Universal Journal of Hydraulics **1 (2013), 23-37**
www.papersciences.com

Discontinuous Galerkin Nonlinear Shallow-Water Solution on Non-Uniform Meshes with Local Time Steps

Georges Kesserwani
Department of Civil & Structural Engineering
University of Sheffield
Mappin St., Sheffield S1 3JD, UK
g.kesserwani@shef.ac.uk

Qiuhua Liang
School of Civil Engineering and Geosciences
Newcastle University
Newcastle upon Tyne NE1 7RU, UK
Qiuhua.liang@ncl.ac.uk

Abstract
This paper investigates a nonlinear water-wave model based on the finite element RKDG2 (second-order Runge-Kutta Discontinuous Galerkin) solution to the SWEs (Shallow Water Equations) on non-uniform meshes, which is herein improved to enable Local Time Steps (LTSs). A new LTS algorithm was incorporated into the RKDG2 scheme and extended to solve the SWE with complex source terms with wetting and drying to fit practical simulations. Two LTS-RKDG2 schemes that adopt 3 LTSs and 4 LTSs are configured on non-uniform meshes with 3 and 4 scales of spatial discretization, respectively. Hydraulic tests are used to verify the LTS-RKDG2 schemes by comparing their performance with their conventional RKDG2 alternatives using a Global Time Step (GTS-RKDG2). The findings show that the improved LTS-RKDG2 models produce very close predictions as the traditional GTS-RKDG2 models but with faster runtimes accelerated by a factor of 1.3 to 2.3 times depending on the simulated case.

Key Words and Phrases
Nonlinear Hyperbolic Shallow-Water Equations, Discontinuous Galerkin, Temporal Adaptation, Source Terms, Wetting and Drying, Computational Efficiency.

1. Introduction

Explicit finite volume (FV) Godunov-type numerical methods for nonlinear hyperbolic conservation laws of the unsteady SWEs [4, 13] are relevant to forecast water flows and have been receiving numerous developments (Delis & Kampanis, 2009). To summarize, a robust Godunov-type solver for the nonlinear SWEs should be able to integrate the nonlinear hyperbolic behavior within the discretization scheme, maintain its accuracy, stability and consistency when a flow discontinuity develops, steep topographic gradients are present, a wet/dry front occurs, and high roughness values are combined with very small water depths. Despite all these advances, it is still imperative to enhance the runtime of these explicit FV numerical forecasting models, which may be done by using a non-uniform adapted mesh and increasing the time step.

From this perspective, it is expected that the efficiency of an explicit numerical scheme may suffer as the size of their time steps is restricted by the CFL stability condition. This criterion provides the maximum allowable Global Time Step (GTS) permitted, which actually reduces as a result of a local increase in the velocity magnitude or a local decrease in the cell size, or both; see Eq. (5). Particularly when using non-uniform grids, few of the smallest cells may impose a restrictive time step on the whole mesh and therefore improvements in accuracy, gained by local mesh refinement, are cancelled by longer runtimes. In such a circumstance, a local time step method (LTS) whereby the solution within different cells is advanced by different time steps seems complementary to increase the computational efficiency of an explicit numerical water flow model that uses non-uniform meshes.

Few published papers have dealt with the design and implementation of LTS algorithms with shock-capturing water-wave models. Crossley & Wright [2] first examined the concept of LTS within the one-dimensional (1D) hydrodynamic numerical solution showing - merit not only in reducing runtimes but also in augmenting the quality of the numerical solution. Sanders [11] later investigated LTS with a robust two-dimensional (2D) SWEs numerical solver based on applications involving frictional flows over irregular topographies with wetting and drying. Moreover, the Implicit Friction Term Discretization (IFTD), which is a commonly used practice to stabilize water flow simulations with wetting and drying, was reported to conflict when activated with a LTS algorithm [2, 11]. In both investigations, LTS algorithms were integrated with first-order FV water-wave numerical flow models and the use of five, or more, levels of LTS was discouraged.

This work extends a LTS algorithm into the finite element second-order Runge-Kutta Discontinuous Galerkin solution for the SWEs (denoted hereafter LTS-RKDG2) on non-uniform meshes. The incorporated LTS algorithm was established by Krivodonova [9], and is found to suit the local structure of the RKDG2 water flow solver [7]. It preserves second-order accuracy, conserves locality, applies straightforwardly to the coefficients of the local finite element solution, and adapts the time step on each cell according to the cell size (i.e., it takes a LTS of Δt relative to the largest cells' size Δx, employs $\Delta t/2$ on cells of size $\Delta x/2$, $\Delta t/4$ on cells of size $\Delta x/4$, and so on). In Krivodonova [7] the LTS algorithm was verified theoretically in combination with an RKDG2 scheme (LTS-RKDG2). However, simulation results were only presented for the homogenous form of the aerodynamic equations, thus excluding the SWEs. Further, no information was provided on the relative gain in terms of runtime efficiency. Herein, the Krivodonova LTS algorithm is reformulated and improved in the framework of the RKDG2 scheme solving the one-dimensional (1D) non-homogeneous SWEs with complex source terms and involving wetting and drying [7]. Using steady and transient tests, the implementation of two LTS-RKDG2 shallow water solvers, that respectively coordinates 3 and 4 LTSs, is verified against the conventional GTS-RKDG2 model to quantify the merit of the present LTS on the runtime efficiency.

2. Nonlinear Shallow Water Equations (SWEs)

The nonlinear system of the SWEs, of hyperbolic type, can be written in a conservative matrix form:

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

where x is the longitudinal coordinate and t is the time. $\mathbf{U} = [\eta, q_x]^{\mathrm{T}}$, $\mathbf{F} = [q_x, q_x^2 h^{-1} + 0.5 g(\eta^2 - 2\eta z)]^{\mathrm{T}}$ and $\mathbf{S}$ is transposed vector containing the source terms. The source term vector $\mathbf{S}$ can be further partitioned into $\mathbf{S} = \mathbf{S_b} + \mathbf{S_f}$ where $\mathbf{S_b} = [0, -g\eta\ \partial_x z]^{\mathrm{T}}$ and $\mathbf{S_f} = [0, -C_f u|u|]^{\mathrm{T}}$ are, respectively, the topography and friction source terms (g is the acceleration due to gravity and $C_f = g n_M^2 / h^{1/3}$ with n_M being the Manning coefficient).

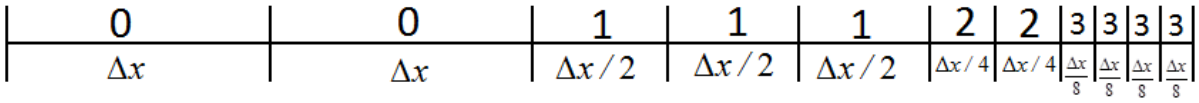

Fig. 1 Regularized 1D non-uniform structured mesh on a segment (particular case where $lev_{max} = 3$)

3. Non-uniform structured mesh

A problem domain $[x_{min}; x_{max}]$ is discretized using a coarse uniform mesh consisting of N cells of size Δx. This baseline mesh is referred to as "*background mesh*" containing the coarsest cells, which are assigned the minimum level of subdivision, *i.e.*, equal to '0'. Secondly, the background

mesh is locally refined (at certain local zones) by specifying higher levels of subdivisions up to a user-specified maximum subdivision level 'lev_{max}' (an integer). The refinement is performed in a fractal manner, *i.e.* the cell size reduces by a factor of two whenever the refinement level increases '1'. Finally, the mesh is regularized so that it does not contain adjacent cells with sizes differing by more than a factor of two. Overall, as shown in Fig. 1, mesh will consist of cells with levels varying between '0' and 'lev_{max}'. Cells with level '0' are the largest whereas cells with level 'lev_{max}' are the smallest. An arbitrary cell I_i can be classified through an assigned level denoted by '$lev(i)$' (where $0 \le lev(i) \le lev_{max}$) and thereby has a size length of $\Delta x_i = \Delta x / 2^{lev(i)}$ (with $\Delta x / 2^{lev(i)} \le \Delta x_i \le \Delta x$). Cell I_i is centered at x_i with the boundary points $x_{i\pm 1/2} = x_i \pm \frac{\Delta x_i}{2}$, *i.e.* $I_i = [x_{i-1/2};\, x_{i+1/2}]$.

4. Review of the traditional GTS-RKDG2 flow solver

Over a cell 'I_i', the RKDG2 method solves for a *local* approximate (piecewise linear) solution to (1), denoted by $\mathbf{U_h} = [\eta_h, q_h]^T$. The solution $\mathbf{U_h}(x,t)\big|_{I_i}$ is locally spanned by an average and a slope coefficient denoted herein by $\left\{\mathbf{U}_i^0(t), \mathbf{U}_i^1(t)\right\}$, and thus locally writes:

$$\mathbf{U_h}(x,t)\big|_{I_i} = \mathbf{U}_i^0(t) + \mathbf{U}_i^1(t)\frac{(x-x_i)}{\Delta x_i/2} \qquad (\forall x \in I_i) \tag{2}$$

Given the initial conditions $\mathbf{U}(x,0)$, the local coefficients, $\mathbf{U}_i^0(0)$ and $\mathbf{U}_i^1(0)$, are initialized. The topography function is approximated similarly, and therefore becomes $z_h(x,t)\big|_{I_i}$, which is discretized locally in a similar way, via topography-associated coefficient z_i^o and z_i^1 to accurately discretize irregular topography data [8]. The local approximate bed gradient thus writes $\partial_x z_h(x)\big|_{I_i} = 2z_i^1 / \Delta x_i$.

Two stages Runge-Kutta (RK) time integrator

The time updating from 't' to '$t + \Delta t$' is performed by employing a two-stage RK time stepping method [12]. Denoting by $\left(\mathbf{U}_i^{0,1}\right)^n$ and $\left(\mathbf{U}_i^{0,1}\right)^{n+1}$ to be the flow solution coefficients at time 't', and '$t + \Delta t$', respectively, the two-stage RK time integration process can be written as:

$$\left(\mathbf{U}_i^{0,1}\right)^{n+1/2} = \left(\mathbf{U}_i^{0,1}\right)^n + \Delta t\left(\mathbf{L}_i^{0,1}\right)^n \tag{3}$$

$$\left(\mathbf{U}_i^{0,1}\right)^{n+1} = \tfrac{1}{2}\left[\left(\mathbf{U}_i^{0,1}\right)^n + \left(\mathbf{U}_i^{0,1}\right)^{n+1/2} + \Delta t\left(\mathbf{L}_i^{0,1}\right)^{n+1/2}\right] \tag{4}$$

At the first RK stage (3), referred to as RK1, the local solution $\left(\mathbf{U}_i^{0,1}\right)^n$ is advanced to an intermediate state $\left(\mathbf{U}_i^{0,1}\right)^{n+1/2}$ relative to '$t^* = t + \Delta t/2$'. Then at the second RK stage (4), denoted as RK2, the solution is marched from the intermediate state at 't^*' to the next time level '$t + \Delta t$'. The DG2 space operators $\left(\mathbf{L}_i^{0,1}\right)^n$ and $\left(\mathbf{L}_i^{0,1}\right)^{n+1/2}$ are evaluated from the solution coefficients at t and t^*, respectively, as we describe in the following. The global time step (GTS) Δt_{GTS} is restricted by the Courant -Friedrich-Lewy (CFL) condition with a Courant number of 0.3 [1], namely:

$$\Delta t_{GTS} = \min_i \left(CFL \frac{\Delta x_i}{\left|\left(u_i^0\right)^n\right| + \sqrt{g\left(h_i^0\right)^n}} \right) \tag{5}$$

It is notable from (5) that when the Global Time Step (GTS) gets smaller with increasing level of refinement (i.e. $\Delta t_{GTS} \to 0$ when $lev_{\max} \to \infty$), which can significantly reduce the runtime efficiency.

Local DG2 space operators

The update of the approximate solution $\mathbf{U_h}(x,t)|_{I_i} = \left\{\mathbf{U}_i^0(t), \mathbf{U}_i^1(t)\right\}$ performs via a detached set of ODEs:

$$\begin{aligned} \partial_t \mathbf{U}_i^0(t) &= \mathbf{L}_i^0\left(\mathbf{U}_i^{0,1}, \mathbf{U}_i^{0,1}\right) \\ \partial_t \mathbf{U}_i^1(t) &= \mathbf{L}_i^1\left(\mathbf{U}_i^{0,1}, \mathbf{U}_i^{0,1}\right) \end{aligned} \tag{6}$$

in which, $\mathbf{L}_i^0$ and $\mathbf{L}_i^1$ are nonlinear vectors of space-functions. The approximating coefficients with subscripts '*in*' refers to data relative to the neighbor cells I_{in} surrounding cell I_i. These operators can be manipulated to

$$\mathbf{L}_i^0 = -\frac{\tilde{\mathbf{F}}_{i+1/2} - \tilde{\mathbf{F}}_{i-1/2}}{\Delta x_i} + \mathbf{S}\left(\mathbf{U}_i^0, z_i^0, z_i^1\right) \tag{7}$$

$$\begin{aligned} \mathbf{L}_i^1 = -\frac{3}{\Delta x_i}\Bigg\{ & \tilde{\mathbf{F}}_{i+1/2} + \tilde{\mathbf{F}}_{i-1/2} - \mathbf{F}\left(\mathbf{U}_i^0 + \frac{\hat{\mathbf{U}}_i^1}{\sqrt{3}}, z_i^0 + \frac{z_i^1}{\sqrt{3}}\right) - \mathbf{F}\left(\mathbf{U}_i^0 - \frac{\hat{\mathbf{U}}_i^1}{\sqrt{3}}, z_i^0 - \frac{z_i^1}{\sqrt{3}}\right) \\ & - \frac{\Delta x_i \sqrt{3}}{6}\left[\mathbf{S}\left(\mathbf{U}_i^0 + \frac{\hat{\mathbf{U}}_i^1}{\sqrt{3}}, z_i^0, z_i^1\right) - \mathbf{S}\left(\mathbf{U}_i^0 - \frac{\hat{\mathbf{U}}_i^1}{\sqrt{3}}, z_i^0, z_i^1\right)\right]\Bigg\} \end{aligned} \tag{8}$$

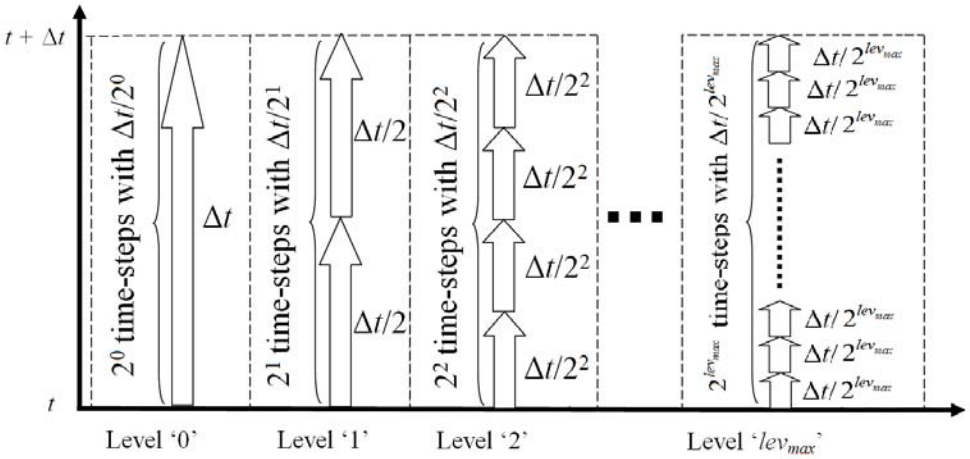

Fig. 2 LTS-RKDG2 calculation of the solution coefficients from 't' to '$t + \Delta t$' on a mesh with multiple levels of refinement '0', ..., 'lev_{max}', where a '***thick arrow***' = one iteration of LTS-RKDG2 calculation

When evaluating (7)-(8), a number of essential *spatial* ingredients should be implemented to maintain the stability of the numerical method and enable practical shallow flow modelling [7]. To summarize, firstly, *local* slope coefficients that likely cause numerical instability around discontinuous flow solution should be detected and locally limited. The "*hat*" symbol above $\mathbf{U}_i^1$ therefore refers to the limited slope coefficient (i.e., $\hat{\mathbf{U}}_i^1$). Secondly, the inter-flux $\tilde{\mathbf{F}}_{i+1/2}$ across an interface $x_{i+1/2}$ (shared by adjacent cells I_i and I_{in} with $in = i+1$) is obtained via the HLL Riemann problem solution of the two states Thirdly, before the final evaluation of (7) and (8), it is important to further implement a conservative *wetting and drying condition* to ensure the positivity of the

water height with time evolution. Lastly, the friction source term $\mathbf{S_f}$ should be separately discretized (i.e. not within (7) and (8)) by implicit discretization technique to avoid numerical instabilities that may arise when modelling water flow over dry zone with high friction factor. To ease presentation in what follows, the approximating coefficients with subscripts '*in*' will refer to the data relative to the *eastern* neighbor (i.e., cell I_{in}) of cell I_i and so $\{x_{i+1/2}\} = I_i \cap I_{in}$ represents the edge separating cells I_i and I_{in}.

5. New RKDG2 water flow model with Local-Time-Stepping (LTS-RKDG2)

The second-order LTS approach of Krivodonova [9] is integrated with the RKDG2 flow model [7] to construct the so-called LTS-RKDG2 water wave model (with improved efficiency). Further to this, special treatments are implemented to retain conservation (in time) and the applicability of the LTS-RKDG2 model for shallow flow simulations over frictional topographies with wetting and drying.

Basic concept

Assuming (for simplicity) that the maximum wave speed does not significantly influence the local *CFL* number, the LTS relative to each cell 'I_i' is therefore solely dependent on its level of refinement *lev*(*i*) (or cell size $\Delta x_i = \Delta x / 2^{lev(ic)}$) so that $\Delta t_i = \Delta t / 2^{lev(ic)}$ where Δt is now the largest time step corresponding to the coarsest cells resolution, which can be actually defined as

$$\Delta t = 2^{lev_{\max}} \times \Delta t_{GTS} \tag{9}$$

As illustrates Fig. 2, the RKDG2 calculation is locally performed with the LTS Δt, $\Delta t/2$, $\Delta t/2^2$, ..., $\Delta t/2^{lev_{\max}}$ for cells with level '0', '1', '2', ..., 'lev_{max}' to recursively advance their flow solution coefficients 1 LTS, 2 LTSs, 4 LTSs,…, $2^{lev_{\max}}$ LTSs. Herein, this iterative calculation process is referred to as "LTS-RKDG2 calculation". In the first iteration, the LTS-RKDG2 calculation achieves at cells with level '0' to allow the corresponding solution coefficients to reach time level '$t + \Delta t$'. In the second iteration, the LTS-RKDG2 calculation is undertaken at cells with level '1', and so on, until the finest cells with level 'lev_{max}' are fully updated after $2^{lev_{\max}}$ iterations of LTS-RKDG2 calculation. During a simulation, computational cells are grouped according to their level of refinements (or spatial-temporal scale) and the associated mesh may be therefore classified as "*inner cells*" and "*interface cells*". Cells neighbored by cells 'I_{in}' with similar level of refinement (or size) are called *inner cells* 'I_i'; otherwise, they are termed as *interface cells*.

At an *inner cell* 'I_i', an LTS-RKDG2 calculation is straightforward. Since 'I_i' and 'Ii_n' have the same level of refinement, they thus have the same LTS, *i.e.* $\Delta t_i = \Delta t_{in}$. One step of LTS-RKDG2 (*i.e.*, (3) and (4) with Δt_i) calculation performs in a regular manner as for the GTS-RKDG2 scheme and no special treatment is needed.

At an *interface cell* 'I_i', at least one of its adjacent neighbors is of different size *e.g.*, $\Delta x_i \neq \Delta x_{in}$, and thus $\Delta t_i \neq \Delta t_{in}$. This indicates that the LTS-RKDG2 calculation at 'I_i' is different from the one at 'I_i'. For that reason, it is essential for LTS-RKDG2 algorithm to impose synchronized '*ghost*' solution coefficients across the cells with different LTSs, and intermediate RK time stage(s), to enable the DG2 spatial operators and thereby allow the RKDG2 calculation to proceed in a classical way as described in the following.

LTS-RKDG2 flow calculation at interface cells

Since the mesh is regularized and the LTS-RKDG2 calculation simply applies recurrently, as shown in Fig. 2, it suffices to explain one elementary LTS-RKDG2 calculation at *interface cells* relative to a mesh configuration that involves two levels of refinement denoted by '*lev*0' and

'*lev*0+1' (*lev*0 is a fixed integer between 0 and $lev_{max} - 1$). Without loss of generality, we assume cell 'I_i' has a '*lev*0' and is defined as a "*large interface cell*" (LIC). Similarly, cell 'I_{in}' has a level of '*lev*0+1' and is referred to as a "*small interface cell*" (SIC). Therefore the solution coefficients at 'I_i' are indicated by a subscript 'L' and those at 'I_{in}' by 'S' (to ease presentation). Firstly, LTS-RKDG2 calculation handles the coefficients at the LIC 'I_i' (*i.e.*, the '*actual*' coefficients) facilitated by artificially reconstructed synchronized coefficients (*i.e.*, '*ghost*' coefficients) at the SIC 'I_{in}'. Then, it is necessary to apply the LTS-RKDG2 calculation to deal with the case where the '*actual*' coefficients are at the SIC 'I_{in}' and the '*ghost*' coefficients are at the LIC 'I_i'.

- *Flow solution's coefficients update at the LIC 'I_i'*

First, the LTS-RKDG2 calculation starts at the LIC cell 'I_i' where the LTS is $\Delta t_L = \Delta t / 2^{lev0}$ and the initial coefficients at time 't' are $(\mathbf{U}_i^{0,1})_L^n$. At its SIC neighbor 'I_{in}', the initial coefficients $(\mathbf{U}_{in}^{0,1})_S^n$ are also available. Therefore, the DG2 space operators $\left(\mathbf{L}_i^{0,1}\right)_L^n$ can be evaluated directly and then plugged into (3) to complete the RK1 stage and produce the '*actual*' coefficients $(\mathbf{U}_i^{0,1})_L^{n+1/2}$ relative to the intermediate time state 't^*'. Similarly, time-matching '*ghost*' coefficients $(\mathbf{U}_{in}^{0,1})_{S,Ghost}^{n+1/2}$ relative to t^* are constructed at I_{in} by advancing its coefficients using (3) with LTS Δt_L, i.e.

$$\left(\mathbf{U}_{in}^{0,1}\right)_{S,Ghost}^{n+1/2} = \left(\mathbf{U}_{in}^{0,1}\right)_S^n + \Delta t_L \left(\mathbf{L}_{in}^{0,1}\right)_S^n \tag{10}$$

After (10), the coefficients are synchronized at 't^*'. The space operators $\left(\mathbf{L}_i^{0,1}\right)_L^{n+1/2}$ can then be evaluated and used to achieve the RK2 stage in (4) to finally produce the coefficients $(\mathbf{U}_i^{0,1})_L^{n+1}$ associated to $'t + \Delta t_L'$.

- *Flow solution's coefficients update at the SIC 'I_{in}'*

After obtaining the '*actual*' coefficients at '$t + \Delta t_L$' over cell 'I_i', the SIC 'I_{in}' is now reconsidered to obtain its '*actual*' coefficients at '$t + \Delta t_L$'. However, since the LTS over a SIC is halved (*i.e.*, $\Delta t_S = \Delta t_L / 2$), LTS-RKDG2 calculation over 'I_{in}' must be carried out in two consecutive iterations, starting from the available initial coefficients, $(\mathbf{U}_{in}^{0,1})_S^n$ and $(\mathbf{U}_i^{0,1})_L^n$, at time 't' at both SIC 'I_{in}' and its neighbor LIC 'I_i'. It should be noted that the '*ghost*' coefficients at SIC 'I_{in}' produced (refer to the previous subsection) using (10) are here inappropriate for the current step and will therefore be ignored. In contrast, at the LIC 'I_i', certain preceding information created by the previous LTS-RKDG2 update for the '*actual*' coefficients over 'I_i' may be saved and reused. In particular, the previously calculated DG2 space operators $(\mathbf{L}_i^{0,1})_L^n$ and $(\mathbf{L}_i^{0,1})_L^{n+1/2}$ at time 't' and time 't^*' are used to define the following quadratic function:

$$\phi_i^{0,1}(\tau) = (\mathbf{U}_i^{0,1})_L^n + (\mathbf{L}_i^{0,1})_L^n (\tau - t) + \frac{(\mathbf{L}_i^{0,1})_L^{n+1/2} - (\mathbf{L}_i^{0,1})_L^n}{2\Delta t_L} (\tau - t)^2 \tag{11}$$

which will serve to reconstruct '*ghost*' coefficients over the neighbor LIC 'I_i' at an inner fractional-time-step 'τ' and its associated time-stage 'τ^*', namely:

$$\left[\mathbf{U}_i^{0,1}(\tau)\right]_{L,Ghost}^n = \phi_i^{0,1}(\tau) \qquad \left(\tau \in \left[t; t + \Delta t_L\right[\right) \tag{12}$$

$$\left[\mathbf{U}_i^{0,1}(\tau^*)\right]_{L,Ghost}^{n+1/2} = \left[\mathbf{U}_i^{0,1}(\tau)\right]_{L,Ghost}^{n} + \Delta t_S \frac{d}{d\tau}\phi_i^{0,1}(\tau) \qquad \left(\tau^* \in \left[\tau; t+\Delta t_L\right[\right) \tag{13}$$

Eqs. (11)-(13) give a second-order accurate data interpolation as confirmed by the theoretical work in Krivodonova (2010). At the first iteration, the coefficients over 'I_{in}' are advanced one LTS from time 't' to time '$t_2 = t + \Delta t_S$'. Initially at time 't', using $(\mathbf{U}_{in}^{0,1})_S^n$ and $(\mathbf{U}_i^{0,1})_L^n$, the space operators $(\mathbf{L}_{in}^{0,1})_S^n$ can be evaluated and then inserted into (3) to achieve the calculation at the RK1 stage and produce $(\mathbf{U}_{in}^{0,1})_S^{n+1/2}$, which are the '*actual*' coefficients at the intermediate time stage $t_1^* = t + \Delta t_S/2$. Meanwhile, over 'I_i', the '*ghost*' coefficients relative to t_1^*, i.e. $(\mathbf{U}_i^{0,1})_{L,Ghost}^{n+1/2}$, should be constructed at $\tau = t_1^*$ by means of (11)-(13).

After this, the DG2 space operators $(\mathbf{L}_{in}^{0,1})_S^{n+1/2}$ at time t_1^* can be evaluated and put in (4) to finalize the RK2 stage and produce $(\mathbf{U}_{in}^{0,1})_S^{n+1}$, which are the 'actual' coefficients over I_{in} at time t_2. Meanwhile, the time-matching '*ghost*' coefficients over 'I_i' at t_2, i.e. $(\mathbf{U}_i^{0,1})_{L,Ghost}^{n+1}$, should be constructed using (11) and (12) evaluated at $\tau = t_2$.

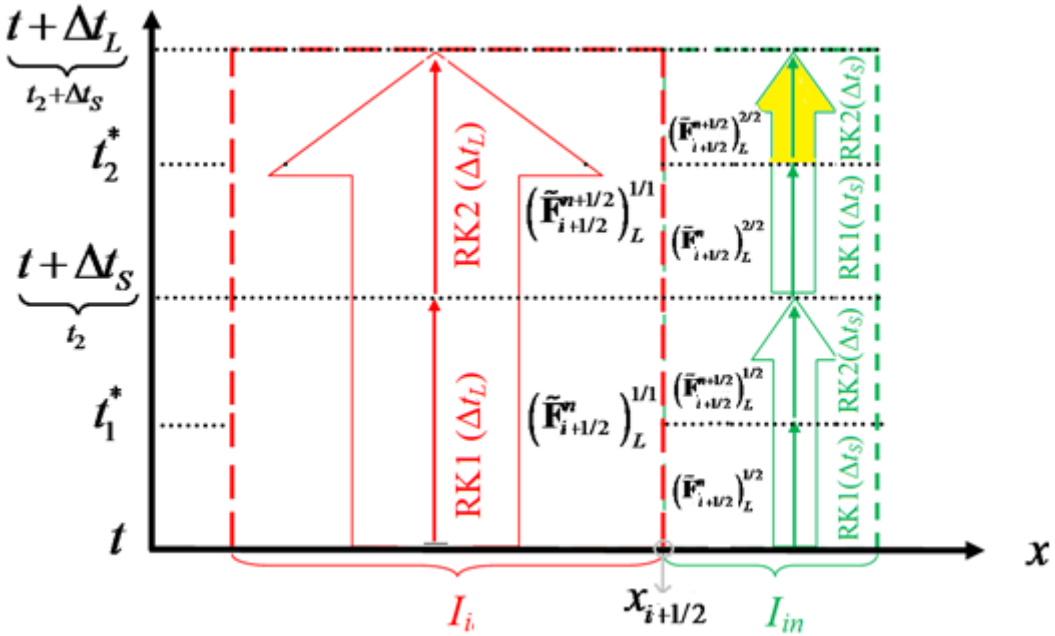

Fig. 3 History of the actual stages of LTS-RKDG2 calculations at a LIC I_i adjacent to a SIC I_{in} and the associated Riemann fluxes. Particular case (when $\Delta t_L = \Delta t$).

At the second iteration, the coefficients are reinitialized at 't_2' and another RKDG2 calculation step with the LTS Δt_S is performed to further elevate the coefficients over I_{in} to time level '$t + \Delta t_L$'. That is, both '*actual*' and '*ghost*' coefficients at I_i and I_{in}, respectively, are reinitialized at 't_2' (*i.e.* $(\mathbf{U}_{in}^{0,1})_S^n \leftarrow (\mathbf{U}_{in}^{0,1})_S^{n+1}$ and $(\mathbf{U}_i^{0,1})_L^n \leftarrow (\mathbf{U}_i^{0,1})_{L,Ghost}^{n+1}$). The previously employed '*actual*' and '*ghost*' coefficients and their associated DG2 space operators at the inner time stages can be reused (*i.e.*, herein overwritten). Initially at time level 't_2', $(\mathbf{U}_{in}^{0,1})_S^n$ and $(\mathbf{U}_i^{0,1})_L^n$ are synchronized and so the operators $(\mathbf{L}_{in}^{0,1})_S^n$ can be evaluated and then put into (3) to achieve the RK1 stage and produce '*actual*' coefficients $(\mathbf{U}_{in}^{0,1})_S^{n+1/2}$, which now represent the coefficients at

the time stage $t_2^* = t_2 + \Delta t_S / 2$. Meanwhile, over '$I_i$', time-matching (*i.e.*, relative to t_2^*) '*ghost*' coefficients are reconstructed using (11)-(13) evaluated at $\tau = t_2^*$ to produce $(\mathbf{U}_i^{0,1})_{L,Ghost}^{n+1/2}$. Now, $(\mathbf{U}_i^{0,1})_{L,Ghost}^{n+1/2}$ and $(\mathbf{U}_{in}^{0,1})_S^{n+1/2}$ are synchronized and can be used to calculate the DG2 operators $(\mathbf{L}_{in}^{0,1})_S^{n+1/2}$ and then inserted into the RK2 stage (4) to finally produce $(\mathbf{U}_{in}^{0,1})_S^{n+1}$, which now represent the coefficients over 'I_{in}' at time '$t + \Delta t_L$'.

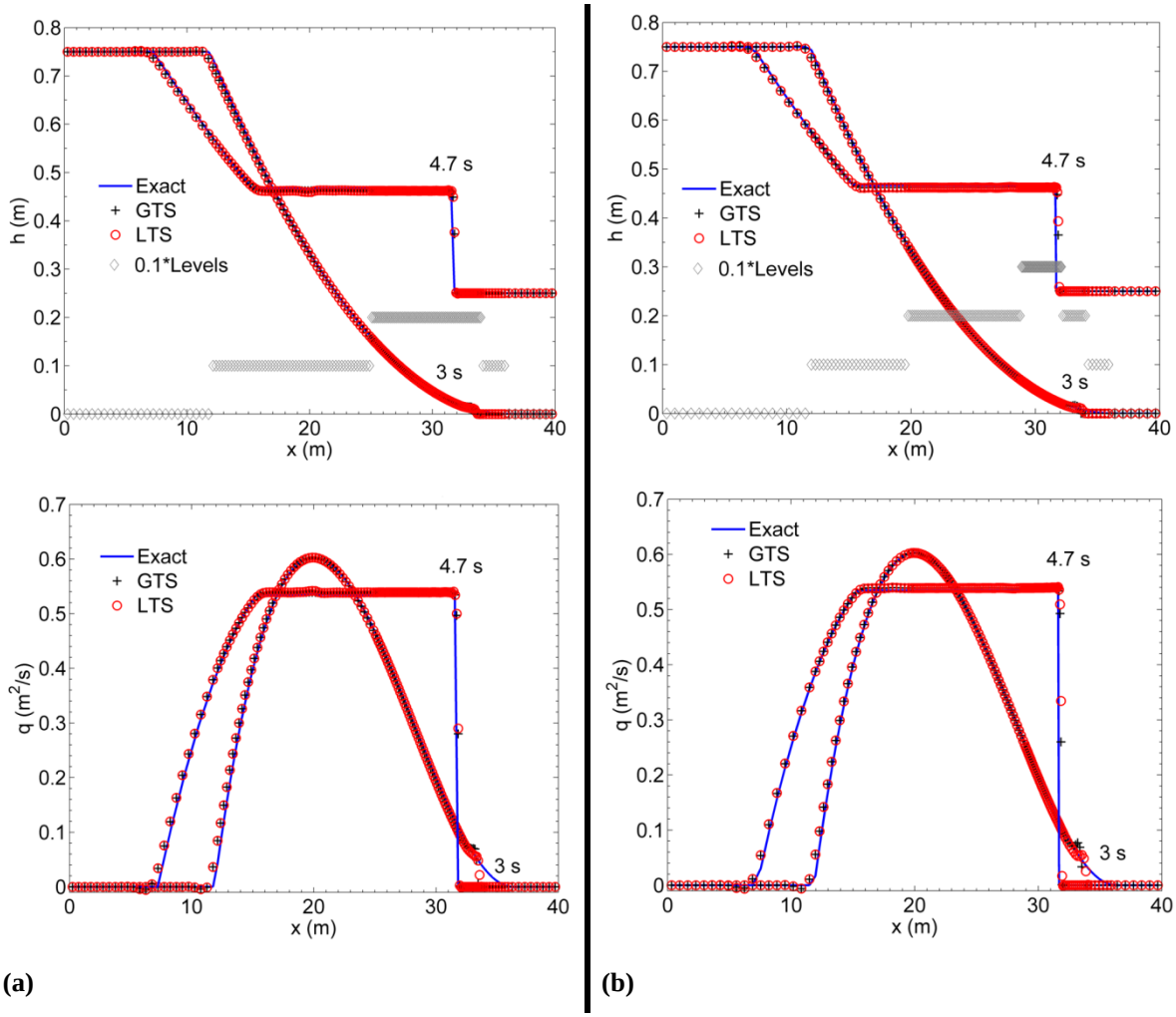

Fig. 4 Idealized dam-break flows. LTS-RKDG2 calculations *vs.* GTS-RKDG2 calculations compared with the analytical solution; (a) $lev_{max} = 2$ and (b) $lev_{max} = 3$.

Friction term issue and flux conservation enforcement

Due to the LTS dependence within the friction term discretization, its minor side effect due to the stable implicit discretization [10], may increasingly magnify at those *inner cells*. Further, this aspect complicates the implicit friction term discretization (IFTD) with the LTS-RKDG2 calculation at *interface cells* because extra phases of '*ghost*' friction advancement, and removal, need to be entailed in line with the '*ghost*' coefficients advancement. Therefore, herein, the usability of the IFTD is restricted to those cells where the water height may potentially become infinitesimal. At the other (flood) cells, the friction source term is discretized explicitly in the DG2 space operators, in a straightforward manner, as it is now free from any LTS dependence.

After the LTS-RKDG2 calculations at the LIC 'I_i' and the SIC 'I_{in}', the sum of Riemann flux quantities cumulated between time 't' and time '$t + \Delta t_L$' at the edge '$x_{i+1/2}$' may not be equal. For

instance, following the notations in Fig. 3, it may happen that the sum of Riemann flux evaluations at '$x_{i+1/2}$' accumulated from the LTS-RKDG2 calculation at the LIC 'I_i' (*i.e.*, , sum of fluxes with superscript '1/1') is different than the sum of Riemann flux evaluations at '$x_{i+1/2}$' accumulated during the LTS-RKDG2 calculations at the SIC 'I_{in}' (*i.e.*, Fig. 3, sum of fluxes with superscript '1/2' and '2/2'). Therefore, flux conservation (in time) is imposed via:

$$\left(\tilde{\mathbf{F}}_{i+1/2}^{n+1/2}\right)_S^{2/2} = \left(\tilde{\mathbf{F}}_{i+1/2}^{n} + \tilde{\mathbf{F}}_{i+1/2}^{n+1/2}\right)_L^{1/1} - \left(\tilde{\mathbf{F}}_{i+1/2}^{n} + \tilde{\mathbf{F}}_{i+1/2}^{n+1/2}\right)_S^{1/2} - \left(\tilde{\mathbf{F}}_{i+1/2}^{n}\right)_S^{2/2} \tag{14}$$

instead of estimating it from the HLL nonlinear Riemann solver (Toro 2001).

6. LTS-RKDG2 model's validation relative the traditional GTS-RKDG2 model

The LTS-RKDG2 scheme is verified in this section. LTS-RKDG2 and GTS-RKDG2 schemes simulations are run on two non-uniform meshes, referred hereafter to as '*mesh-3LTSs*' and '*mesh-4LTS*', which have been configured to allow up to '2' and '3' levels of refinement, respectively, while retaining the same total number of computational cell for both meshes. Over these meshes, LTS-RKDG2 local solutions coordinates LTSs of {Δt, $\Delta t/2$, $\Delta t/4$} and {Δt, $\Delta t/2$, $\Delta t/4$, $\Delta t/8$}, respectively, at cells of sizes {Δx, $\Delta x/2$, $\Delta x/4$} and {Δx, $\Delta x/2$, $\Delta x/4$, $\Delta x/8$}. The level of refinement relative to each cell is indicated by a gray '*diamond*' marker within the sub-figures that illustrate the free-surface elevations. Selected benchmarks are considered to investigate the performance of LTS-RKDG2 scheme with respect to the GTS-RKDG2 scheme, while discussing practical hydrodynamic issue and quantifying the relative saving in runtime efficiency (via the ratio 'runtime GTS'/'runtime LTS').

Idealized dam-break flows

These problems involve an idealized dam-break flow in a 40m long channel with frictionless and horizontal bed. At x_0 = 20m, an imaginary dam separating two discontinuous still water levels is assumed, which breaks totally at t = 0s producing (nonlinear) wave propagations in either directions. The initial water level upstream of the dam is 0.75m and two downstream depths are selected splitting this problem into two cases. The first one involves a flow in a fully wet domain and associates to an initial downstream water height of 0.25m. The second case produces wave front propagation over a dry zone and obtains by taking a zero (initial) downstream height downstream of the dam. In this test, '*mesh-3LTSs*' and '*mesh-4LTS*' consisted of 162 cells. Simulations are created until t = 4.7s and t = 3s for the first and second cases, respectively. The depth and discharge profiles predicted by the LTS- and GTS-RKDG2 schemes on both meshes are displayed in Fig. 4 compared with the exact solutions. For both cases, the LTS-RKDG2 scheme is seen to successfully capture the nonlinear discontinuous patterns and produces identical predictions as the one due to GTS-RKDG2 scheme. For the second case, the LTS-RKDG2 scheme produces a slightly better trail to the movement of the wet/dry front than the GTS-RKDG2 model. On the '*mesh-3LTSs*' (Fig. 4a), the LTS-RKDG2 scheme is found to be 1.38 times and 1.32 times faster than the GTS-RKDG2 model for, respectively, the first and the second cases. On the other '*mesh-4LTSs*' (i.e., Fig. 4b), the LTS-RKDG2 scheme is noted 1.76 times and 1.62 times more computationally efficient than the GTS-RKDG2 scheme for the two cases. This demonstrates the ability of the LTS-RKDG2 method to handle the discontinuous and nonlinear shallow flow solutions with a similar accuracy as the GTS-RKDG2 but at a reduced CPU time cost.

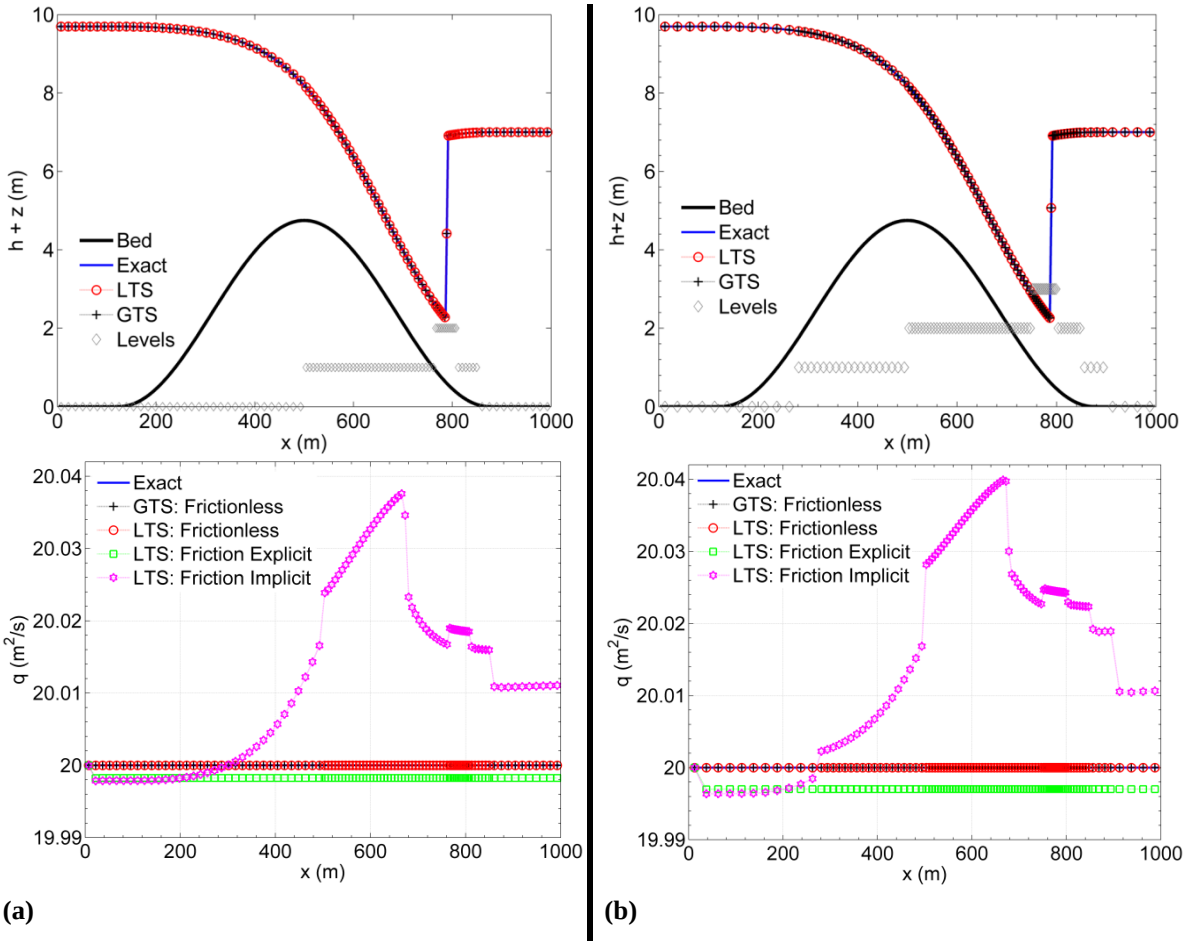

Fig. 5 Transcritical flow over a hump with shock. LTS-RKDG2 calculations *vs.* GTS-RKDG2 calculations compared with the analytical solution; (a) $lev_{max} = 2$ and (b) $lev_{max} = 3$.

Steady flow over topography with shock

In this section an academic test case involving moving steady transcritical flow over non-flat topography, with a shock, is investigated. This test is usually employed to simultaneously demonstrate the capability of a numerical method to: converge towards a steady state, accurately balance the flux gradient with the topography gradient (i.e., verify the well-balanced property [7]), inherently capture flow smooth transitions and discontinuities. The channel is 1000m length with a hump-shape topography located between $x = 125$m and $x = 875$m. An upstream (steady) subcritical inflow is imposed through a unit discharge of $20\text{m}^2/\text{s}$ and the outflow depth is fixed to 7m. Under these conditions, a steady transitional flow takes place where the flow changes from subcritical to supercritical. Then at the downstream of the topography, a hydraulic jump is formed and the flow restores to be subcritical. A simulation starts from an initial water height of 9.7m and is required to stop after a relatively long time evolution (i.e. t = 2000s). Simulations are performed on '*mesh-3LTSs*' and '*mesh-4LTSs*' both consisting of 100 cells.

At first, the channel's bed is assumed frictionless and the GTS-RKDG2 and LTS-RKDG2 schemes are run on the two non-uniform meshes. Fig. 5a and Fig. 5b present the corresponding steady state profiles acquired by the two RKDG2 solvers on '*mesh-3LTSs*' and '*mesh-4LTSs*', respectively. It can be seen that the numerical water depths produced by both solvers match well the analytical solutions and no visual difference is detected among the two schemes. On the other hand, both numerical models reached the expected conservative state for the steady discharge solution; this indicates that the current LTS algorithm do not disturb the well-balanced property when integrated with the RKDG2 scheme. For '*mesh-3LTSs*' and '*mesh-4LTSs*' meshes, respectively, the runtime of the LTS-RKDG2 model is noted to reduce by 1.9 times and 2.3 times

with respect to the GTS-RKDG2 scheme. This shows that the usefulness of the transient LTS-RKDG2 shallow water solver in accelerating convergence to steady flow problems.

Secondly, we use this test case to further point out the inconvenience of the IFTD when implemented solely in conjunction with the LTS-RKDG2 scheme. Therefore, the LTS-RKDG2 method is reconsidered with a Manning factor of 0.033 s/m$^{1/3}$; the simulations are remade on the same meshes but with a focus on comparing the IFTD discretization (i.e., time-dependent) vs. the explicit friction term discretization (i.e., independent of the time-step). The solution to the momentum equation, in terms of steady discharge numerical result, is appended within the discharge plots of Fig. 5a and 5b. As discussed before, the use of the IFTD with the LTS-RKDG2 tends to magnify the side effect of the IFTD, due to its dependence on the LTS, by increasing the amount of numerical diffusion and thereby disturb the ability of the RKDG2 scheme to predict steady flow discharges over non-flat topography. This side effect was noted to increase in line with either an increase in the Manning factor (herein, discharge illustrations in Figs. 5a and 5b contains the results relative to the highest value of n_M that was tested, i.e., n_M = 0.033 s/m$^{1/3}$) or in the level of LTSs (i.e. the LTS-RKDG2-IFTD in Fig. 5a is less diffusive than the one in Fig. 5b). As expected, the discharge solution reproduced by the LTS-RKDG2 scheme with the explicit friction discretization on both '*mesh-3LTSs*' and '*mesh-4LTSs*' remain comparatively unaffected—despite an insignificant drop that is believed to occur as a results of coarsening the mesh at the boundary and also due to the heuristic nature of Manning's formula. These results justify the motivation to avoid global application of the IFTD. In the following tests, therefore, the friction term discretization is set to be explicit at each cell that is surrounded by flood cells; otherwise, the LTS-RKDG2 code was set to switch to the IFTD.

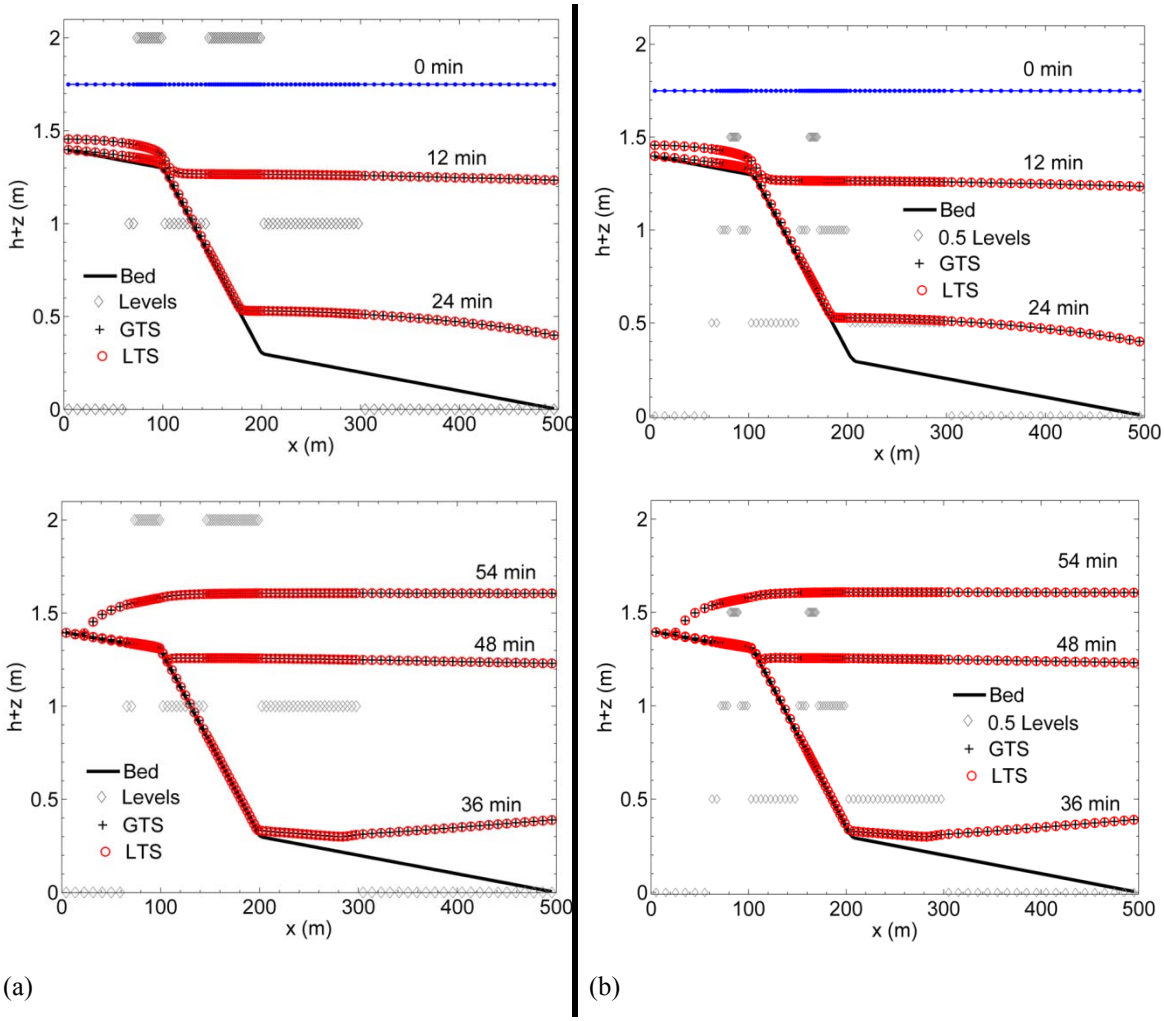

Fig. 6 LTS-RKDG2 calculations *vs.* GTS-RKDG2 calculations; (a) lev_{max} = 2 and (b) lev_{max} = 3.

Tidal flow cycle over topography with friction

This synthetic tidal wave case was initiated by Heniche et al. [6] and is a commonly used test case to verify the stability and robustness of a numerical model when reproducing the movement of a wet/dry front over an uneven frictional topography. It can be regarded as a tidal wave running up and down over sloping beach in a 1D domain [0m; 500m] with a slope of -0.001 over [0m; 100m], -0.01 over]100m; 200m] and -0.001 over]200m ;500m]. The friction effects are quite significant as they associate to a Manning coefficient of n_M = 0.03. The flow is initially still with a constant surface elevation of 1.75m. The eastern end of the domain (x = 500m) is assumed to be the inlet where the varying water depth reads

$$h(500,t) = 1 + 0.75\cos\left(\frac{2\pi t}{T}\right) \tag{15}$$

which mimics a tidal wave with T = 60min representing the period of a tidal cycle. The western end of the domain is a standing solid wall. The LTS-RKDG2 scheme is applied on '*mesh-3LTSs*' and '*mesh-4LTSs*' that both involved a total of 100 cells. A simulation is run up to t = 60min (i.e. one tidal cycle). The LTS-RKDG2 simulations of the advancing and recessing shoreline, at t = 0, 12, 24, 36, 48 and 54min are presented in Fig. 6a and Fig. 6b, comparing satisfactorily with the predictions obtained by the GTS-RKDG2 scheme on the same meshes. Driven by the inflow defined by (15), the initial water surface starts to decrease gradually from 1.75m. At t = 12min, the originally submerged domain is about to dry out from the first turning point of the sloping bed (x = 100m) where the flow is separated into two parts. Upstream of the turning point, the flow moves more slowly due to the gentler slope, as can be observed in the water surface profile recorded at t = 24min. At t = 36 min, the upstream slowly moving water has been entirely drained to downstream and the inflow water depth has started to increase after touching the minimum. The shoreline runs slowly up the beach (as shown by the results at t = 48 and t = 54min) until the entire domain is submerged again and the free-surface reaches its original level. The current predictions also agree closely with those presented in literature. Altogether, the LTS-RKDG2 scheme delivered similar results to those produced by the GTS-RKDG2 model but with 1.5 times and 2.5 times less runtime cost on '*mesh-3LTSs*' and '*mesh-4LTSs*', respectively.

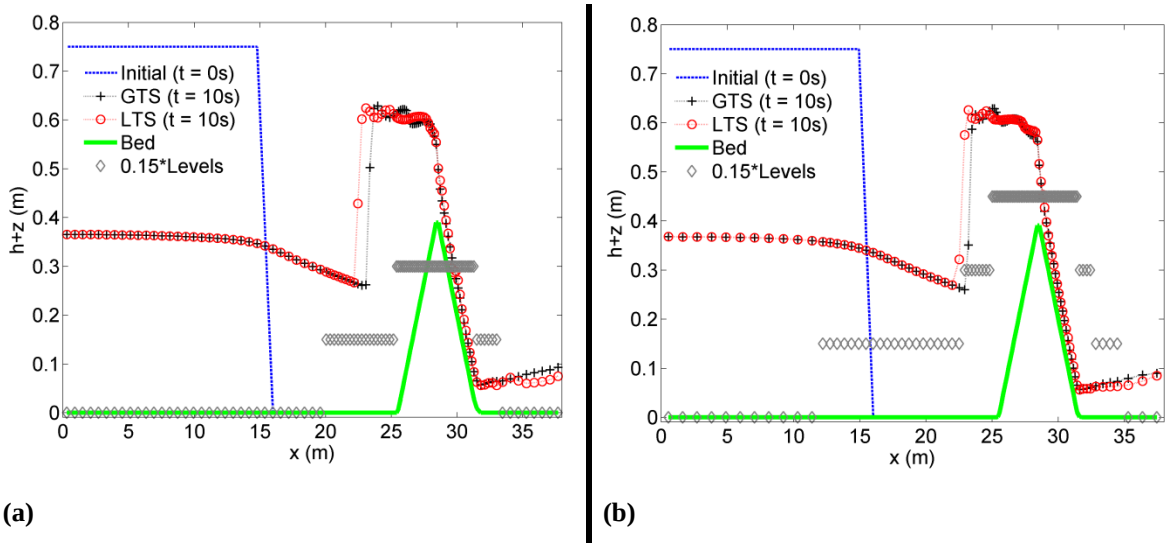

Fig. 7 Dam-break flow interacting with a triangular obstacle at t = 10s. LTS-RKDG2 calculations *vs.* GTS-RKDG2 calculations; (a) lev_{max} = 2 and (b) lev_{max} = 3.

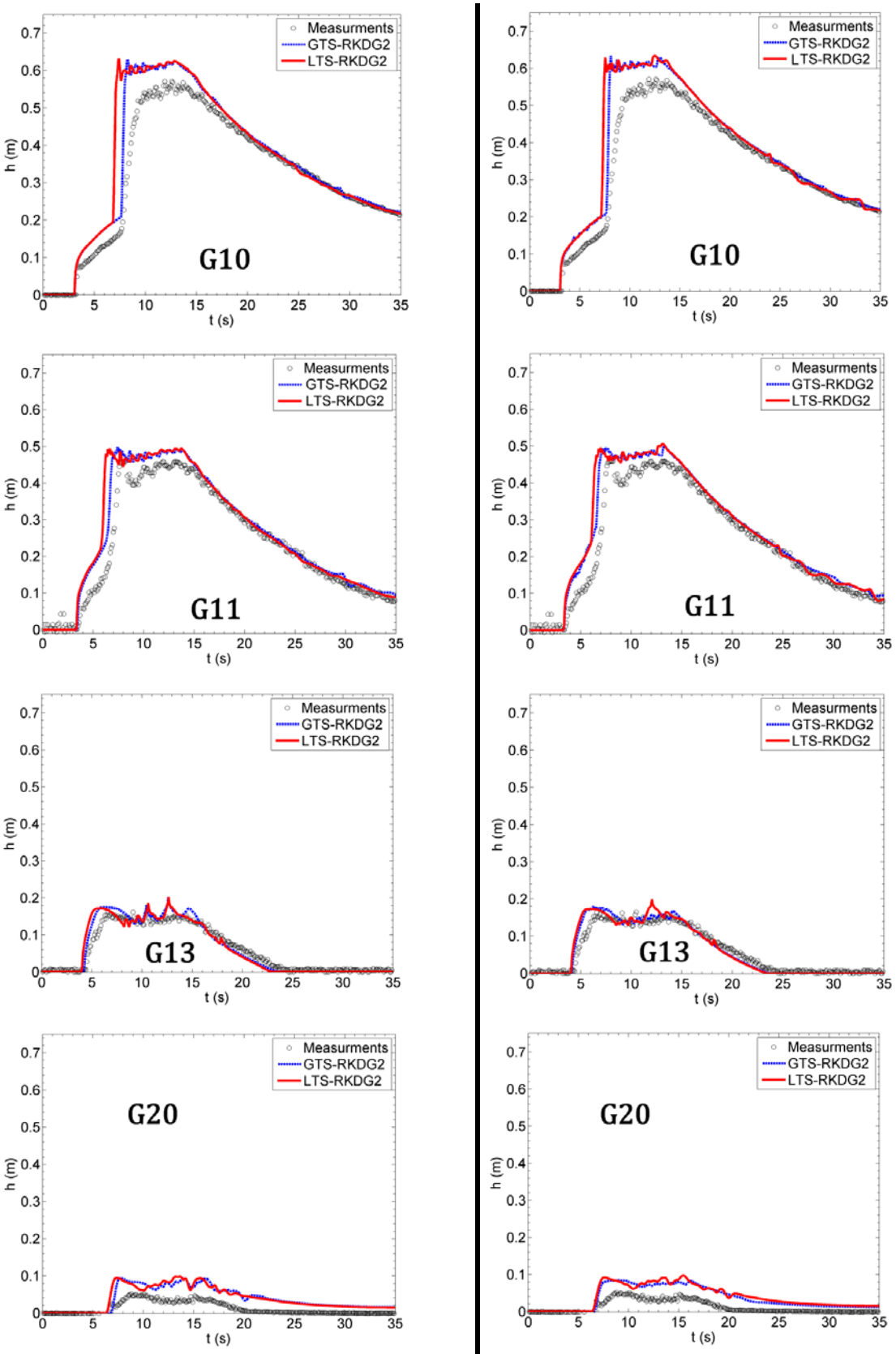

Fig. 8 RKDG2 scheme dam-break experiment simulations: (a) lev_{max} = 2 and (b) lev_{max} = 3.

Dam-break experiment test

The LTS-RKDG2 scheme is finally assessed by replicating a dam-break wave propagating over an initially dry and rough floodplain and interacting with a solid obstacle (see Fig. 7). The length of the domain is 38m; the initial condition is a still water state (i.e. 0.75 m) held by an imaginary dam and a dry floodplain located downstream of the dam (see Fig. 7). For this problem, measured time histories of the water depth are available at point G10, G11, G13 and G20 that are respectively located 10 m, 11 m, 13 m and 20 m downstream of the dam. The friction effects are associated to a Manning factor of 0.0125 and the upstream boundary is a solid wall. A total of 100 cells is used to form meshes of type '*mesh-3LTSs*' and '*mesh-4LTSs*' for the 1D domain [0m; 38m]. The motionless initial condition is plotted in Fig. 7 and the simulation output time is set to t = 35s. The longitudinal profiles of the free-surface elevation predicted by the two RKDG2 alternatives on both meshes are illustrated Fig. 7 (i.e., at t = 10s). Fig. 8 contains the predicted time histories that are seen to favorably agree with the available experimental data [6]. As anticipated, the generated wave front propagates towards the obstacle, climbs up and overtops the triangular hump, creates a reflected shock-wave moving towards the upstream wall. After reaching the upstream boundary, it bounds back and propagates from side to side until it is dampened by the effect of friction. Both RKDG2 schemes survived this challenging benchmark for the two considered meshes. In contrast to the previous tests, difference between the LTS-RKDG2 and GTS-RKDG2 predictions is detected; this may be attributed to dependence of the IFTD on the LTS and the involvement of relatively high local velocities in this test. However, these differences are little and seem to have inconsequential effects on the usability of the LTS-RKDG2 model. Further, it should be noted that the over-predictive aspect delivered by all the RKDG2 predictions at G20 has no concern with the numerical algorithms; it is usually attributed to the fact that the wave pattern downstream of the obstacle becomes highly complex and unstable and so the hydrostatic assumption of the shallow water equations is no longer valid. For this case, the use of the non-uniform mesh LTS-RKDG2 scheme over '*mesh-3LTSs*' and '*mesh-4LTSs*' enhanced, respectively, the time efficiency by 1.32 times and 1.36 times over the GTS-RKDG2 scheme.

7. Summary and Conclusions

A second-order LTS algorithm has been integrated with a robust finite element RKDG2 water model on structured non-uniform meshes (LTS-RKDG2). Stabilizing spatial discretization ingredients that enable the practical utility of the RKDG2 shallow water numerical models were retained. Further considerations were given to maintain the flux conservation across cells of different sizes, and to also diminish the adverse effects of the implicit friction effects discretization (due to the involvement of the LTS within). Two LTS-RKDG2 water flow models, which adopt LTSs of $\{\Delta t, \Delta t/2, \Delta t/4\}$ and $\{\Delta t, \Delta t/2, \Delta t/4, \Delta t/8\}$ on non-uniform meshes were set and tested with reference to the associated GTS-RKDG2 versions. Our numerical experiments show that the use of LTS algorithm in an RKDG2 SWEs numerical solver is able to generically produce similar prediction as the GTS-RKDG2 counterparts. For the considered tests, the LTS-RKDG2 scheme boosted the computational efficiency, over the traditional GTS-RKDG2 model, by 1.32-to-1.99 times when adapting 3 LTSs, and by 1.36-to-2.3 times when adapting 4 LTSs. Research is currently underway to investigate the merit of LTS-RKDG2 model in the context of 2D hydrodynamic simulations.

References

1. *Cockburn B., Shu C.-W.,* 'Runge-Kutta discontinuous Galerkin methods for convection-dominated problems', *J. Sci. Comput.*, ***16*** *(2001), 173-261.*
2. *Crossley A.J., Wright N.G.,* 'Time accurate local time stepping for the unsteady shallow water equations', *Int. J. Numer. Methods Fluids*, ***48*** *(2005),775-799.*
3. *Delis A.I., Kampanis N.A.,* '*Numerical flood simulation by depth averaged free surface flow models*', in *Environmental Systems, in Encyclopedia of Life Support Systems (EOLSS)*, A. Sydow, Editor, *2009.*

4. *Guinot V.* '*Godunov-type schemes: an introduction for engineers*', Elsevier, Amsterdam, *2003.*
5. *Heniche M., Secretan Y., Boudreau P., Leclerc M.,* 'A two-dimensional finite element drying-wetting shallow water model for rivers and estuaries', *Adv. Wat. Resour., **23** (2000), 359-372,*
6. *Hiver J.M.* '*Adverse-slope and slope (bump)*' in Concerted Action on Dam Break Modelling: Objectives, Project Report, Test Cases, Meeting Proceedings. Université catholique de Louvain, Civ. Eng. Dept., Hydraulics Division, Louvain-la-Neuve, Belgium, *2000.*
7. *Kesserwani G. Liang, Q.,* 'Locally limited and fully conserved RKDG2 shallow water solutions with wetting and drying', *J. Sci. Comput.*, ***50** (2012) 120-144.*
8. *Kesserwani G., Liang Q., Vazquez, J., Mosé, R.,* 'Well-balancing issues related to the RKDG2 scheme for the shallow water equations', *Int. J. Numer. Methods Fluids*, ***62** (2010), 428-448.*
9. *Krivodonova L.,* 'An efficient local time-stepping scheme for solution of nonlinear conservation laws', *J. Comput. Phys., **229** (2010), 8537-8551.*
10. *Murillo J., García-Navarro P., Burguete J.,* 'Time step restrictions for well-balanced shallow water solutions in non-zero velocity steady states', *Int. J. Numer. Methods Fluids, **60** (2009), 1351-1377.*
11. *Sanders B.F.,* 'Integration of a shallow water model with a local time step', *J Hydraul. Res., **46** (2008), 466-475.*
12. *Shu C.-W., Osher S.,* 'Efficient implementation of essentially non-oscillatory shock-capturing schemes', *J. Comput. Phys., **77** (1988) 439-471.*
13. Toro E.F., '*Shock-capturing methods for free-surface shallow flows'*, John Wiley & Sons, Chichester *2001.*