth Fluids 2011; DOI: 10.1002/fld.2376.
68. Xing Y, Shu CW. A new approach of high order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms. Commun Comput Phys 2006; 1(1):101–135.
69. Xing Y, Shu CW. High order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms. J

Comput Phys 2006; 214(2):567–598

70. Xing Y, Zhang X, Shu CW. Positivity-preserving high order well-balanced discontinuous Galerkin methods for the shallow water equations. Adv Wat Res 2010; 33(12), 1476–1493.
71. Yoon TH, Kang SK. Finite volume model for two-dimensional shallow water flows on unstructured grids. ASCE J Hydr Engrg 2004; 130(7):678–688.
72. Zhang M, Shu CW. An analysis of and a comparison between the discontinuous Galerkin and the spectral finite volume methods. Comput Fluids 2003, 34(4-5):581–592.
73. Zhou JG, Causon DM, Ingram DM, Mingham CG. The surface gradient method for the treatment of source terms in the shallow-water equations. J Comput Phys 2001; 168(1):1–25.
74. Zhou T, Li Y, Shu CW. Numerical comparison of WENO finite volume and Runge–Kutta discontinuous Galerkin methods. J Sci Comput 2001; 16(2):145–171.
75. Zokagoa JM, Soulaimani A. Modeling of wetting‐drying transitions in free surface flows over complex topographies. Comput Methods Appl Mech Engrg 2010; 199(33-36), 2281–2304.
76. Liang Q. A Simplified Adaptive Cartesian Grid System for Solving the 2D Shallow Water Equations. Int J Numer Meth Fluids 2011; DOI: 10.1002/fld.2568.
77. Thacker WC. Some exact solutions to the nonlinear shallow water wave equations. J Fluid Mech 1981; 107:499–508.
78. Delis AI, Nikolos IK, Kazolea M. Performance and Comparison of Cell-Centered and Node-Centered Unstructured Finite Volume Discretizations for Shallow Water Free Surface Flows. Arch Comput Methods Eng 2011; 18: 57–118.
79. Kesserwani G, Liang Q. Influence of TVD slope-limiting on local discontinuous Galerkin solutions of the shallow water equations. ASCE J Hydr Engrg 2011;

10.1061/(ASCE)HY.1943-7900.0000494.