Article:
Gerhard, N., Caviedes-Voullième, D., Müller, S. et al. (1 more author) (2015) Multiwavelet-based grid adaptation with discontinuous Galerkin schemes for shallow water equations. Journal of Computational Physics, 301. 265 - 288. ISSN 0021-9991

https://doi.org/10.1016/j.jcp.2015.08.030

Multiwavelet-Based Grid Adaptation with Discontinuous Galerkin Schemes for Shallow Water Equations

Nils Gerhard[a], **Daniel Caviedes-Voullième**[b]
Siegfried Müller[a] **and Georges Kesserwani**[b]

Preprint No. 411 **November 2014**

Key words: Grid adaptivity, Multiresolution analysis, Multiwavelets, Discontinuous Galerkin, Shallow water ows, Positivity-preserving, Well-balancing, Limiters, Shock detection

AMS Subject Classifications: 65M50, 65M60, 35L65, 65T60

Institut für Geometrie und Praktische Mathematik
RWTH Aachen

Templergraben 55, D-52056 Aachen, (Germany)

[a] Institut für Geometrie und Praktische Mathematik, RWTH Aachen University, Templergraben 55, 52056 Aachen, Germany

[b] Civil & Structural Engineering, University of Sheffield, Sheffield, UK

Multiwavelet-Based Grid Adaptation with Discontinuous Galerkin Schemes for Shallow Water Equations

Nils Gerhard[a], Daniel Caviedes-Voullième[b], Siegfried Müller[a], Georges Kesserwani[b]

[a] *Institut für Geometrie und Praktische Mathematik, RWTH Aachen University, Templergraben 55, 52056 Aachen, Germany*
[b] *Civil & Structural Engineering, University of Sheffield, Sheffield, UK*

Abstract

We provide an adaptive strategy for solving shallow water equations with dynamic grid adaptation including a sparse representation of the bottom topography. A challenge in computing approximate solutions to the shallow water equations including wetting and drying is to achieve the positivity of the water height and the well-balancing of the approximate solution. A key property of our adaptive strategy is that it guarantees that these properties are preserved during the refinement and coarsening steps in the adaptation process.

The underlying idea of our adaptive strategy is to perform a multiresolution analysis using multiwavelets on a hierarchy of nested grids. This provides difference information between successive refinement levels that may become negligibly small in regions where the solution is locally smooth. Applying hard thresholding the data are highly compressed and local grid adaptation is triggered by the remaining significant coefficients. Furthermore we use the multiresolution analysis of the underlying data as an additional indicator whether the limiter has to be applied on a cell or not. By this the number of cells where the limiter is applied is reduced without spoiling the accuracy of the solution.

By means of well-known 1D and 2D benchmark problems, we verify that multiwavelet-based grid adaptation can significantly reduce the computational cost by sparsening the computational grids, while retaining accuracy and keeping well-balancing and positivity.

Keywords: Grid adaptivity, Multiresolution analysis, Multiwavelets, Discontinuous Galerkin, Shallow water flows, Positivity-preserving, Well-balancing, Limiters, Shock detection

1. Introduction

The shallow water equations are a hyperbolic system of conservation laws which have been commonly used to model shallow free surface water flows. These are given by

$$\begin{pmatrix} h \\ h\mathbf{v} \end{pmatrix}_t + \nabla \cdot \begin{pmatrix} h\mathbf{v} \\ h\mathbf{v} \otimes \mathbf{v} + \frac{1}{2} g h^2 I \end{pmatrix} = \begin{pmatrix} 0 \\ -gh\nabla b \end{pmatrix}, \tag{1}$$

where h denotes the water height and $\mathbf{v}$ is the velocity vector. The bottom topography is represented by b, g is the gravitational acceleration and I is the $d \times d$ identity matrix, where d is the number of space dimensions.

The shallow water equations have been widely discretized using finite volume (FV) schemes providing good results, see for instance [1]. First order FV schemes are widely found in the literature [2, 3, 4, 5, 6, 7, 8]. The conceptual simplicity of first order FV schemes and its rather straightforward implementation and ease for parallelization have made it very popular. However, the first order FV approach suffers from high sensitivity to the computational mesh

[9], thus requiring high mesh resolution to ensure accuracy. Higher order FV methods have been attempted to solve this issue. In many cases they remain at second order [10, 11, 12]. When higher order is achieved [13, 14, 15] they introduce a new problem, the need to enlarge the computational stencil and thus the loss of local support. In particular, this requires more data transfer on distributed memory architectures and, thus, affects the efficiency of parallel implementations.

Discontinuous Galerkin (DG) schemes have been recently applied to the shallow water equations [16, 17, 18, 19, 20, 21]. The DG method generalizes the concept of the FV method, while relying on the Finite Element notion of projecting the solution onto a space of trial functions, but, without the restriction of keeping the functions continuous. Therefore, the piecewise discontinuous description of the solution allows to achieve high order accuracy while keeping locality. Furthermore, the discontinuous interfaces at the cell edges requires the solution of Riemann problems employing approximate Riemann solvers [22] and related techniques [23, 24, 25] that have been widely studied in the context of FV schemes and shallow water equations.

Important issues relating to shallow water equations must also be addressed. In particular, a DG scheme for the shallow water equations must be well-balanced [26, 27, 28] and positivity-preserving [29], i.e., it must be able (i) to keep steady states over non-constant topography, and (ii) to keep positive values of depth, specially near the wet/dry front. It is also of great importance to have a sufficiently high resolution representation of the topography at very large gradients [9]. This has led to the need of not only properly discretizing the bed source term, but to build meshes that can harbor the required information with precision.

The high order nature of DG methods allows for accurate solutions on coarser meshes than first order FV schemes. However, the computational overhead introduced by the required operations for DG, as well as the increased number of time steps required due to a lower CFL number can make DG very expensive even compared to a FV approximation on a fine mesh. Therefore, in order to improve efficiency, adaptive meshes can be designed that will refine on those regions in which the solution shows large gradients or even discontinuities and coarsen the mesh on those regions where the solution is smooth. In this way, high accuracy can be obtained while reducing computational effort. Furthermore, the high order approximation can also be used to represent the topography, thus also reducing the need of high resolution meshes.

In order to control local grid refinement numerous refinement indicators have been developed, for instance, they are based on interpolation error estimates of some key quantity using a priori knowledge of the solution. In [30, 31] Remacle et al. present a second order scheme for shallow water equations with anisotropic grid adaptivity on triangular meshes, but without adressing well-balancing and positivity-preserving. Since these concepts offer no reliable error control, a priori as well as a posteriori error estimates have been developed to control the adaptive process, e.g., Bey and Oden [32], Adjerid et al. [33], Houston et al. [34, 35, 36], Dedner et al. [37], and recently Mavriplis et al. [38].

These approaches are aiming at estimating the error of the solution. Since there are in general no rigorous error estimates available for nonlinear systems of conservation laws, the concept of *multiresolution-based grid adaptation* has been developed in the context of FV schemes [39, 40]. The idea of a multiresolution representation of data originates from the field of wavelets [41, 42, 43, 44]. The starting point for a multiresolution analysis is a hierarchy of nested grids at different resolutions. Consider two different levels of resolution in this hierarchy, the cell averages on the finer level can be represented as a combination of the cell averages on the coarser level and of *detail coefficients* encoding the difference between the two levels. Since detail coefficients may become negligibly small in regions where the underlying data are locally smooth one can locally decide if the additional resolution of the finer mesh is needed. If these details are sufficiently small the solution can be represented on the coarser grid without significant loss of accuracy. Therefore the size of these details is a good indicator for adaptivity.

Multiscale-based mesh adaptivity has been quite successful with FV solvers for compressible fluid flow, cf. [45, 46, 47, 48]. In recent years the concept was extended to the framework of DG schemes. Originally in [49, 50] the concept was analytically and numerically investigated for the one-dimensional scalar case. The method was then extended to the Euler equations [51] and later on to the multi-dimensional case [52]. We track this idea and apply it to shallow water equations and add modifications in order to address common problems in solving shallow water equations, i.e., make sure that the presented grid adaptation is well-balanced and positivity-preserving. In [53] multiresolution-based mesh adaptation for shallow water flows in the context of FV schemes is discussed. Additionally we address the problem of wetting and drying. For shallow water modeling, the mesh must adapt to both the solution and the topography, since the latter affects significantly the first. This is also treated in the method proposed herein. In addition, we use the multiresolution analysis of the data in the grid adaptation as an additional indicator for limiting

to avoid limiting in areas, where the solution is smooth.

This paper is structured as follows. Section 2 provides a brief description of the DG method. Well-balancing and positivity-preserving is described in Section 3. In Section 4 we introduce the multiresolution analysis using multiwavelets and describe the grid adaptation process. Section 5 shows numerical results for six benchmarks and test cases. We present several one-dimensional and two-dimensional test cases. A summary of conclusions is presented in Section 6.

2. A review of the reference DG scheme

In this section we briefly summarize the main ideas of the Runge-Kutta DG (RKDG) schemes applied on a fixed grid, in Section 4 we will combine these methods with our dynamic grid adaptation. For this purpose the shallow water equations (1) are written in conservative form

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot \mathbf{f}(\mathbf{u}) = \mathbf{s}(\mathbf{x}, \mathbf{u}) \tag{2}$$

in the space-time domain $\Omega \times (0, T)$ with $\Omega \subset \mathbb{R}^d$, $d = 1, 2$, and $T > 0$ describing the evolution of the quantities $\mathbf{u} : \Omega \times [0, T] \to \mathbb{R}^m$ with the flux $\mathbf{f} : \mathbb{R}^m \to \mathbb{R}^{m \times d}$, $m = d + 1$, and suffering forcing by the source term $\mathbf{s}$. This problem has to be supplemented with initial data

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega$$

and suitable boundary conditions, e.g., periodic conditions or Dirichlet conditions at the inflow part.

2.1. Weak formulation

The entropy solution is approximated by a DG scheme following the idea of the method of lines: for the spatial discretization a DG method is used and the resulting system of ordinary differential equations in time is solved by a Runge-Kutta (RK) scheme. A detailed description can be found in [54] and [55]. Since computations are only performed on bounded domains, we consider in the following a finite discretization. For this purpose we discretize the domain Ω by a finite number of cells V_λ such that

$$\Omega = \bigcup_{\lambda \in \mathcal{I}} \overline{V_\lambda}, \quad \text{with} \quad V_\lambda \cap V_\mu = \emptyset, \quad \lambda \neq \mu \in \mathcal{I}.$$

Here the index set $\mathcal{I}$ characterizes the numbering of the cells. We call $\mathcal{G} := \{V_\lambda\}_{\lambda \in \mathcal{I}}$ the computational grid. On this grid we introduce the DG space

$$S := \{f \in L^2(\Omega) : \ f|_{V_\lambda} \in \Pi_{p-1}(V_\lambda) \ \forall \ \lambda \in \mathcal{I}\}, \tag{3}$$

where $\Pi_{p-1}(V)$ denotes the space of polynomials on the element V_λ of total degree less than p. For the discretization space S we introduce a set of basis functions $\Phi := \{\varphi_{\lambda,i}\}_{\lambda \in \mathcal{I}, i \in \mathcal{P}}$, which are orthonormalized with respect to L_2 and are compactly supported, i.e., supp $\varphi_{\lambda,i} = V_\lambda$, $i \in \mathcal{P}$, where $\mathcal{P}$ characterizes the numbering of the degrees of freedom of Π_{p-1}. We assume that the zeroth basis function is constant, i.e., $\varphi_{\lambda,0} = 1/\sqrt{|V_\lambda|}$. The global degrees of freedom are characterized by $\mathcal{I}^S := \mathcal{I} \times \mathcal{P}$, hence, we may write $\Phi = \{\varphi_{\mathbf{i}}\}_{\mathbf{i} \in \mathcal{I}^S}$.

For the semi-discretization of (2), we now assume that the approximate solution $\mathbf{u}_h = (h_h, (h\mathbf{v})_h)^t$ can be written as an expansion of the basis Φ, i.e.,

$$\mathbf{u}_h(\cdot, t) := \sum_{\mathbf{i} \in \mathcal{I}^S} \mathbf{u}_{\mathbf{i}}(t) \varphi_{\mathbf{i}}(\cdot) \in S^m, \quad \text{where} \quad \mathbf{u}_{\mathbf{i}}(t) = \langle \mathbf{u}_h(\cdot, t), \varphi_{\mathbf{i}} \rangle_\Omega. \tag{4}$$

Here $\langle f, g \rangle_\Omega := \int_\Omega f(\mathbf{x})\, g(\mathbf{x})\, d\mathbf{x}$ denotes the standard L^2-inner product. Due to the orthogonality of the basis functions the cell average $\overline{\mathbf{u}}_\lambda$ on V_λ is given by

$$\overline{\mathbf{u}}_\lambda := \frac{1}{|V_\lambda|} \int_{V_\lambda} \mathbf{u}_h \, d\mathbf{x} = \frac{\mathbf{u}_{\lambda,0}}{\sqrt{|V_\lambda|}} = \mathbf{u}_{\lambda,0} \varphi_{\lambda,0}. \tag{5}$$

In order to derive an evolution equation for the coefficients $\mathbf{u_i}$ we first rewrite (2) in a weak formulation. For this purpose we multiply (2) by a test function $\mathbf{w}_h \in S^m$ and integrate over a cell V_λ. Then we perform integration by parts and introduce numerical fluxes $\hat{\mathbf{f}}$ approximating the flux $\mathbf{f} \cdot \vec{n}$ in direction $\vec{n}$. This results in the semi-discrete DG-formulation:
Find $\mathbf{u}_h(\cdot, t) \in S^m, t \in [0, T]$, such that for all $\mathbf{w}_h \in S^m$ and $\lambda \in \mathcal{I}$:

$$\int_{V_\lambda} \frac{\partial \mathbf{u}_h}{\partial t} \cdot \mathbf{w}_h \, d\mathbf{x} - \int_{V_\lambda} \mathbf{f}(\mathbf{u}_h) : \nabla \mathbf{w}_h + \mathbf{s}(\mathbf{x}, \mathbf{u}_h) \cdot \mathbf{w}_h \, d\mathbf{x} + \int_{\partial V_\lambda} \hat{\mathbf{f}}(\mathbf{u}_h^+, \mathbf{u}_h^-, \vec{n}_\lambda) \cdot \mathbf{w}_h \, dS = 0, \tag{6}$$

where $\mathbf{a} \cdot \mathbf{b}, \mathbf{a}, \mathbf{b} \in \mathbb{R}^m$ denotes the standard inner product and for $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times d}$ we define the product $\mathbf{A} : \mathbf{B} := \sum_{i,j} A_{ij} B_{ij}$. Here $\mathbf{u}_h^+$ denotes the inner and $\mathbf{u}_h^-$ the outer value of $\mathbf{u}_h$ at the boundary of V_λ. Furthermore $\vec{n}_\lambda$ is the outward pointing unit normal vector corresponding to the boundary of V_λ. In our computations we choose the local Lax-Friedrichs flux [55] for reasons of simplicity and low computational cost:

$$\hat{\mathbf{f}}(\mathbf{u}^+, \mathbf{u}^-, \vec{n}) := \frac{1}{2} \left(\mathbf{f}(\mathbf{u}^+)\vec{n} + \mathbf{f}(\mathbf{u}^-)\vec{n} \right) - \alpha_\lambda \left(\mathbf{u}^+ - \mathbf{u}^- \right),$$

where α_λ is the maximum of the largest eigenvalue of $\mathbf{f}'(\mathbf{u}^+) \cdot \vec{n}$ and $\mathbf{f}'(\mathbf{u}^-) \cdot \vec{n}$, where $\mathbf{f}'$ is the Jacobian of $\mathbf{f}$.

By choosing the basis functions in Φ for the test functions in (6) we obtain a system of ordinary differential equations in time for the coefficients $\mathbf{u_i}$ in the basis representation (4) of the solution $\mathbf{u}_h$. To obtain a fully-discrete method, we have to discretize this system in time. For this purpose we use strong-stability-preserving Runge-Kutta methods [56] in our computations in Section 5. Thus, at time level t_n we may write

$$\mathbf{u}^n(\mathbf{x}) := \sum_{\mathbf{i} \in \mathcal{I}^S} \mathbf{u}_\mathbf{i}^n \varphi_\mathbf{i}(\mathbf{x}) := \mathbf{u}_h(\mathbf{x}, t_n)$$

and $\mathbf{u}^n = (h^n, (h\mathbf{v})^n)^t$.

Finally we have to apply a limiter to $\mathbf{u}_\mathbf{i}^n$ to avoid instabilities near discontinuities. Here the limiting is performed on the local characteristic variables to avoid spurious oscillations [57]. In each direction we apply the TVB limiter by Cockburn and Shu [58]. The idea behind this limiter is based on the assumption, that spurious oscillations only occur in the approximate solution if they occur in its Π_1-part. Therefore, the limiting is only applied on the Π_1-part of the solution. If the solution is limited all higher order coefficients will be set to zero after limiting the slopes from its Π_1-part. In the adaptive case we add a multiresolution based indicator to this limiting strategy.

3. Well-balancing and positivity-preserving

A common problem in discretizing shallow water equations with bottom topography is to preserve quiescent flow conditions, or the so-called *lake at rest*, where

$$h + b = const \qquad \text{and} \qquad \mathbf{v} = \mathbf{0}.$$

A scheme that can deal with this situation is said to satisfy the exact C-property according to Bermúdez and Vázquez [26]. It is well-known that a straightforward discretization of the equations (1) leads to non-physical oscillations caused by spurious waves traveling in the solution, thus, the discretization cannot preserve quiescent flow conditions. Therefore, the DG scheme has to be modified to preserve the exact C-property.

Furthermore also steady state flows are of interest. When the numerical scheme can satisfy this, it is said to be *well-balanced* [27, 28] in the sense that fluxes and source terms are properly balanced to keep steady states. In order to get a well-balanced scheme we follow the idea of Xing and Shu [59]. For this purpose we modify the numerical fluxes in (6) and project the bottom topography on the DG space, i.e., $b_h \in S$. The numerical fluxes are replaced by the modified flux

$$\int_{\partial V_\lambda} \hat{\mathbf{f}}_\lambda(\mathbf{u}_h^+, \mathbf{u}_h^-, b_h^+, b_h^-, \vec{n}_\lambda) \cdot \mathbf{w}_h \, dS,$$

which is defined by

$$\hat{\mathbf{f}}_\lambda(\mathbf{u}^+, \mathbf{u}^-, b^+, b^-, \vec{n}) := \hat{\mathbf{f}}(\mathbf{u}_m^+, \mathbf{u}_m^-, \vec{n}) + \frac{g}{2}\left((h^+)^2 - (h_m^+)^2\right)\begin{pmatrix}0\\ \vec{n}\end{pmatrix}. \tag{7}$$

The modified values are defined by

$$\mathbf{u}_m^\pm := (h_m^\pm, (h_m^\pm \mathbf{v}^\pm))^t, \quad \text{where} \quad h_m^\pm := \max(0, h^\pm + b^\pm - \max(b^+, b^-)).$$

In case of a fixed grid, Xing and Shu proved in [59] that the resulting scheme fulfills the exact C-property.

An additional crucial problem in discretizing shallow water equations is the treatment of wetting and drying. For that matter one has to make sure that the water height h remains positive during the computation. In order to ensure positivity we follow the ideas of Xing et al. [60] and Gandham et al. [61]. Xing et al. showed that applying a positivity correction for high order coefficients with a linear scaling around the cell average $\overline{\mathbf{u}}_\lambda^n := \frac{1}{|V_\lambda|}\int_{V_\lambda} \mathbf{u}^n \, d\mathbf{x} = \mathbf{u}_{\lambda,0}^n \, \varphi_{\lambda,0}$ after each forward Euler step is sufficient to guarantee positivity. This scaling is defined by

$$\tilde{\mathbf{u}}^n(\mathbf{x}) := \sum_{\lambda \in \mathcal{I}} \theta_\lambda \left(\sum_{i \in \mathcal{P}\setminus\{0\}} \mathbf{u}_{\lambda,i}^n \varphi_{\lambda,i}(\mathbf{x}) \right) + \overline{\mathbf{u}}_\lambda^n, \tag{8}$$

where θ_λ is defined by

$$\theta_\lambda := \min\left\{1, \frac{\overline{h}_\lambda^n}{\overline{h}_\lambda^n - h_{\min,\lambda}}\right\} \qquad \text{and} \qquad h_{\min,\lambda} := \min_{\mathbf{x}_i \in X_\lambda} h^n(\mathbf{x}_i). \tag{9}$$

Since in the DG scheme flux integrals over the cell volume and the cell interfaces have to be computed with suitable quadrature rules, we use these (volume and surface) quadrature points for the control points $\mathbf{x}_i \in X_\lambda$. By this we can make sure that h remains positive in all control points involved in the flux computations. Furthermore, a cell is considered to be dry if the mean value of h is smaller than the dry tolerance Tol. Note that for cells which are dry and all of its neighbours are dry as well no update in the DG scheme has to be performed.

Since the local projection limiter has to be applied after each stage in the Runge-Kutta scheme, we can simply add this positivity correction to the limiting procedure. In order to maintain the well-balancing property we apply the limiter in the local characteristic variables in non-dry areas, i.e., $h_{\min,\lambda} > Tol$, on $(h + b, h\mathbf{v})$ instead of $(h, h\mathbf{v})$. In dry or nearly dry areas, i.e, $h_{\min,\lambda} < Tol$, the limiting is performed on $h, h\mathbf{v}$ in the conserved variables itself instead of local characteristic variables. In the following algorithm we summarize the limiting procedure:

Algorithm 1. *(Limiting including positivity correction)*
For each cell V_λ, $\lambda \in \mathcal{I}$, the limiting step consists of the following two steps:

(i) If ($\min_{\mathbf{x}_i \in X_\lambda} h^n(\mathbf{x}_i) > Tol$)
apply TVD limiter in the local characteristic variables from [58] on $(h + b, h\mathbf{v})^t$
else
project $u|_{V_\lambda}$ to Π_1 and apply the TVD limiter in the conserved variables from [58] on $(h, h\mathbf{v})^t$ with Shu constant $M = 0$ on this cell.

(ii) Compute θ_λ and apply the positivity correction (8)*, i.e., multiply θ_λ to the higher order coefficients to ensure non-negative water height at all quadrature nodes.*

Note that the order in which the cells are limited does not matter, since the limiter relies only on the mean values of the neighbouring cells that are not changed by the limiter or the positivity correction. Additionally we set $\mathbf{v}$ to zero if $h < Tol$, since there should be physically no high velocity in almost dry areas. Otherwise the velocity might become very large because of division by a small value for h. In order to ensure the convergence of the scheme, we have to make sure that $Tol \to 0$ for $\max_{\lambda \in \mathcal{I}} |V_\lambda| \to 0$.

4. Multiresolution analysis and grid adaptation

The DG discretization typically works on an array of coefficients of the polynomial representation of the solution. In order to realize a certain target accuracy at the expense of a possibly low number of degrees of freedom, viz. a possibly low computational effort, one should keep the size of the cells large wherever the data exhibit little variation, reflecting a high regularity of the searched solution components. Our analysis of the local regularity behavior of the data is based on the concept of multiwavelets, cf. [62]. Here we will briefly summarize the basic ideas of the MRA concept, for details we refer to [52]. The core ingredients are (i) a hierarchy of nested grids, (ii) (orthogonal) multiwavelets and (iii) the multi-scale decomposition.

4.1. Nested grid hierarchy

The starting point for the multiresolution analysis is a coarse mesh $\mathcal{G}_0$ with N_0 cells. Then the cells of the mesh $\mathcal{G}_0$ are subdivided recursively and we obtain grids $\mathcal{G}_\ell$ for increasing number of refinement levels ℓ:

$$\mathcal{G}_\ell := \{V_\lambda\}_{\lambda \in \mathcal{I}_\ell} \quad \text{with} \quad \Omega = \bigcup_{\lambda \in \mathcal{I}_\ell} \overline{V_\lambda}, \quad \ell \in \mathbb{N}_0,$$

Obviously, the resulting grid hierarchy is nested:

$$\overline{V_\lambda} = \bigcup_{\mu \in \mathcal{M}_\lambda} \overline{V_\mu}, \qquad \forall\, \lambda \in \mathcal{I}_\ell,\ \ell \in \mathbb{N}_0, \tag{10}$$

where $\mathcal{M}_\lambda \subset \mathcal{I}_{\ell+1}$ denotes the *refinement set* of cell V_λ. The concept we present in this section can be applied to general nested grid hierarchies. For the two-dimensional computations in Section 5 we use on each level a uniform quadrilateral grid with $|M_\lambda| = 4$. By h_ℓ we denote the length of the minimum side of the rectangles on level ℓ. For more general grids h_ℓ can be chosen as the minimum diameter of cells V_λ on level ℓ.

4.2. Multiresolution analysis

Similar to (3) we introduce for each grid $\mathcal{G}_\ell$ the discretization space

$$S_\ell := \{f \in L^2(\Omega) :\ f|_{V_\lambda} \in \Pi_{p-1}(V_\lambda)\ \forall\ \lambda \in \mathcal{I}_\ell\}, \quad \ell \in \mathbb{N}_0$$

of piecewise polynomial functions of total degree less than p. To this multiresolution sequence we apply the concept of *multiresolution analysis* [41]. Due to the nestedness of the grid hierarchy these spaces form a *multiresolution sequence* $\mathcal{S}$ which is a nested sequence of closed linear subspaces of $L^2(\Omega)$, i.e.,

$$S_0 \subset S_1 \subset \ldots \subset S_\ell \subset S_{\ell+1} \subset \ldots \subset L^2(\Omega) \tag{11}$$

such that $\mathcal{S}$ is dense in $L^2(\Omega)$. The nestedness of the discretization spaces implies that there exist orthogonal complement spaces W_ℓ such that $S_{\ell+1} = S_\ell \oplus W_\ell$. Recursively applying this two-scale relation we obtain:

$$S_L = S_0 \oplus W_0 \oplus \ldots \oplus W_{L-1} \tag{12}$$

for arbitrary but fixed highest refinement level $L \in \mathbb{N}_0$. This multi-scale relation is the core ingredient of our adaptation strategy, because the complement spaces reveal insight into the *local* regularity behavior of the solution that can be exploited to trigger the grid refinement process.

To perform the multi-scale decomposition of the single-scale space S_L in (12) we need to specify a bases for the single-scale spaces S_ℓ and the complement spaces W_ℓ. Then the multi-scale decomposition corresponds to a change of basis. With regard to an efficient transformation we assume that the basis functions are compactly supported and orthonormal. We may write these spaces as

$$S_\ell = \operatorname*{span}_{\mathbf{i} \in \mathcal{I}_\ell^S} \varphi_{\mathbf{i}} \qquad \text{and} \qquad W_\ell = \operatorname*{span}_{\mathbf{i} \in \mathcal{I}_\ell^W} \psi_{\mathbf{i}}$$

with index sets $\mathcal{I}_\ell^S := \mathcal{I}_\ell \times \mathcal{P}$ and $\mathcal{I}_\ell^W := \mathcal{I}_\ell \times \mathcal{P}^*$ corresponding to the global degrees of freedom of S_ℓ and W_ℓ, respectively. $\mathcal{P}$ and $\mathcal{P}^*$ characterize the local degrees of freedom of S_ℓ and W_ℓ, respectively. The functions $\varphi_{\mathbf{i}}$ and $\psi_{\mathbf{i}}$ are referred to as *scaling functions* and *multiwavelets*, respectively.

By definition of the complement spaces W_ℓ the wavelets are orthogonal to the scaling functions:

$$\langle \psi_{\mathbf{i}}, \varphi_{\mathbf{j}} \rangle_\Omega = 0 \qquad \mathbf{i} \in \mathcal{I}_\ell^W, \ \mathbf{j} \in \mathcal{I}_\ell^S, \quad \ell \in \mathbb{N}_0. \tag{13}$$

The scaling functions can be constructed by the Gram-Schmidt process. The wavelets can be constructed with Alpert's general construction principle [63]. For a detailed description of the construction of multi-dimensional wavelets we refer to [52]. The following proposition summarizes the properties of the basis functions:

Proposition 1. *(Properties of the basis)*
For $\mathbf{i}, \mathbf{j} \in \mathcal{I}_\ell^S$, $\mathbf{i}', \mathbf{j}' \in \mathcal{I}_\ell^W$, $\lambda \in \mathcal{I}_\ell$, $i \in \mathcal{P}$ *and* $i' \in \mathcal{P}^*$ *there holds:*

(i) Orthonormality:

$$\langle \varphi_{\mathbf{i}}, \varphi_{\mathbf{j}} \rangle_\Omega = \delta_{\mathbf{ij}}, \quad \langle \psi_{\mathbf{i}'}, \psi_{\mathbf{j}'} \rangle_\Omega = \delta_{\mathbf{i}'\mathbf{j}'} \quad \textit{and} \quad \langle \varphi_{\mathbf{i}}, \psi_{\mathbf{i}'} \rangle_\Omega = 0. \tag{14}$$

(ii) Vanishing moments:

$$\langle \psi_{\mathbf{i}'}, P \rangle_\Omega = 0 \ \ \forall\, P \in \Pi_{p-1}(\Omega). \tag{15}$$

(iii) Compact support:

$$supp(\varphi_{\lambda,i}) = V_\lambda \quad \textit{and} \quad supp(\psi_{\lambda,i'}) = V_\lambda. \tag{16}$$

(iv) The zero order basis function is constant:

$$\varphi_{\lambda,0} = 1/\sqrt{|V_\lambda|}. \tag{17}$$

(v) Due to the nestedness of the spaces there exist the following two-scale relations:

$$\varphi_{\lambda,i} = \sum_{\mu \in \mathcal{M}_\lambda} \sum_{j \in \mathcal{P}} \langle \varphi_{\mu,j}, \varphi_{\lambda,i} \rangle_{V_\mu} \ \varphi_{\mu,j}, \qquad \psi_{\lambda,i'} = \sum_{\mu \in \mathcal{M}_\lambda} \sum_{j \in \mathcal{P}} \langle \varphi_{\mu,j}, \psi_{\lambda,i'} \rangle_{V_\mu} \ \varphi_{\mu,j} \tag{18}$$

and

$$\varphi_{\mu,i} = \sum_{j \in \mathcal{P}} \langle \varphi_{\lambda,j}, \varphi_{\mu,i} \rangle_{V_\mu} \ \varphi_{\lambda,j} + \sum_{j' \in \mathcal{P}^*} \langle \psi_{\lambda,j'}, \varphi_{\mu,i} \rangle_{V_\mu} \ \psi_{\lambda,j'}, \quad \mu \in \mathcal{M}_\lambda. \tag{19}$$

4.3. Multi-scale transformation

In order to exploit the potential of data compression we make use of the multi-scale transformation (12), and determine the relation between the coefficients in the single-scale and the multi-scale representation of $\mathbf{u} \in (S_L)^m$:

$$\mathbf{u} = \sum_{\mathbf{i} \in \mathcal{I}_L^S} \mathbf{u}_{\mathbf{i}} \varphi_{\mathbf{i}} = \sum_{\mathbf{i} \in \mathcal{I}_0^S} \mathbf{u}_{\mathbf{i}} \varphi_{\mathbf{i}} + \sum_{\ell=0}^{L-1} \sum_{\mathbf{i} \in \mathcal{I}_\ell^W} \mathbf{d}_{\mathbf{i}} \psi_{\mathbf{i}}. \tag{20}$$

From the orthonormality of the basis (14) we conclude that the single-scale coefficients $\mathbf{u}_{\mathbf{i}}$ and the detail coefficients $\mathbf{d}_{\mathbf{i}}$ are determined as the inner products of $\mathbf{u}$ with the basis functions, i.e.,

$$\mathbf{u}_{\mathbf{i}} = \langle \mathbf{u}, \varphi_{\mathbf{i}} \rangle_\Omega \quad \text{and} \quad \mathbf{d}_{\mathbf{i}} = \langle \mathbf{u}, \psi_{\mathbf{i}} \rangle_\Omega. \tag{21}$$

Recursively, the array of single-scale coefficients corresponding to the finest discretization level

$$\mathbf{U}_L := (\mathbf{u}_\mathbf{i})_{\mathbf{i}\in\mathcal{I}_L^S}$$

can be transformed into a sequence of coarse grid data $\mathbf{U}_0$ and details $\mathbf{D}_l$

$$\mathbf{U}_0 := (\mathbf{u}_\mathbf{i})_{\mathbf{i}\in\mathcal{I}_0^S}, \quad \mathbf{D}_\ell := (\mathbf{d}_\mathbf{i})_{\mathbf{i}\in\mathcal{I}_\ell^W}, \quad \ell = 0,\dots,L-1,$$

where the arrays of the details represent the successive update from a coarser resolution to a finer resolution. We refer to this transformation as the *multi-scale transformation* determined by the *multi-scale operator* $\mathcal{M}_L : \mathbf{U}_L \longrightarrow (\mathbf{U}_0, \mathbf{D}_0, \dots, \mathbf{D}_{L-1})$. The *inverse multi-scale transformation* is described by the *inverse multi-scale operator* $\mathcal{M}_L^{-1} : (\mathbf{U}_0, \mathbf{D}_0, \dots, \mathbf{D}_{L-1}) \longrightarrow \mathbf{U}_L$. From the two-scale relations for the basis functions (18) and (19) we can derive similar relations for the coefficients: the multi-scale transformation can be implemented very efficiently by recursively applying:

$$\mathbf{u}_{\lambda,i} = \sum_{\mu\in\mathcal{M}_\lambda}\sum_{j\in\mathcal{P}}\langle\varphi_{\mu,j},\varphi_{\lambda,i}\rangle_{V_\mu}\,\mathbf{u}_{\mu,j}, \quad \lambda\in\mathcal{I}_\ell,\ i\in\mathcal{P} \tag{22}$$

and

$$\mathbf{d}_{\lambda,i} = \sum_{\mu\in\mathcal{M}_\lambda}\sum_{j\in\mathcal{P}}\langle\varphi_{\mu,j},\psi_{\lambda,i}\rangle_{V_\mu}\,\mathbf{u}_{\mu,j}, \quad \lambda\in\mathcal{I}_\ell,\ i\in\mathcal{P}^* \tag{23}$$

proceeding from fine to coarse, i.e., $\ell = L-1,\dots,0$. The inverse multi-scale transformation can be implemented analogously by recursively applying:

$$\mathbf{u}_{\mu,i} = \sum_{j\in\mathcal{P}}\langle\varphi_{\lambda,j},\varphi_{\mu,i}\rangle_{V_\mu}\,\mathbf{u}_{\lambda,j} + \sum_{j'\in\mathcal{P}^*}\langle\psi_{\lambda,j'},\varphi_{\mu,i}\rangle_{V_\mu}\,\mathbf{d}_{\lambda,j'}, \quad \mu\in\mathcal{M}_\lambda,\ \lambda\in\mathcal{I}_\ell,\ i\in\mathcal{P} \tag{24}$$

proceeding from coarse to fine, i.e., $\ell = 0,\dots,L-1$. In the case of Cartesian grids the inner products of the basis functions in the transformations, e.g., $\langle\varphi_{\mu,j},\varphi_{\lambda,i}\rangle_{V_\mu}$, can be computed on a reference element once before the computation and stored to reduce the computational cost of these transformations. Therefore the computation of the transformation can be done very efficiently. For a detailed description we refer to [52].

4.4. Thresholding and grid adaptation

We call a detail coefficient $\mathbf{d}_\mathbf{i}$ *significant* if it is larger than a level-dependent threshold value:

$$d_\mathbf{i}^{\max} := \max_{1\le k\le m}\frac{|(\mathbf{d}_\mathbf{i})_k|\,\|\psi_\mathbf{i}\|_{L_\infty(\Omega)}}{\overline{u}_k^{\max}} > \varepsilon_{\ell(\mathbf{i})}, \tag{25}$$

where $\ell(\mathbf{i})$ denotes the level of the cell corresponding to the index $\mathbf{i}$. In order to account for the different orders of magnitude of the conserved quantities it is scaled with $\overline{u}_k^{\max} := \max(\max_{\lambda\in\mathcal{G}^n}|(\overline{\mathbf{u}}_\lambda)_k|, 1)$, where $\overline{\mathbf{u}}_\lambda$ is the mean value of $\mathbf{u}$ on V_λ according to (5).

For a given threshold value $\varepsilon \ge 0$ we define level-dependent threshold values ε_ℓ by $\varepsilon_\ell := a^{\ell-L}\varepsilon$, $a > 1$. Due to the cancellation property detail coefficients may become negligibly small whenever the underlying function is locally smooth, for details we refer to [52]. This gives rise to *hard thresholding* characterized by the set of cell indices

$$\mathcal{D}_\varepsilon := \left\{\mathbf{i}\in\bigcup_{\ell=0}^{L-1}\mathcal{I}_\ell^W : d_\mathbf{i}^{\max} > \varepsilon_{\ell(\mathbf{i})}\right\}. \tag{26}$$

We will use this index set to find a sparser approximate representation of $\mathbf{u}$. Therefore we define the approximation

$$\mathbf{u}_\varepsilon := \sum_{\mathbf{i}\in\mathcal{I}_0^S}\mathbf{u}_\mathbf{i}\varphi_\mathbf{i} + \sum_{\mathbf{i}\in\mathcal{D}_\varepsilon}\mathbf{d}_\mathbf{i}\psi_\mathbf{i}. \tag{27}$$

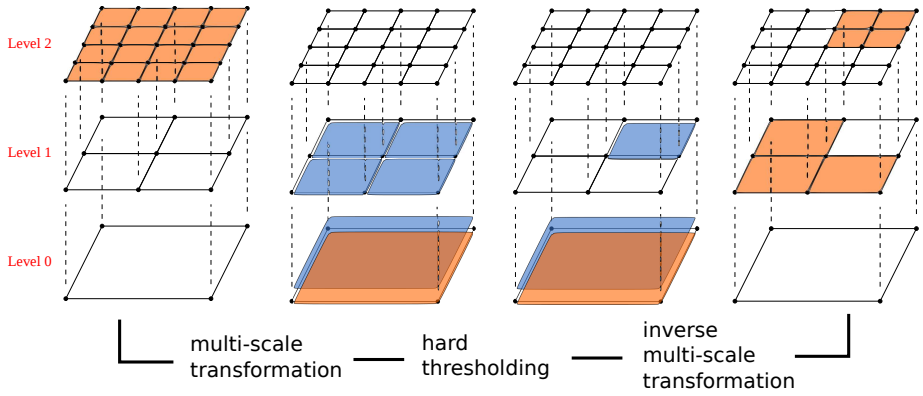


Figure 1. Example for coarsening via thresholding.

This approximation may be seen as a projection to a coarser grid if $\mathcal{D}_\varepsilon$ is a tree, i.e.,

$$(\mu, i) \in \mathcal{D}_\varepsilon \Rightarrow (\lambda, j) \in \mathcal{D}_\varepsilon \;\; \forall\, \lambda, V_\mu \subsetneq V_\lambda \text{ and } j \;\in \mathcal{P}^*.$$

Thus we determine a locally refined grid $\mathcal{G}_\varepsilon$ from the index set $\mathcal{D}_\varepsilon$: First of all we modify $\mathcal{D}_\varepsilon$ such that it is a tree. Then $\mathcal{G}_\varepsilon$ can be determined recursively by traversing through the levels from coarse to fine: if $(\lambda, i) \in \mathcal{D}_\varepsilon$ for some $i \in \mathcal{P}^*$, then the cell V_λ is locally refined:

$$\mathcal{G}_\varepsilon = \{\mu \in \mathcal{I}_\ell,\, \ell \in \{0, ..., L\} :\; (\mu, i) \notin \mathcal{D}_\varepsilon \;\forall\, i \in \mathcal{P}^* \text{ and } \exists\, i \in \mathcal{P}^*, \lambda \in \mathcal{I}_{\ell-1},\; \text{s.t. } \mu \in \mathcal{M}_\lambda \text{ and } (\lambda, i) \in \mathcal{D}_\varepsilon \,\}.$$

For an illustration of grid coarsening via hard thresholding see Figure 1. In the following proposition we summarize the properties of the thresholding.

Proposition 2. *(Properties of thresholding)*
Let Ω be a bounded domain, $\boldsymbol{u} \in (S_L)^m$ and $\varepsilon_\ell = a^{\ell-L}\varepsilon$ with $a > 1$. Furthermore let $\mathbf{u}_\varepsilon$ be the coarsened approximation corresponding to $\mathcal{D}_\varepsilon$.

(i) The approximation error with respect to the set of significant details $\mathcal{D}_\varepsilon$ is uniformly bounded with respect to $L^q(\Omega)$, $q \in [1, \infty)$, i.e.,

$$\| (\boldsymbol{u} - \boldsymbol{u}_\varepsilon)_k \|_{L^q(\Omega)} \leq \frac{\overline{u}_k^{max}\, |\mathcal{P}^*|\, |\Omega|^{\frac{1}{q}}}{a-1}\varepsilon, \quad 1 \leq k \leq m.$$

(ii) The average value $\overline{\boldsymbol{u}} := \frac{1}{|\Omega|}\int_\Omega \boldsymbol{u} d\boldsymbol{x}$ is not modified by thresholding, i.e.

$$\overline{\boldsymbol{u}} = \overline{\boldsymbol{u}}_\varepsilon.$$

(iii) The positivity of the water height in the mean is preserved by the thresholding procedure, i.e.,

$$\overline{h}_\lambda \geq 0, \quad \forall \lambda \in \mathcal{I}_L \quad \Rightarrow \quad \overline{h}_\lambda \geq 0, \quad \forall \lambda \in \mathcal{G}_\varepsilon,$$

where $\overline{\boldsymbol{u}}_\lambda = (\overline{h}_\lambda, (\overline{h\boldsymbol{v}})_\lambda)^t$ is the mean value on V_λ according to (5).

Proof: The index set for all detail coefficients is defined by

$$\mathcal{D} := \bigcup_{\ell=0}^{L-1} \mathcal{I}_\ell^W.$$

Then the threshold error $\mathbf{u} - \mathbf{u}_\varepsilon$ is given by

$$\mathbf{u}(\mathbf{x}) - \mathbf{u}_\varepsilon(\mathbf{x}) = \sum_{\mathbf{i} \in \mathcal{D} \setminus \mathcal{D}_\varepsilon} \mathbf{d}_\mathbf{i}\, \psi_\mathbf{i}(\mathbf{x}), \quad \mathbf{x} \in \Omega.$$

By the definition of $\mathcal{D}_\varepsilon$ there holds for $\mathbf{x} \in V_\lambda$, $\lambda \in \mathcal{I}_\ell$:

$$|\,(\mathbf{u}(\mathbf{x}) - \mathbf{u}_\varepsilon(\mathbf{x}))_k\,| \leq \sum_{\mathbf{i} \in \mathcal{D} \setminus \mathcal{D}_\varepsilon} |(\mathbf{d}_\mathbf{i})_k|\, |\psi_\mathbf{i}(\mathbf{x})| \leq \overline{u}_k^{max} \sum_{\mathbf{i} \in \mathcal{D} \setminus \mathcal{D}_\varepsilon} \varepsilon_{\ell(\mathbf{i})} \leq \overline{u}_k^{max} \sum_{\mathbf{i} \in \mathcal{D}} \varepsilon_{\ell(\mathbf{i})} \leq \overline{u}_k^{max}\, |\mathcal{P}^*| \sum_{\ell=0}^{L-1} \varepsilon_\ell.$$

Furthermore we have

$$\sum_{\ell=0}^{L-1} \varepsilon_\ell = \varepsilon a^{-L} \sum_{\ell=0}^{L-1} a^\ell = \varepsilon \frac{1 - a^{-L}}{a - 1} \leq \frac{1}{a-1}\varepsilon.$$

Property (i) follows directly by combining these two inequalities and the definition of the L^q-norm. Property (ii) holds true, since we conclude by (15)

$$\overline{\mathbf{u}} - \overline{\mathbf{u}}_\varepsilon = \frac{1}{|\Omega|} \langle \mathbf{u} - \mathbf{u}_\varepsilon, 1 \rangle_\Omega = \frac{1}{|\Omega|} \sum_{\mathbf{i} \in \mathcal{D} \setminus \mathcal{D}_\varepsilon} \mathbf{d}_\mathbf{i}\, \langle \psi_\mathbf{i}, 1 \rangle_\Omega = 0.$$

Since the cells in $\mathcal{G}_\varepsilon$ are not smaller than the cells in $\mathcal{I}_L$ it is sufficient for the proof of (iii) to show

$$\overline{h}_\mu \geq 0 \quad \forall\, \mu \in M_\lambda \quad \Rightarrow \quad \overline{h}_\lambda \geq 0, \quad \forall\, \lambda \in \bigcup_{\ell=1}^{L-1} \mathcal{I}_\ell.$$

First we note by (17) that

$$1 = \sqrt{|V_\lambda|}\, \varphi_{\lambda,0}(\mathbf{x}) = \sqrt{|V_\mu|}\, \varphi_{\mu,0}(\mathbf{x}), \quad \forall\, \mathbf{x} \in V_\mu \subset V_\lambda. \tag{28}$$

Then we conclude by (22) and (14) that

$$h_{\lambda,0} = \sum_{\mu \in M_\lambda} \sum_{i \in \mathcal{P}} \langle \varphi_{\mu,i}, \varphi_{\lambda,0} \rangle_{V_\mu}\, h_{\mu,i} = \sum_{\mu \in M_\lambda} \sum_{i \in \mathcal{P}} \frac{\sqrt{|V_\mu|}}{\sqrt{|V_\lambda|}}\, \langle \varphi_{\mu,i}, \varphi_{\mu,0} \rangle_{V_\mu}\, h_{\mu,i} = \sum_{\mu \in M_\lambda} \frac{\sqrt{|V_\mu|}}{\sqrt{|V_\lambda|}}\, h_{\mu,0} \geq 0.$$

□

Since we are interested both in a large threshold value resulting in sparse grids and in a small error we choose $a = 2$ for the computations in Section 5.

4.5. Refinement (prediction)

The multiresolution analysis is used as an indicator to perform local grid adaptation, where due the time evolution the adaptive grid may dynamically change. At each time step t_n the adaptive grid is characterized by an index set

$$\mathcal{G}^n \subset \{\lambda \,:\, \lambda \in \mathcal{I}_\ell,\ \ell = 0, \ldots, L\}$$

satisfying $\Omega = \bigcup_{\lambda \in \mathcal{G}^n} V_\lambda$ with $V_\lambda \cap V_{\lambda'} = \emptyset$ for $\lambda \neq \lambda'$. At each time step t_n associated to this grid there exists a discretization space $S_n \subset S_L$. Then any $\mathbf{u} \in (S_n)^m$ can be written as

$$\mathbf{u} = \sum_{\mathbf{i} \in \mathcal{G}^n \times \mathcal{P}} \mathbf{u}_\mathbf{i} \varphi_\mathbf{i} = \sum_{\mathbf{i} \in \mathcal{I}_0^S} \mathbf{u}_\mathbf{i} \varphi_\mathbf{i} + \sum_{\mathbf{i} \in \mathcal{D}^n} \mathbf{d}_\mathbf{i} \psi_\mathbf{i},$$

where the corresponding index set of detail coefficients is given by

$$\mathcal{D}^n := \left\{ \mathbf{i} = (\lambda, i) \in \bigcup_{\ell=0}^{L-1} \mathcal{I}_\ell^W \,:\, \exists \mu \in \mathcal{G}^n \text{ s. t.} V_\mu \subsetneq V_\lambda \right\}. \tag{29}$$

The adaptive grid can be characterized by the set of elements $\mathcal{G}^n$, the set of significant coefficients $\mathcal{D}^n$ or the corresponding space S_n equivalently. In between two time steps we have to determine the new grid for the next time step and project the DG-solution to the corresponding space. Thus, the adaptation strategy consists of two steps at each time step t_n:

- After having performed the update step of the DG-scheme on $(S_n)^m$, we apply thresholding and determine a coarsened grid $\mathcal{G}_\varepsilon^n$, the solution is projected to the corresponding space $(S_{n,\varepsilon})^m$.
- Before the next time step we refine the grid $\mathcal{G}_\varepsilon^n$ to determine $\mathcal{G}^{n+1}$ and project the DG-solution from $(S_{n,\varepsilon})^m$ to $(S_{n+1})^m$.

Based on $\mathbf{u} \in (S_{n,\varepsilon})^m$ at the current time step t_n we have to *predict* which details will be significant at the new time step t_{n+1} in order to refine the grid and determine $\mathcal{G}^{n+1}$. The refinement is characterized by the set $\mathcal{D}^{n+1}$, where we inflate the set $\mathcal{D}_\varepsilon^n$ corresponding to $\mathcal{G}_\varepsilon^n$. For the design of such a prediction set we modify the idea of Harten's strategy [39]:

(i) significant details may remain significant within one time step:

$$\mathcal{D}_\varepsilon^n \subset \mathcal{D}^{n+1}, \tag{30}$$

(ii) details in a *local* neighborhood of a significant detail may also become significant within one time step due to the *finite* speed of propagation, i.e.,

$$\max_{i\in\mathcal{P}^*} d_{\lambda,i}^{\max} > \varepsilon_\ell \quad \Rightarrow \quad \{(\lambda', i) : V_{\lambda'} \text{ is neighbour of } V_\lambda, i \in \mathcal{P}^*\} \subset \mathcal{D}^{n+1}, \tag{31}$$

(iii) gradients may steepen due to the formation of shocks causing significant details on higher levels, i.e.,

$$\max_{i\in\mathcal{P}^*} d_{\lambda,i}^{\max} \geq 2^{\bar{M}+1}\varepsilon_\ell \quad \Rightarrow \quad \mathcal{M}_\lambda \times \mathcal{P}^* \subset \mathcal{D}^{n+1} \tag{32}$$

where $\bar{M} = p$ is the number of vanishing moments of the multiwavelets,

(iv) $\mathcal{D}^{n+1}$ is a tree:

$$(\mu, i) \in \mathcal{D}^{n+1} \Rightarrow (\lambda, j) \in \mathcal{D}^{n+1} \ \ \forall\, \lambda, V_\mu \subsetneq V_\lambda \text{ and } j \ \in \mathcal{P}^*. \tag{33}$$

Analogously to Figure 1 we can determine the adaptive grid for the new time step $\mathcal{G}^{n+1}$ from $\mathcal{D}^{n+1}$ after the prediction step. If the grid is refined in the prediction step we have to project the current solution to $(S_{n+1})^m$. In general this can be done by applying a local inverse multi-scale transformation (24), where non-existing detail coefficients are set to zero. In order to ensure positivity of the mean values in the refined grid $\mathcal{G}^{n+1}$ we have to modify this projection, where we split the inverse multi-scale transformation in two parts. Due to (30) we know that $\mathcal{G}_\varepsilon^n$ is coarser than $\mathcal{G}^{n+1}$, therefore we may determine the single-scale representation of $\mathbf{u}$ on the grid $\mathcal{G}_\varepsilon^n$ from the previous time step first and then take care for the additional refinement added by (31), (32) and (33) in a second step:

First of all, we apply recursively for all λ, which satisfy $(\lambda, i) \in \mathcal{D}_\varepsilon^n$ for some $i \in \mathcal{P}^*$, the inverse multi-scale transformation (24) from coarse to fine to determine a single-scale representation of $\mathbf{u}$ on the grid $\mathcal{G}_\varepsilon^n$. In a second step we take care of the additional refinement. The cells which will be additionally refined are determined by

$$\mathcal{R}^{n+1} := \{\lambda : (\lambda, i) \in \mathcal{D}^{n+1} \setminus \mathcal{D}_\varepsilon^n \text{ for some } i \in \mathcal{P}^*\}.$$

The refinement is done recursively from coarse to fine since a cell might be refined again on the next refinement level. If a cell V_λ, $\lambda \in \mathcal{R}^{n+1}$, is additionally refined by the prediction step we have to project the DG-solution to its finer sub-cells $\mu \in M_\lambda$. Since $\mathbf{d_i} = 0$ for all $\mathbf{i} \in \mathcal{D}^{n+1} \setminus \mathcal{D}_\varepsilon^n$, the local inverse multi-scale transformation (24) provides us with

$$\mathbf{u}_{\mu,i} := \sum_{i'\in\mathcal{P}} \langle \varphi_{\lambda,i'}, \varphi_{\mu,i}\rangle_{V_\mu}\, \mathbf{u}_{\lambda,i'}, \quad \mu \in \mathcal{M}_\lambda, \quad \lambda \in \mathcal{R}^{n+1}. \tag{34}$$

However, this could result in negative mean values for the height h. Therefore we modify the inverse multi-scale transformation (34) by adding the following positivity correction to the higher order coefficients:

$$\tilde{\mathbf{u}}_{\mu,i} := \langle \varphi_{\lambda,0}, \varphi_{\mu,i}\rangle_{V_\mu} \tilde{\mathbf{u}}_{\lambda,0} + \alpha_\lambda \sum_{i'\in\mathcal{P}\setminus\{0\}} \langle \varphi_{\lambda,i'}, \varphi_{\mu,i}\rangle_{V_\mu}\, \tilde{\mathbf{u}}_{\lambda,i'}, \quad \mu \in \mathcal{M}_\lambda,\ \lambda \in \mathcal{R}^{n+1}, \tag{35}$$

where initially $\tilde{\mathbf{u}}_{\lambda,i} = \mathbf{u}_{\lambda,i}$ for all $\lambda \in \mathcal{G}_\varepsilon^n$ and $i \in \mathcal{P}$. The local correction factor α_λ is defined as

$$\alpha_\lambda := \min(1, \frac{\tilde{h}_{\lambda,0}\varphi_{\lambda,0}}{\Delta h_{\min}}), \quad \text{where} \quad \Delta h_{\min} := -\min_{\mu\in M_\lambda} \sum_{i'\in\mathcal{P}\setminus\{0\}} \langle \varphi_{\lambda,i'}, \varphi_{\mu,0}\rangle_{V_\mu} \frac{\tilde{h}_{\lambda,i'}}{\sqrt{|V_\mu|}}. \tag{36}$$

Note that (35) coincides with (34) for $\alpha_\lambda = 1$. The step is repeated recursively from coarse to fine until the desired local refinement level is reached. We summarize the procedure for the refinement including the modified projection (35) in the following algorithm.

Algorithm 2. *(Projection to refined grid via inverse multi-scale transformation)*

(0) Initialize: $\mathcal{G}^{n+1} = \mathcal{I}_0$

(1) For $\ell = 0$ *to* $L-1$ *do*
If $(\lambda, i) \in \mathcal{D}_\varepsilon^n$ *for some* $i \in \mathcal{P}^*$ *then*

(i) compute $\mathbf{u}_{\mu,i}$, $\mu \in \mathcal{M}_\lambda$, $i \in \mathcal{P}$, *according to* (24)*;*

(ii) add $\mu \in \mathcal{M}_\lambda$ *to the set* $\mathcal{G}^{n+1}$*;*

(iii) remove λ *from the set* $\mathcal{G}^{n+1}$.

(2) Initialize $\tilde{\mathbf{u}}_{\lambda,i} := \mathbf{u}_{\lambda,i}$ *for all* $\lambda \in \mathcal{G}_\varepsilon^n$ *and* $i \in \mathcal{P}$.

(3) For $\ell = 0$ *to* $L-1$ *do*
If $(\lambda, i) \in \mathcal{D}^{n+1} \setminus \mathcal{D}_\varepsilon^n$ *for some* $i \in \mathcal{P}^*$ *then*

(i) compute $\tilde{\mathbf{u}}_{\mu,i}$, $\mu \in \mathcal{M}_\lambda$, $i \in \mathcal{P}$, *according to* (35)*;*

(ii) add $\mu \in \mathcal{M}_\lambda$ *to the set* $\mathcal{G}^{n+1}$*;*

(iii) remove λ *from the set* $\mathcal{G}^{n+1}$.

Finally the modified projection $\tilde{\mathbf{u}}$ *onto the refined grid* $\mathcal{G}^{n+1}$ *is defined by* $\tilde{\mathbf{u}} := \sum_{\mathbf{i}\in\mathcal{G}^{n+1}\times\mathcal{P}} \tilde{\mathbf{u}}_\mathbf{i}\, \varphi_\mathbf{i}$.

In the following proposition we summarize the properties of our refinement strategy including the modified projection.

Proposition 3. *(Properties of refinement)*
Let Ω *be a bounded domain,* $\boldsymbol{u} \in (S_{n,\varepsilon})^m$. *Furthermore let* $\tilde{\boldsymbol{u}} \in (S_{n+1})^m$ *be the projection to the refined grid* $\mathcal{G}^{n+1}$ *using the corrected projection* (35).

(i) The corrected projection (35) *modifies the solution only in dry areas, i.e.,*

$$h(\mathbf{x}) \geq 0 \qquad \forall \mathbf{x} \in V_\lambda,\ \lambda \in \mathcal{G}_\varepsilon^n \quad \Rightarrow \quad \mathbf{u}(\mathbf{x}) = \tilde{\mathbf{u}}(\mathbf{x}) \quad \forall\, \mathbf{x} \in\ V_\lambda.$$

(ii) The average value $\overline{\boldsymbol{u}} = \frac{1}{|\Omega|}\int_\Omega \boldsymbol{u}d\boldsymbol{x}$ *is not modified by the refinement including the corrected projection* (35)*, i.e.,*

$$\overline{\boldsymbol{u}} = \overline{\tilde{\boldsymbol{u}}}.$$

(iii) The positivity of the water height in the mean is preserved by the refinement, i.e.,

$$\overline{h}_\lambda \geq 0, \quad \forall \lambda \in \mathcal{G}_\varepsilon^n \quad \Rightarrow \quad \overline{\tilde{h}}_\lambda \geq 0, \quad \forall \lambda \in \mathcal{G}^{n+1},$$

where $\overline{\boldsymbol{u}}_\lambda = (\overline{h}_\lambda, \overline{(h\boldsymbol{v})}_\lambda)^t$ *and* $\overline{\tilde{\boldsymbol{u}}}_\lambda = (\overline{\tilde{h}}_\lambda, \overline{\widetilde{(h\boldsymbol{v})}}_\lambda)^t$ *are the mean values according to definition* (5).

Proof: Since the refinement is done locally level-wise from coarse to fine, we assume for simplicity that a cell V_λ, $\lambda \in \mathcal{G}_\varepsilon^n$, is refined only once into its sub-cells V_μ, $\mu \in M_\lambda$. Otherwise we repeat the following arguments recursively to its sub-cells. In this case, according to (24) and (35), $\mathbf{u} = (h, h\mathbf{v})^t$ and $\tilde{\mathbf{u}} = (\tilde{h}, \widetilde{h\mathbf{v}})^t$ are defined on V_λ by

$$\mathbf{u}(\mathbf{x}) = \sum_{i\in\mathcal{P}} \mathbf{u}_{\lambda,i}\varphi_{\lambda,i}(\mathbf{x}), \quad \mathbf{x} \in V_\lambda \quad \text{and} \tag{37}$$

$$\tilde{\mathbf{u}}(\mathbf{x}) = \sum_{\mu\in\mathcal{M}_\lambda} \sum_{i\in\mathcal{P}} \tilde{\mathbf{u}}_{\mu,i}\varphi_{\mu,i}(\mathbf{x}), \quad \mathbf{x} \in V_\lambda. \tag{38}$$

Note that due to the refinement assumption $\tilde{\mathbf{u}}_{\lambda,i} = \mathbf{u}_{\lambda,i}$ for all $i \in \mathcal{P}$. Let $\overline{h}_\mu$ denote the mean values of h in the sub-cells V_μ, $\mu \in M_\lambda$. By (37), (17) and (28) we conclude

$$\overline{h}_\mu := \frac{1}{|V_\mu|}\int_{V_\mu} h(\mathbf{x})\, d\mathbf{x} = \frac{1}{|V_\mu|}\langle h, 1\rangle_{V_\mu} = \overline{h}_\lambda + \sum_{i'\in\mathcal{P}\setminus\{0\}} \langle\varphi_{\lambda,i'}, \varphi_{\mu,0}\rangle_{V_\mu}\, \frac{h_{\lambda,i'}}{\sqrt{|V_\mu|}}, \quad \mu \in \mathcal{M}_\lambda, \tag{39}$$

where by (17) and (21) the mean value of h on V_λ is determined by

$$\overline{h}_\lambda := \frac{1}{|V_\lambda|}\int_{V_\lambda} h(\mathbf{x})\, d\mathbf{x} = \frac{1}{|V_\lambda|}\langle h, 1\rangle_{V_\lambda} = h_{\lambda,0}\varphi_{\lambda,0}. \tag{40}$$

By definition (36) and $h_{\lambda.0} = \tilde{h}_{\lambda,0}$ we conclude that

$$\Delta h_{\min} = \overline{h}_\lambda - \min_{\mu\in\mathcal{M}_\lambda} \overline{h}_\mu. \tag{41}$$

In order to show (i) we first note by (34) that for all $\mathbf{x} \in V_\lambda$ there holds

$$\mathbf{u}(\mathbf{x}) = \sum_{\mu\in\mathcal{M}_\lambda} \sum_{i\in\mathcal{P}} \mathbf{u}_{\mu,i}\varphi_{\mu,i}(\mathbf{x}), \quad \text{where} \quad \mathbf{u}_{\mu,i} := \sum_{i'\in\mathcal{P}} \langle\varphi_{\lambda,i'}, \varphi_{\mu,i}\rangle_{V_\mu}\, \mathbf{u}_{\lambda,i'}, \quad \mu \in \mathcal{M}_\lambda \text{ and } i \in \mathcal{P}. \tag{42}$$

Since by assumption $h(\mathbf{x}) \geq 0,\ \forall \mathbf{x} \in V_\lambda$, then $\overline{h}_\lambda \geq 0$ and $\overline{h}_\mu \geq 0,\ \forall \mu \in M_\lambda$, as well. Thereby we conclude with (41) that

$$\frac{\overline{h}_\lambda}{\Delta h_{\min}} = \frac{\overline{h}_\lambda}{\overline{h}_\lambda - \min\limits_{\mu\in M_\lambda} \overline{h}_\mu} \geq 1.$$

Note that $\min_{\mu\in M_\lambda} \overline{h}_\mu \leq \overline{h}_\lambda$, otherwise all mean values on the sub-cells $V_\mu \subset V_\lambda$ would be larger than the mean value of the cell V_λ. By definition (36) and (40) this implies $\alpha_\lambda = 1$. Thus, the coefficients $\tilde{\mathbf{u}}_{\mu,i}$ in (35) and $\mathbf{u}_{\mu,i}$ in (34) coincide and $\mathbf{u}(\mathbf{x}) = \tilde{\mathbf{u}}(\mathbf{x})$ for all $\mathbf{x} \in V_\lambda$, respectively.

For (ii) we have to verify that $\overline{\tilde{\mathbf{u}}}_\lambda = \overline{\mathbf{u}}_\lambda$. First of all we note by (17) and (14) that

$$\int_{V_\mu} \varphi_{\mu,i}(\mathbf{x})d\mathbf{x} = \sqrt{|V_\mu|}\langle\varphi_{\mu,i}, \varphi_{\mu,0}\rangle_{V_\mu} = 0, \quad \forall\, \mu \in \mathcal{M}_\lambda \text{ and } i \in \mathcal{P}\setminus\{0\}. \tag{43}$$

Analogously, we obtain by (10) and (17)

$$\sum_{\mu\in\mathcal{M}_\lambda} \langle\varphi_{\lambda,i'}, \varphi_{\mu,0}\rangle_{V_\mu} \sqrt{|V_\mu|} = \langle\varphi_{\lambda,i'}, 1\rangle_{V_\lambda} = \sqrt{|V_\lambda|}\, \langle\varphi_{\lambda,i'}, \varphi_{\lambda,0}\rangle_{V_\lambda} = \sqrt{|V_\lambda|}\, \delta_{i'0}, \quad \forall\, i' \in \mathcal{P}. \tag{44}$$

From (43) we conclude with (38), (17) and (16) that

$$\overline{\tilde{\mathbf{u}}}_\lambda = \frac{1}{|V_\lambda|}\int_{V_\lambda} \tilde{\mathbf{u}}(\mathbf{x})d\mathbf{x} = \sum_{\mu\in\mathcal{M}_\lambda} \frac{\sqrt{|V_\mu|}}{|V_\lambda|}\, \tilde{\mathbf{u}}_{\mu,0}. \tag{45}$$

Replacing the coefficients $\tilde{\mathbf{u}}_{\mu,0}$ by (35) we obtain

$$\overline{\tilde{\mathbf{u}}}_\lambda = \sum_{\mu\in\mathcal{M}_\lambda} \frac{\sqrt{|V_\mu|}}{|V_\lambda|} \left(\langle \varphi_{\lambda,0}, \varphi_{\mu,0} \rangle_{V_\mu} \, \mathbf{u}_{\lambda,0} + \alpha_\lambda \sum_{i\in\mathcal{P}\setminus\{0\}} \langle \varphi_{\lambda,i'}, \varphi_{\mu,0} \rangle_{V_\mu} \, \mathbf{u}_{\lambda,i} \right), \tag{46}$$

where we use that $\tilde{\mathbf{u}}_{\lambda,i} = \mathbf{u}_{\lambda,i}$ for all $i \in \mathcal{P}$. According to (44), the second term in (46) vanishes. For the first term we conclude with (17), (21) and (44)

$$\overline{\tilde{\mathbf{u}}}_\lambda = \frac{\mathbf{u}_{\lambda,0}}{\sqrt{|V_\lambda|}} = \mathbf{u}_{\lambda,0}\varphi_{\lambda,0} = \overline{\mathbf{u}}_\lambda.$$

In order to show (iii) we first note that by (5) the mean value of the modified function $\tilde{h}$ on V_μ is determined by

$$\overline{\tilde{h}}_\mu := \frac{1}{|V_\mu|} \int_{V_\mu} \tilde{h}(\mathbf{x}) \, d\mathbf{x} = \varphi_{\mu,0} \langle \tilde{h}, \varphi_{\mu,0} \rangle_{V_\mu} = \tilde{h}_{\mu,0}\varphi_{\mu,0} = \frac{\tilde{h}_{\mu,0}}{\sqrt{|V_\mu|}}.$$

Inserting (35) on the right-hand side we obtain

$$\overline{\tilde{h}}_\mu = \frac{1}{\sqrt{|V_\mu|}} \left(\langle \varphi_{\lambda,0}, \varphi_{\mu,0} \rangle_{V_\mu} h_{\lambda,0} + \alpha_\lambda \sum_{i\in\mathcal{P}\setminus\{0\}} \langle \varphi_{\lambda,i}, \varphi_{\mu,0} \rangle_{V_\mu} h_{\lambda,i} \right),$$

where we use $\tilde{h}_{\lambda,i} = h_{\lambda,i}$, $i \in \mathcal{P}$. By (17) and (40) this simplifies to

$$\overline{\tilde{h}}_\mu = \overline{h}_\lambda + \alpha_\lambda \sum_{i\in\mathcal{P}\setminus\{0\}} \langle \varphi_{\lambda,i}, \varphi_{\mu,0} \rangle_{V_\mu} \frac{h_{\lambda,i}}{\sqrt{|V_\mu|}}.$$

By definition (36) of $\Delta h_{\min}$ we finally conclude

$$\min_{\mu\in\mathcal{M}_\lambda} \overline{\tilde{h}}_\mu = \overline{h}_\lambda + \alpha_\lambda \min_{\mu\in\mathcal{M}_\lambda} \sum_{i\in\mathcal{P}\setminus\{0\}} \langle \varphi_{\lambda,i}, \varphi_{\mu,0} \rangle_{V_\mu} \frac{h_{\lambda,i}}{\sqrt{|V_\mu|}} = \overline{h}_\lambda - \alpha_\lambda \Delta h_{\min}. \tag{47}$$

Then we distinguish two cases

- ($\alpha_\lambda = 1$): In this case it holds

  $$\frac{\overline{h}_\lambda}{\Delta h_{\min}} \geq 1$$

  according to (36). Then we conclude by (47)

  $$\min_{\mu\in M_\lambda} \overline{\tilde{h}}_\mu = \overline{h}_\lambda - \Delta h_{\min} \geq 0.$$

- ($\alpha_\lambda = \frac{\overline{h}_\lambda}{\Delta h_{\min}}$): In this case (47) yields

  $$\min_{\mu\in M_\lambda} \overline{\tilde{h}}_\mu = \overline{h}_\lambda - \alpha_\lambda \Delta h_{\min} = 0.$$

□

4.6. Grid Adaptation in combination with well-balancing

In order to guarantee that the adaptive scheme is well-balanced as well, we will have to add some modifications to our adaptation strategy such that coarsening and refinement will not destroy this property. For this purpose we claim that the following conditions are fulfilled:

(i) $b_h \in S_n$ for each time step t_n.

(ii) Coarsening and refinement indicators are based on $h + b$ and $h\mathbf{v}$ instead of h and $h\mathbf{v}$.

(iii) The topography b should be involved in the adaptation process.

One of the key ideas from Xing and Shu [59] to ensure well-balancing besides the flux modification (7) is to project the bottom topography b to the same space as the solution $\mathbf{u}_h$. In the adaptive scheme the grid is changing from time step to time step and thus the discretization space is changing as well. Therefore we need (i). For efficiency, we use a static grid for b (not necessarily uniform). To make sure that $b_h \in S^n$ for each time-step t_n we determine an adaptive representation of b_h by applying hard thresholding. The corresponding index set $\mathcal{D}_\varepsilon^b$ of significant detail coefficients of the bottom topography is then added in each time step to $\mathcal{D}_\varepsilon^n$ as an additional constraint to the coarsening and the refinement procedure. By this we make sure that the shape of b is involved in the adaptation process, i.e., wherever b is discontinuous or has a strong gradient the corresponding adaptive grid is refined in this area. Several other strategies to ensure well-balancing in the reference scheme rely on the continuity of b_h [18]. In this case one has only to make sure that the initial projection of b to its static adaptive grid ensures the continuity.

The coarsening step might destroy the preservation of the still water state $h + b = const$. This can be prevented by applying the multiscale transformation, prediction and thresholding on $h + b$ and by adding the constraint $\mathcal{D}_\varepsilon^b$ to the prediction and the thresholding, i.e., $\mathcal{D}_\varepsilon^n$ is replaced by $\mathcal{D}_\varepsilon^b \cup \mathcal{D}_\varepsilon^n$. However, the corrected projection (35) in the refinement is still applied on h.

4.7. Adaptive MR-DG scheme

At the beginning of the computation we have to initialize a suitable adaptive grid. Ideally one could project the initial data on the finest level and apply thresholding to obtain an adaptive grid. But this leads to the computational cost and the memory requirement of the reference scheme. We refer to [52] for a more efficient way to initialize the adaptive grids by a bottom-up strategy instead of the aforementioned top-down strategy. At $t = 0$ we apply this initialization to the initial data $\mathbf{u}_0$ and to the bottom topography b to obtain adaptive grids for both of them. Next we apply the multi-scale transformation to the grid of the bottom topography to determine $\mathcal{D}_\varepsilon^b$.

In Algorithm 3 we summarize the main steps of our scheme. Therein prediction and thresholding are performed on the multi-scale coefficients, while all the other operations are carried out on the single-scale coefficients.

Algorithm 3. *One time-step of the adaptive scheme:*

(1) ***Grid refinement (on*** $(h + b, h\mathbf{v})^t$***):***

- *(a) Apply the multi-scale transformations* (22) *and* (23) *to determine the detail coefficients,*
- *(b) perform the prediction according to* (30)*,* (31)*,* (32)*,* (33) *and the additional constraint* $\mathcal{D}_\varepsilon^b \subset \mathcal{D}^{n+1}$*,*
- *(c) determine the refined adaptive grid* $\mathcal{G}_n$ *by applying the inverse multi-scale transformation including the positivity modification* (35) *according to Algorithm 2.*
- *(d) Apply the positivity correction of high-order coefficients according to* (8)*.*

(2) ***Time evolution (on*** $(h, h\mathbf{v})^t$***):***

Perform the Runge-Kutta time evolution using an SSP-RK scheme [56] where at each stage of the RK scheme the single-scale coefficients corresponding to $(h + b, h\mathbf{v})^t$ *are limited according to Algorithm 1.*

(3) ***Grid coarsening (on*** $(h + b, h\mathbf{v})^t$***):***

- *(a) Apply the multi-scale transformations* (22) *and* (23) *to determine the detail coefficients,*
- *(b) perform thresholding of detail coefficients according to* (25) *and* (26) *where* $\mathcal{D}_\varepsilon^n$ *is replaced by* $\mathcal{D}_\varepsilon^b \cup \mathcal{D}_\varepsilon^n$*,*
- *(c) determine the adaptive grid by applying the inverse multi-scale transformation* (24)*,*
- *(d) Apply the positivity correction of the high-order coefficients according to* (8)*.*

Note that if the evolution step of the DG-scheme is well-balanced and positivity-preserving, the grid refinement and the grid coarsening will maintain these properties as well according to Proposition 2 and 3. This holds true independently of the underlying strategy to ensure these properties in the evolution step of the DG scheme, i.e., this strategy can be applied to other positivity-preserving and well-balancing strategies as well.

4.8. Choice of the threshold value

The efficiency of the adaptive scheme relies essentially on the appropriate choice of the threshold parameter ε. If it is chosen too large, the additional error introduced by the thresholding is dominating and spoils the accuracy of the solution. On the other hand, if ε is chosen too small the grid contains too many cells and efficiency is lost. For the choice of the threshold value ε we apply here the idea presented in [51] and proceed as follows:

- If the solution is expected to be non-smooth, we choose $\varepsilon = C\, h_L$.
- If the solution is expected to be smooth, we choose $\varepsilon = C\, h_L^p$.

For details we refer to [51]. The choice of C has no influence on the asymptotic behavior of the scheme. However, the efficiency and quality of the adaptive scheme is influenced by the choice of C, i.e., the size of the smallest features which are resolved by the adaptive scheme. For instance, if we have a discontinuity of strength $\varepsilon_p \ll 1$, e.g.,

$$u(\mathbf{x}, t) := \begin{cases} 1 & \text{if } x_1 < 0, \\ 1 - \varepsilon_p & \text{else} \end{cases},$$

then, this discontinuity will be resolved in the adaptive scheme provided that $\varepsilon \leq \varepsilon_p$. In this way the adaptivity acts like a filter on the data: dependent on the choice of the threshold value small perturbations might be neglected due to a local coarse grid resolution. We might be interested to refine the grid up to finest level near such a small discontinuity. Motivated by numerous computations it turned out choosing $C \sim \varepsilon_p$ triggers refinement up the finest level. In most cases $C = 1$ is a suitable choice. If we are interested in resolving very small features we have to adjust C as described above.

4.9. Limiting in the adaptive scheme

The limiter should only be applied on cells, where the solution contains discontinuities. In order to ensure that the limiter does not damage the solution in regions where the solution is smooth Cockburn and Shu [58] added the Shu constant to their limiting strategy. Provided that we have chosen a suitable threshold value the grid is refined up to the finest level near discontinuities due to the multiresolution analysis. This cannot be proven rigorously but has been observed in numerous computations. Thus, if the solution is locally smooth we expect the grid not to be refined up to the finest level in this region. Since the limiting process is local, we can easily use this information as an additional indicator for the limiting process: the limiter described in Algorithm 1 is only applied in cells on the finest level L. Therefore, we make sure that a wet/dry interface will be refined. For this purpose we perform the following changes in the adaptation process.

At each time step t_n the cells at the wet/dry interface are characterized by

$$\mathcal{G}^n_{\text{Interface}} := \{\lambda \in \mathcal{G}^n_\varepsilon : h^\lambda_{\max} > Tol \text{ and } h^\lambda_{\min} < Tol\ \}, \tag{48}$$

where the maximum value is defined by the quadrature nodes $\mathbf{x}_i \in X_\lambda$ and the mean values in the cell and its neighbours

$$h^\lambda_{\max} := \max\left(\overline{h}^n_\lambda,\ \max_{\mathbf{x}_i \in X_\lambda} h^n(\mathbf{x}_i),\ \max_{\mu : V_\mu \text{ is a neighbour of} V_\lambda} \overline{h}^n_\mu\right).$$

The minimum value is defined analogously. Following definition (48) we define

$$\mathcal{D}^n_{\text{Interface},1} := \left\{\mathbf{i} = (\lambda, i) \in \bigcup_{\ell=0}^{L-1} \mathcal{I}^W_\ell : \exists \mu \in \mathcal{G}^n_{\text{Interface}} \text{ s. t.} V_\mu \subsetneq V_\lambda\right\}$$

analogously to (29). By this we make sure that the grid near a wet/dry interface will not be coarsened. Since a new wet/dry interface might appear during the computation, we refine the corresponding cell if not yet resolved on the finest level, i.e., we inflate this index set by

$$\mathcal{D}^n_{\text{Interface},2} := \mathcal{D}^n_{\text{Interface},1} \cup \left\{(\lambda, i) : \lambda \in \mathcal{G}^n_{\text{Interface}} \setminus \mathcal{I}_L \text{ and all } i \in \mathcal{P}^*\right\}.$$

The wet/dry interface might move during one time step, therefore we add the indices of the neighbouring cells analogously to (31), i.e.,

$$\mathcal{D}^n_{\text{Interface}} := \left\{(\mu, i) : V_\mu \text{ is neighbour of } V_\lambda, (\lambda, i) \in \mathcal{D}^n_{\text{Interface},2}\right\}.$$

Finally we replace $\mathcal{D}^n_\varepsilon$ by $\mathcal{D}^n_\varepsilon \cup \mathcal{D}^n_{\text{Interface}}$ in order to incorporate the refinement triggered by the wet/dry interface into the dynamic grid adaptation. Note that if the limiter is applied to all cells in the grid this is not necessary. Since we are interested in resolving the wet/dry interface as good as possible, the additional refinement is needed anyway not alone for limiting purposes. For this reason we refine a wet/dry interface in the initial data up to the finest refinement level L.

5. Numerical results

In order to show that multiwavelet-based grid adaptation can significantly reduce the computational cost by sparsening the computational grids, while retaining accuracy and keeping well-balancing and positivity, we present numerical results for several well-known benchmark problems. First of all, we validate our method performing several one-dimensional test cases. Then we show results for two two-dimensional test cases.

5.1. One-dimensional results

Simulation parameters, unless stated otherwise were as follows. All one-dimensional cases were performed with piecewise-quadratic elements, i.e., $p = 3$. For the computational grid we used a coarsest grid cell number of $N_0 = 2$ and the number of refinement levels $L = 8$. Since in all test cases the solution is expected to be non-smooth, we have chosen $\varepsilon_L = Ch_L^p$, where the constant C was set to 0.1. For the discretization in time we have chosen the explicit third-order SSPRK scheme with CFL = 0.2. The Shu constant in the limiter was set to $M = 0$ for all cases.

5.1.1. Quiescent flow over a bump

This is a standard test case [60, 64] of quiescent flow over a continuous and smooth bed described by

$$b(x) = \max\Big(0, 0.25 - 5(x - 0.5)^2\Big). \tag{49}$$

Initial conditions for velocity are $\mathbf{v} = 0$ corresponding to a lake-at-rest condition. We consider the computational domain $\Omega = [0, 1]$. Two sets of initial conditions are included for $h + b$. The first describes a fully wet domain, where

$$h + b = 0.26. \tag{50}$$

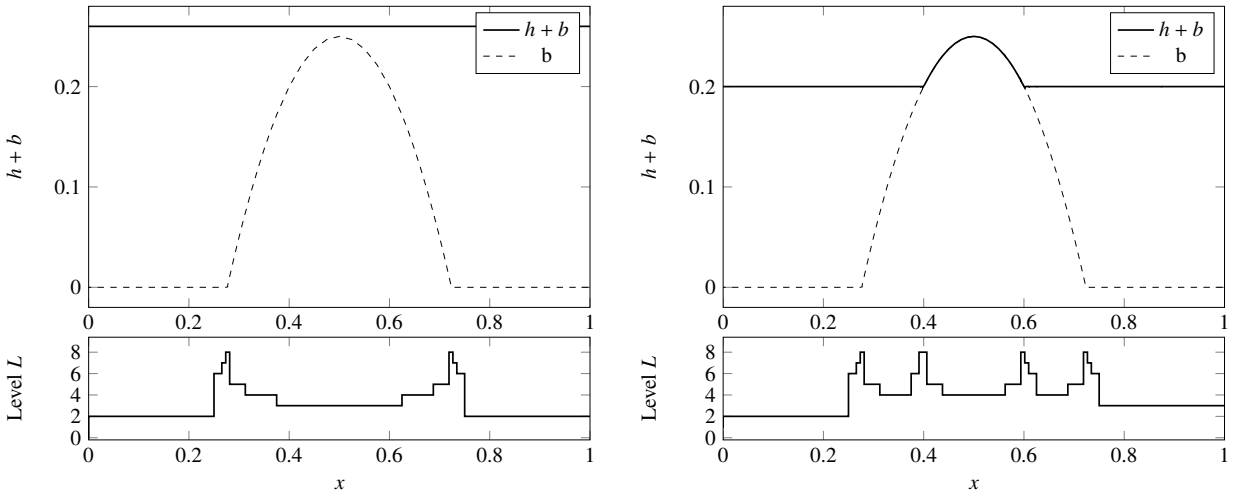
The second initial condition

$$h + b = \max\Big(0.2, b\Big) \tag{51}$$

includes a wet/dry boundary which must be preserved by the numerical scheme, while conserving positivity of the h solution.

Although it is conceptually simple, this test case allows to test several properties of the numerical strategy including well-balancing, positivity-preserving and limiting under single-resolution and multi-resolution schemes. This means that the expected outcome of the simulations is exactly the same initial state with zero velocity, and positive values of depth. Both cases were run until $t = 1\ s$.

Results for the fully wet initial condition are shown in Figure 2(a), and for the wet/dry case in Figure 2(b). For clarity in the figures, only adaptive results are shown. The adaptive solution perfectly preserves the well-balanced properties of the scheme (the uniform mesh results are identical), and zero velocity was verified throughout the domain. Furthermore, in the wet/dry case, the adaptive solution is verified to be positivity-preserving. Note that the resolution levels show how the mesh is adapted following the bed, thus producing high resolution cells at kinks and reducing resolution in the curved and constant regions. In the wet/dry case, the mesh is also adapted to the wet/dry front, which results in highest-level resolution at the front itself. Note that the adaptive solutions required, at the most, 17 and 44 cells for the fully wet and wet/dry cases, respectively, whereas the uniformly refined grid is composed of 512 cells.



(a) wet/wet case (b) wet/dry case

Figure 2. Quiescent flow: adaptive solutions for (a) the wet/wet case and (b) the wet/dry case. Top: surface $h+b$ and bottom topography b. Bottom: refinement of the grid after thresholding represented by the level of a cell.

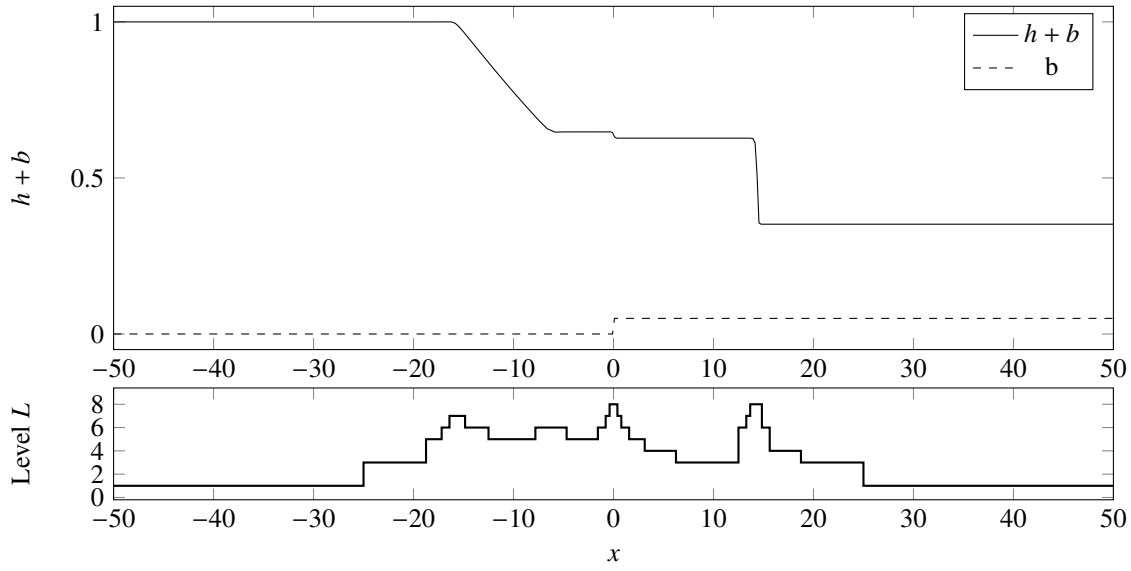


Figure 3. Dam break over a step: adaptive solution. Top: surface $h+b$ and bottom topography b. Bottom: refinement of the grid after thresholding represented by the level of a cell.

5.1.2. Dam break over a step

This test case allows to evaluate the performance of the well-balanced multiresolution strategy in the presence of shocks, rarefactions and bed discontinuities. It was presented in [7] in the context of benchmarking FV based methods and further used by [65]. It is a demanding case for well-balancing and for the slope limiter. The geometry is a frictionless rectangular channel, 100 m long, with a discontinuous bed step at $x = 0$. The bed is described by

$$b(x) = \begin{cases} 0.05 & , \text{if } x > 0 \\ 0 & , \text{if } x < 0 \end{cases} \tag{52}$$

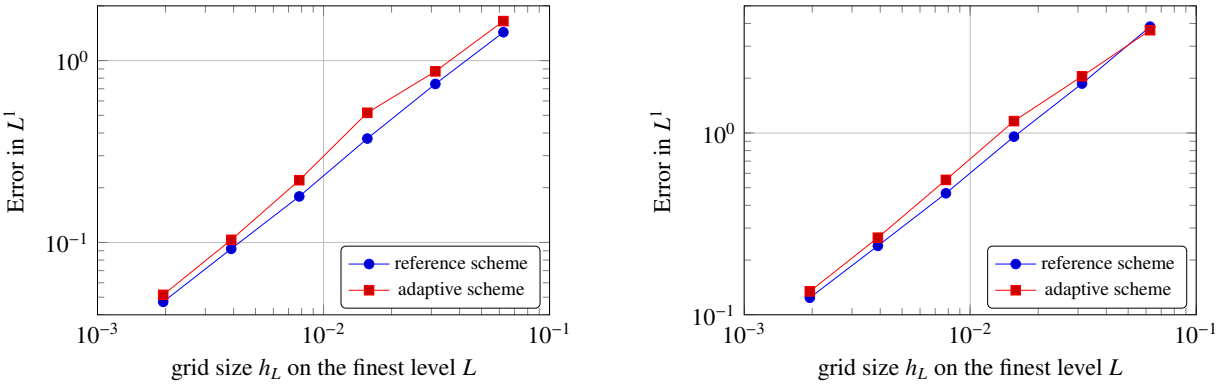
with initial conditions

$$h(x, t = 0) = \begin{cases} 1.0 & , \text{if } x > 0 \\ 0.30179953 & , \text{if } x < 0 \end{cases} \tag{53}$$

and $\mathbf{v}(x, t = 0) = 0$.

Results are shown in Figure 3 at time $T = 5\ s$ for the adaptive case with 8 refinement levels only. The adaptive scheme refines to the top level of resolution at two locations: the bed discontinuity ($x = 0$) and the water surface discontinuity ($x \approx 14$). The left moving rarefaction wave is described by mid-level resolution ($l = 4$), and with slightly higher resolution around the continuous but non-smooth regions, for example at the kink ($x \approx -16$), where $l = 5$ occurs. In the unperturbed region left of the rarefaction ($x < -25$) refinement is kept very low, at the first level of resolution. Succinctly, this is the expected outcome of the strategy.

In order to identify the quality of the adaptivity we compare the adaptive solution with a solution computed on a uniform fully refined grid. Therefore we have computed adaptive and uniform solutions for different numbers of refinement levels $3 \leq L \leq 8$. For $L = 8$ the adaptive grid has at most 53 cells, whereas the uniform grid consists of 512 cells. In order to appreciate the accuracy of the adaptive strategy, in Figure 4 we compare the L_1-error of the uniform and the adaptive scheme at $t = 5\ s$. Here we can observe that the asymptotic behaviour in accuracy of the uniform scheme is maintained by the adaptive scheme. Thereby the error of the adaptive is non-significantly larger than the error of the uniform scheme.



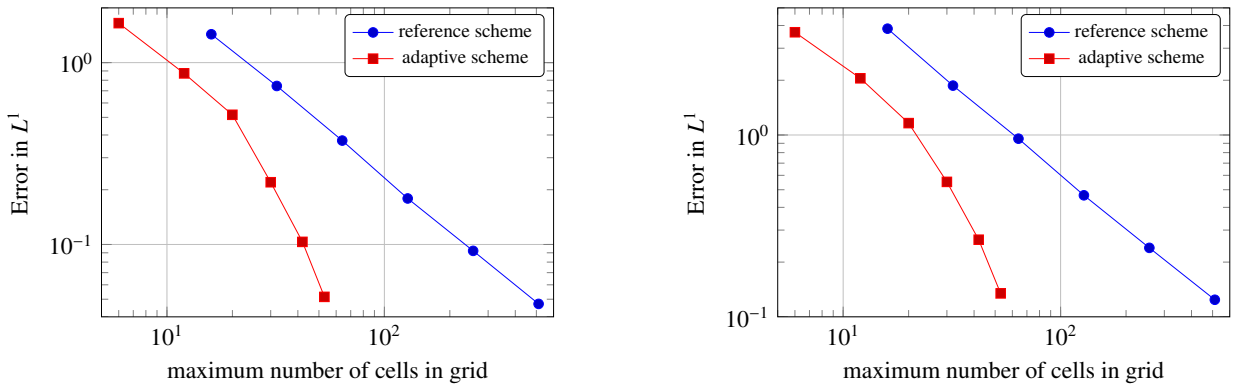
(a) height h (b) discharge hu

Figure 4. Dam break over a step: Accuracy

In order to examine the efficiency we compare the maximum number of cells in the adaptive grid for different refinement levels in Figure 5. Both figures eloquently show how the adaptive solutions are able to achieve an error in the same order as the uniform case of equivalent level but with less cells, for both conserved variables. Efficiency increases with the number of levels, achieving in the case of the adaptive $L = 8$ case an error in the order of the uniform $L = 8$, but using at the most around 10% of the cells used in the uniform $L = 8$ case.

5.1.3. Travelling wave

This test was proposed by LeVeque [66]. It consists of an initial perturbation of a quiescent state, which propagates over a smooth bed. It is well known and has been used extensively for evaluating numerical schemes [17, 20, 53, 67,



(a) height h

(b) discharge hu

Figure 5. Dam break over a step: Efficiency of the adaptive scheme.

59]. The exact solution is not known, but the qualitative behavior of the solution is well-known.

We consider the computational domain $\Omega = [0, 1]$. The bed is described by

$$b(x) = \begin{cases} \frac{1}{4}\left[\cos\left(5\pi(2x-1)\right)+1\right] & \text{, if } |x-\frac{1}{2}| < \frac{1}{10} \\ 0 & \text{otherwise} \end{cases} \tag{54}$$

with initial conditions

$$h(x, t=0) = \begin{cases} 1 - b(x) + \epsilon_p & \text{, if } \frac{1}{10} < x < \frac{1}{5} \\ 1 - b(x) & \text{otherwise} \end{cases} \quad \text{and} \quad \mathbf{v}(x, t=0) = 0. \tag{55}$$

Two cases are commonly simulated with this setup:

1. A *large* perturbation of the quiescent state with $\epsilon_p = 0.2$.
2. A *small* perturbation of the quiescent state with $\epsilon_p = 0.001$.

Following our strategy to choose the threshold value ε in Section 4.8 we have to adjust the constant C in the second case with the small perturbation. Therefore we choose $C = \epsilon_p = 0.001$ in this case. Results for the large and the small perturbation are shown in Figure 6(a) and 6(b), respectively, in terms of $h + b$, again for clarity we only present the adaptive case. Figure 6(b) shows a very narrow scale for $h + b$, in which b is out-of-scale, so that it is possible to observe the dynamics of the water surface, given that the perturbation is very small.

The uniform cases were computed with 512 cells. For $\epsilon_p = 0.2$, the maximum number of cells was 130, around 25% of the number of cells of the uniform case. The $\epsilon_p = 0.001$ case required a maximum of 152 cells, that is, around 30% of the number of cells of the uniform case. The results for both cases reproduce well the expected and well known solutions.

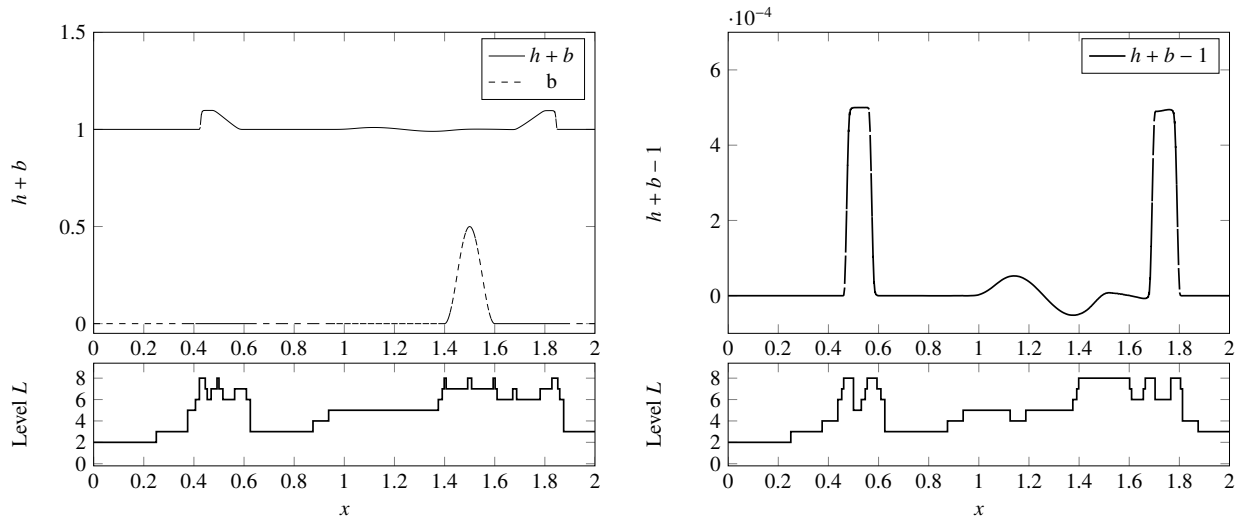
5.1.4. Dam break over a sloping bed

This test case describes a dam break over a slope in such a way that water will advance upslope. It includes no drying, therefore it is interesting to show wet front tracking capabilities of the adaptive scheme. The bed topography is described by

$$b(x) = \tan\left(\frac{\pi}{60}\right)x. \tag{56}$$

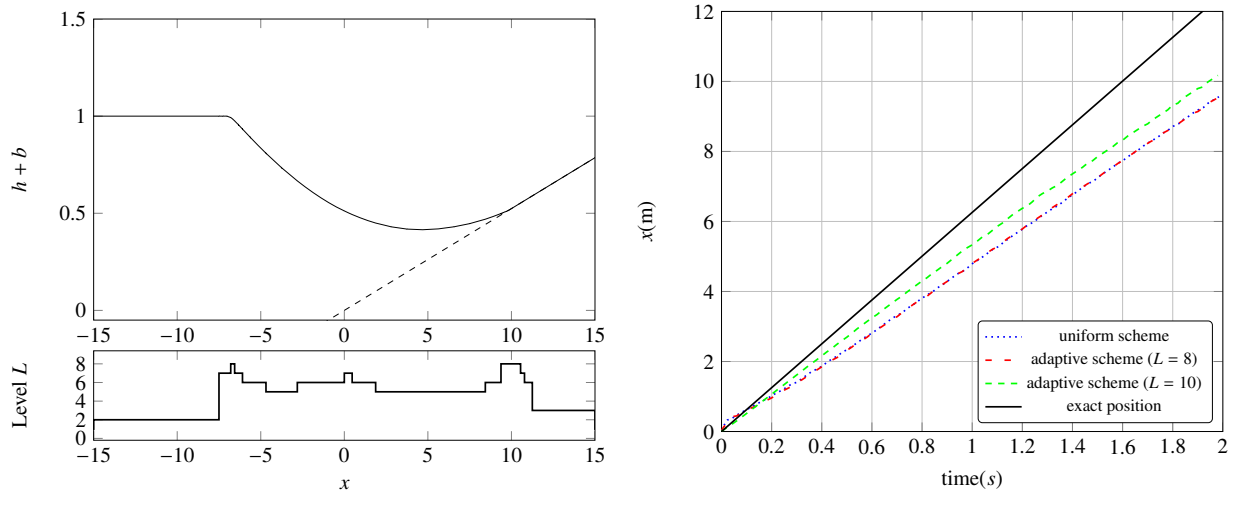
Initial conditions for the water height are $h(x, t = 0) = \max(1 - b(x), 0)$.

Figure 7(a) shows results for the test case at $T = 2\ s$. The adaptive process results in high resolution cells tracking the wetting front and the kink moving upstream. Furthermore, in the dry regions the resolution quickly drops to low levels, as it does also in the unperturbed upstream side.



(a) $\epsilon_p = 0.2$ (b) $\epsilon_p = 0.001$

Figure 6. Travelling wave with perturbation: adaptive solution. Top: surface and bottom topography. Bottom: refinement of the grid after thresholding represented by the level of a cell.



(a) adaptive solution (b) Position of the wet/dry front

Figure 7. Adaptive results and position of the wet/dry front for small dam break over a sloping bed.

Figure 7(b) shows the position of the wet/dry front in time. The exact position of the front follows [60]

$$x_f(t) = 2t\sqrt{g\cos\left(\frac{\pi}{60}\right)} - \frac{1}{2}gt^2\tan\left(\frac{\pi}{60}\right), \tag{57}$$

where g is gravitational acceleration. The numerical wet/dry front was obtained by identifying the first integration point (from left to right) that showed a depth under a threshold value of 10^{-4}. Observe how the numerical solutions approximately track the advance of the wet/dry front (with an accuracy comparable to [60]), but more importantly, how they resemble each other. This means that the adaptive process is able to track the wet/dry front with the same accuracy as the uniform, fixed mesh solution. Additionally we performed an adaptive computation with $L = 10$ refinement levels. The position of the wet/dry interface of the finer adaptive computation ($L = 10$) is closer to the exact position as expected due to grid convergence. The uniform case ($L = 8$) was computed with 512 cells, while the adaptive solution reached a maximum of 71 cells.

5.2. Two-dimensional results

In order to show that our adaptive strategy works in the two-dimensional case as well we show results for two well-known test cases. The first test case focuses on well-balancing and the second test case includes wetting and drying.

5.2.1. A Small perturbation of steady-state water

In this test case we consider a small perturbation from a steady-state water. This test case was proposed by LeVeque [68]. We consider a rectangular computational domain $\Omega = [-2, 2] \times [-1, 2]$ with the bottom topography

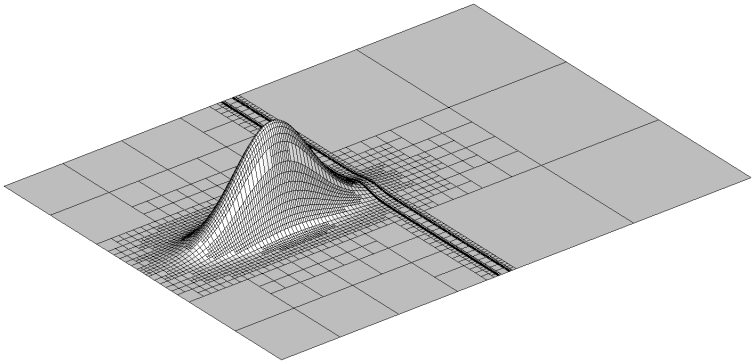


Figure 8. Small perturbation form steady-state water: Bottom shape and initial adaptive grid

$$b(\mathbf{x}) = 0.8e^{-5(\mathbf{x}_1-0.9)^2-50(\mathbf{x}_2-0.5)^2}.$$

The initial surface is given by

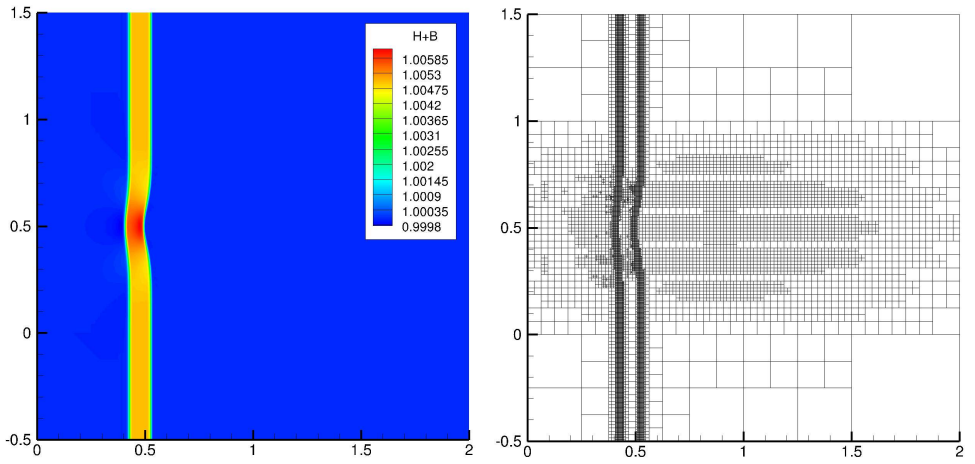
$$h(\mathbf{x}, t = 0) = \begin{cases} 1 - b(\mathbf{x}) + 0.01 & \text{if } 0.05 \leq \mathbf{x}_1 \leq 0.15 \\ 1 - b(\mathbf{x}) & \text{otherwise} \end{cases}$$

and the initial velocity is set to zero:

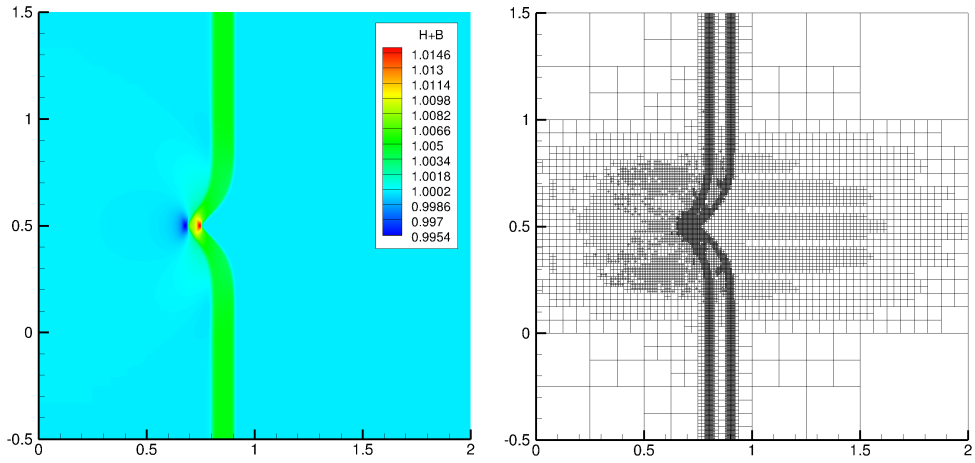
$$\mathbf{v}(\mathbf{x}, t = 0) = 0.$$

Starting from the initial perturbation two waves are propagating. One of these waves is moving towards the bump in the bottom topography and is interacting with it. A complex wave structure is generated. The difficulty of this test case is that the waves are very small in comparison to average values of the quantities.

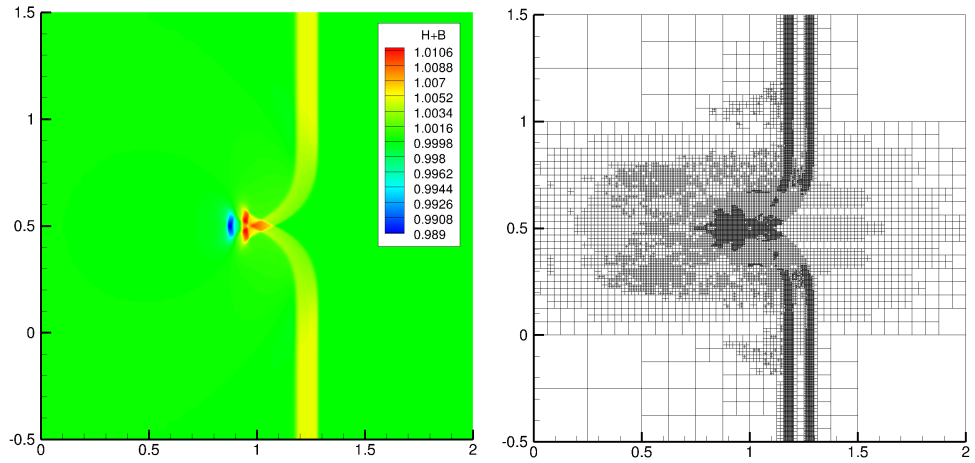
We choose 4×3 cells on the coarsest mesh $\mathcal{G}_0$ and consider $L = 8$ levels of refinement. This corresponds to a uniform grid with $786,432$ cells in the non-adaptive case. The Shu constant in the limiter is chosen as $M = 0$. Since



(a) $t = 0.12$



(b) $t = 0.24$



(c) $t = 0.36$

Figure 9. Small perturbation form steady-state water: Adaptive solution ($h + b$) and grids at different times t.

the bottom topography is smooth but includes strong gradients we choose cubic polynomials, i.e., $p = 4$, and for the time discretization we choose CFL number 0.08 and the explicit fourth-order SSP-RK(5,4) method. Since the solution contains discontinuities, we choose $\varepsilon = C\ h_L$. The constant C is chosen as 0.01 since the perturbation in the initial surface is in the same order of magnitude and we want to resolve all effects coming from this perturbation. In Figure 8 the shape of the bottom topography and the initial grid are shown. The initial adaptive grid is moderately refined in the surrounding area of the bed bump and is refined up to the finest level at the discontinuities of the initial water surface. The solution is computed up to $T = 0.6$. The maximum number of cells in the adaptive computation was about 59511 cells. This is only 7.6% of the reference mesh. In Figure 9 the adaptive solution and the corresponding grid are shown for different times, thereby we focus on the interaction of the perturbation with the bump in the bottom topography.

5.3. Two dimensional oscillating lake

We consider a parabolic bowl on the squared domain $\Omega = [-2, 2] \times [-2, 2]$ with the bottom topography $b(\mathbf{x}) = \frac{\mathbf{x}_1^2+\mathbf{x}_2^2}{10}$. The initial states are given by

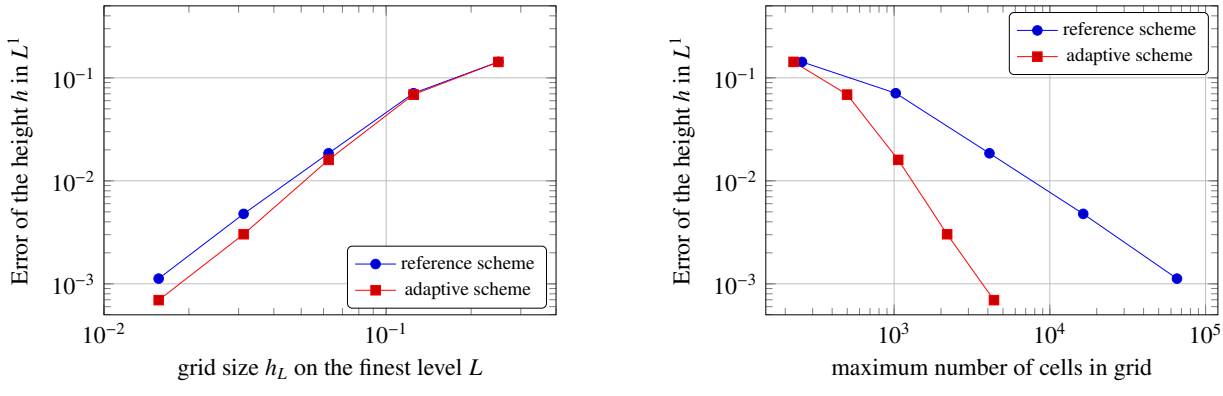
$$h(\mathbf{x}) = \max\left(0, \frac{1}{20}\left(2\mathbf{x}_1 - 0.5\right) + 0.1 - b(\mathbf{x})\right), \quad u(\mathbf{x}) = 0 \quad \text{and} \quad v(\mathbf{x}) = \frac{\omega}{2},$$

with the frequency $\omega = \sqrt{\frac{g}{5}}$. The water is circling around in the bowl and after one period the initial position is reached again. This test case was proposed by Gallardo et al [29] with the goal to test positivity-preserving and limiting of numerical schemes. The analytical solution is given by

$$h(\mathbf{x}, t) = \max\left(0, \frac{1}{20}\left(2\mathbf{x}_1 \cos(\omega t) + 2\mathbf{x}_2 \sin \omega t - 0.5\right) + 0.1 - b(\mathbf{x})\right), \quad u(\mathbf{x}, t) = -\frac{\omega}{2} \sin(\omega t) \text{ and } v(\mathbf{x}, t) = \frac{\omega}{2} \cos(\omega t).$$

Since the boundary remains dry during the computation we can choose any boundary conditions, for simplicity we have chosen reflecting boundary conditions. We choose 2×2 cells on the coarsest mesh $\mathcal{G}_0$. The Shu constant in the limiter is chosen as $M = 0.1$. We choose quadratic polynomials, i.e., $p = 3$, and for the time discretization we choose CFL number 0.01 and the explicit third-order SSPRK(3,3) method. We consider several maximum refinement levels $3 \le L \le 7$. The threshold value is chosen as $\varepsilon_L = h_L$. The tolerance in the wetting and drying is chosen dependent on the number of refinement levels L: $Tol_L = \overline{C} h_L^2$, where $\overline{C}$ is determined such that $Tol_7 = 6.25 \cdot 10^{-7}$ for $L = 7$.

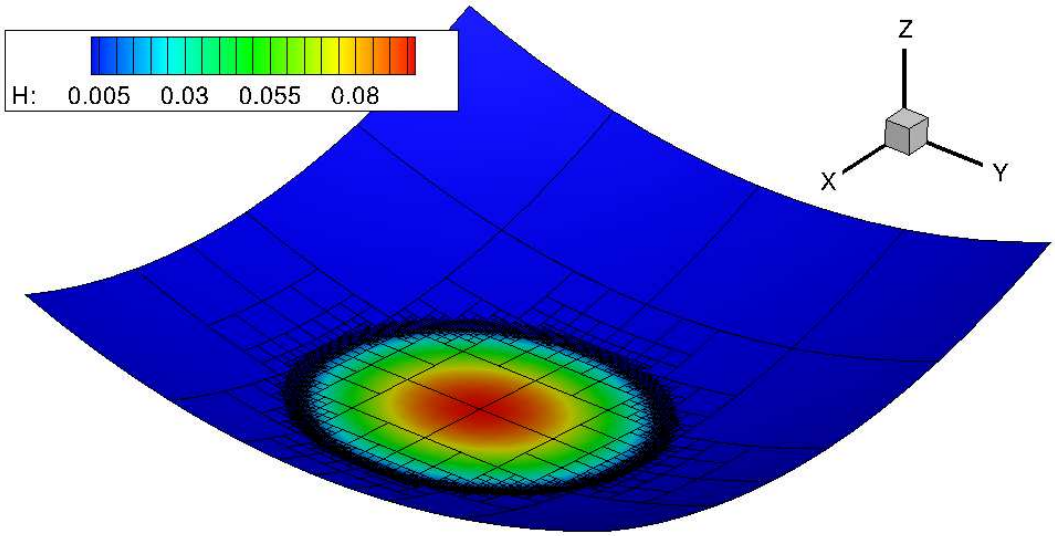
For the different levels of refinement we have computed adaptive solutions up to $T = 4.5$, such that one period is completed. Exemplarily we present the adaptive solution and the corresponding grid for $L = 7$ in Figure 11.



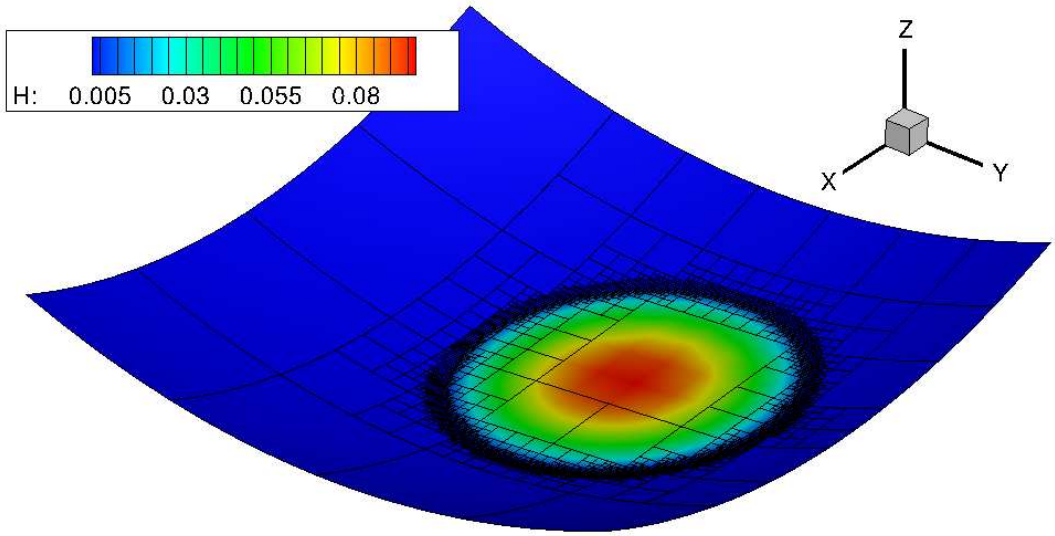
(a) Accuracy (b) Efficiency

Figure 10. Two dimensional oscillating lake: Accuracy and efficiency of the adaptive scheme.

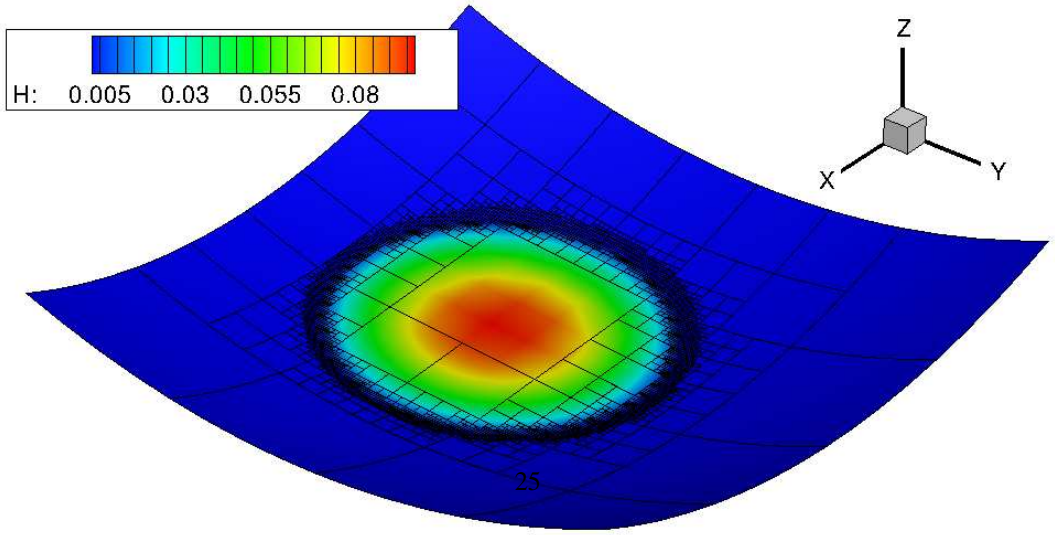
Our goal is to preserve the accuracy of the uniform scheme asymptotically with the adaptive method. In order to verify this, we compare in Figure 10(a) the error of the adaptive scheme with the error of the reference scheme



(a) $t = 0.0$



(b) $t = 1.5$



(c) $t = 3.0$

Figure 11. Oscillating lake ($L = 8$): Adaptive solution and grids and grid different times t. Contour: height h, shape: total surface $h + b$

for different refinement levels to verify that the asymptotic behaviour of the uniform scheme is maintained by the adaptive scheme. We observe that accuracy of the uniform scheme is not only maintained in the adaptive scheme, but the error of the adaptive scheme is smaller than the error of the uniform scheme. This is a surprising result. This effect might be explained by the fact that adaptivity can be understood as a filter, where small perturbations in the solution are discarded when the grid is coarse. We note that the adaptive and the uniform scheme show the same asymptotic behaviour. In Figure 10(b) we compare the maximum number of cells in the adaptive computation with the number of cells in the uniform computation. Thereby we note that the computational cost is reduced significantly by the adaptivity while the accuracy of the uniform scheme is maintained.

6. Conclusion

We have applied the multiwavelet-based grid adaptivity procedure to shallow water equations. In order to address well-balancing and positivity-preserving we have added some modifications to the adaptive strategy. For these modifications we were able to proof that positivity-preserving and well-balancing are maintained by the adaptivity. The general nature of these properties and their proof leads to the conclusion that, it is not relevant *how* these properties are achieved in the reference scheme. But if they are present in the reference scheme, the proposed adaptive strategy will preserve them across the scales and throughout the multiresolution analysis. Therefore our proposed adaptive strategy can be combined with other well-balancing and positivity-preserving strategies. Additionally we intertwined the multiwavelet-based grid adaptation with the limiting strategy. Based on a multiresolution analysis of the underlying data the limiter is only applied to cells on the finest refinement level.

The benchmark tests verify that the multiresolution DG scheme is capable of accurately solving the shallow water equations. They also show that high accuracy can be achieved with the adaptive strategy, in the same order as very fine uniform meshes, but with significantly smaller cell numbers. The adaptive strategy allows to incorporate source term effects (such as bed discontinuities) accurately as well as to accurately track discontinuities in the solution and wet/dry fronts.

The adaptive strategy relies only on one parameter: the threshold value ε. For the choice of this parameter we present a reliable strategy. Since all considered benchmark test cases are expected to have a non-smooth solution, we have chosen the threshold value as $\varepsilon = Ch_L$, with a constant C and the grid size h_L on the finest refinement level L. The choice of C was observed to be of minor importance. It was found that, although C is case-dependent, only the order of magnitude matters, i.e., it should be chosen in the same order of magnitude as the size of the features, e.g., strength of discontinuities, which are of interest. Then a proper detection of the significant details in the solution is ensured.

Acknowledgements

This work was partially funded by the DAAD (German Academic Exchange Service) through grant A1372005 and the EPSRC (UK Engineering and Physical Research Council) through grant EP/K031023/1.

References

[1] E. Toro, Shock-capturing methods for free-surface shallow flows, John Wiley, 2001.
[2] P. Brufau, P. García-Navarro, M. E. Vázquez-Cendón, Zero mass error using unsteady wetting-drying conditions in shallow flows over dry irregular topography, International Journal for Numerical Methods in Fluids 45 (10) (2004) 1047–1082.
[3] J. Burguete, P. García-Navarro, Efficient construction of high-resolution TVD conservative schemes for equations with source terms: application to shallow water flows, International Journal for Numerical Methods in Fluids 37 (2) (2001) 209–248.
[4] A. Duran, Q. Liang, F. Marche, On the well-balanced numerical discretization of shallow water equations on unstructured meshes, Journal of Computational Physics 235 (0) (2013) 565 – 586.
[5] V. Guinot, S. Soares-Frazão, Flux and source term discretization in two-dimensional shallow water models with porosity on unstructured grids, International Journal for Numerical Methods in Fluids 50 (3) (2006) 309–345.
[6] M. Morales-Hernández, M. Hubbard, P. García-Navarro, A 2D extension of a large time step explicit scheme (CFL >1) for unsteady problems with wet/dry boundaries, Journal of Computational Physics 263 (0) (2014) 303 – 327.
[7] J. Murillo, P. García-Navarro, Weak solutions for partial differential equations with source terms: Application to the shallow water equations, Journal of Computational Physics 229 (11) (2010) 4327–4368.
[8] B. F. Sanders, Integration of a shallow water model with a local time step, Journal of Hydraulic Research 46 (4) (2008) 466–475.

[9] D. Caviedes-Voullième, P. García-Navarro, J. Murillo, Influence of mesh structure on 2D full shallow water equations and SCS Curve Number simulation of rainfall/runoff events, Journal of Hydrology 448-449 (0) (2012) 39 – 59.
[10] E. Audusse, M.-O. Bristeau, A well-balanced positivity preserving second-order scheme for shallow water flows on unstructured meshes, Journal of Computational Physics 206 (1) (2005) 311 – 333.
[11] J. Murillo, P. García-Navarro, J. Burguete, R. Brufau, The influence of source terms on stability, accuracy and conservation in two-dimensional shallow flow simulation using triangular finite volumes, International Journal for Numerical Methods in Fluids 54 (5) (2007) 543–590.
[12] Y. Wang, Q. Liang, G. Kesserwani, J. W. Hall, A positivity-preserving zero-inertia model for flood simulation, Computers & Fluids 46 (1, SI) (2011) 505–511, 10th Institute for Computational Fluid Dynamics (ICFD) Conference, Univ. Reading, ENGLAND, 2010.
[13] V. Caselles, R. Donat, G. Haro, Flux-gradient and source-term balancing for certain high resolution shock-capturing schemes, Computers & Fluids 38 (1) (2009) 16 – 36.
[14] S. Noelle, Y. Xing, C.-W. Shu, High-order well-balanced finite volume WENO schemes for shallow water equation with moving water, Journal of Computational Physics 226 (1) (2007) 29 – 58.
[15] G. Vignoli, V. Titarev, E. Toro, ADER schemes for the shallow water equations in channel with irregular bottom elevation, Journal of Computational Physics 227 (4) (2008) 2463 – 2480.
[16] Q. Araud, P. Finaud-Guyot, V. Guinot, R. Mos, J. Vazquez, An eigenvector-based linear reconstruction approach for time stepping in discontinuous Galerkin scheme used to solve shallow water equations, International Journal for Numerical Methods in Fluids 70 (12) (2012) 1590–1604.
[17] G. Kesserwani, Q. Liang, Well-balanced RKDG2 solutions to the shallow water equations over irregular domains with wetting and drying, Computers & Fluids 39 (10) (2010) 2040–2050.
[18] G. Kesserwani, Q. Liang, A conservative high-order discontinuous Galerkin method for the shallow water equations with arbitrary topography, International Journal for Numerical Methods in Engineering 86 (1) (2011) 47–69.
[19] Y. Xing, C.-W. Shu, High order finite difference WENO schemes with the exact conservation property for the shallow water equations, Journal of Computational Physics 208 (1) (2005) 206 – 227.
[20] Y. Xing, C.-W. Shu, High order well-balanced finite volume WENO schemes and discontinuous Galerkin methods for a class of hyperbolic systems with source terms, Journal of Computational Physics 214 (2) (2006) 567 – 598.
[21] Y. Xing, Exactly well-balanced discontinuous Galerkin methods for the shallow water equations with moving water equilibrium, Journal of Computational Physics 257, Part A (0) (2014) 536 – 553.
[22] E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction, Springer, 2009.
[23] G. Kesserwani, R. Ghostine, J. Vazquez, A. Ghenaim, R. Mose, Riemann solvers with Runge-Kutta discontinuous Galerkin schemes for the 1D shallow water equations, Journal of Hydraulic Engineering-asce 134 (2) (2008) 243–255.
[24] J. Murillo, P. García-Navarro, Augmented versions of the HLL and HLLC Riemann solvers including source terms in one and two dimensions for shallow flow applications, Journal of Computational Physics 231 (20) (2012) 6861–6906.
[25] J. Murillo, P. García-Navarro, Wave Riemann description of friction terms in unsteady shallow flows: Application to water and mud/debris floods, Journal of Computational Physics 231 (4) (2012) 1963–2001.
[26] A. Bermudez, E. Vázquez, Upwind methods for hyperbolic conservation laws with source terms, Computer & Fluids 23 (8) (1994) 1049–1071.
[27] L. Gosse, A. Leroux, A well-balanced scheme designed for inhomogeneous scalar conservation laws, Comptes Rendus De L Academie Des Sciences Serie I-mathematique 323 (5) (1996) 543–546.
[28] J. Greenberg, A. Leroux, A well-balanced scheme for the numerical processing of source terms in hyperbolic equations, SIAM Journal on Numerical Analysis 33 (1) (1996) 1–16.
[29] J. Gallardo, C. Pares, M. Castro, On a well-balanced high-order finite volume scheme for shallow water equations with topography and dry areas, Journal of Computational Physics (227) (2007) 574–601.
[30] J.-F. Remacle, J. Flaherty, M. Shephard, An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems, SIAM Review 45 (1) (2003) 53–72.
[31] J.-F. Remacle, S. Frazao, X. Li, M. . Shephard, An adaptive discretization of shallow-water equations based on discontinuous Galerkin methods, Int. J. Numer. Meth. Fluids (52) (2006) 903–923.
[32] K. Bey, J. Oden, *hp*-version discontinuous Galerkin methods for hyperbolic conservation laws, Comput. Methods Appl. Mech. Eng. 133 (3-4) (1996) 259–286.
[33] S. Adjerid, K. Devine, J. Flaherty, L. Krivodonova, A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems, Comput. Methods Appl. Mech. Eng. 191 (2002) 1097–1112.
[34] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations, J. Comp. Phys. 183 (2002) 508–532.
[35] R. Hartmann, P. Houston, Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws, SIAM J. Sci. Comput. 24 (2002) 979–1004.
[36] P. Houston, B. Senior, E. Süli, *hp*-discontinuous Galerkin finite element methods for hyperbolic problems: Error analysis and adaptivity, Int. J. Numer. Methods Fluids 40 (1-2) (2002) 153–169.
[37] A. Dedner, C. Makridakis, M. Ohlberger, Error control for a class of Runge Kutta discontinuous Galerkin methods for nonlinear conservation laws., SIAM J. Numer. Anal. 45 (2007) 514–538.
[38] L. Wang, D. Mavriplis, Adjoint-based *hp* adaptive discontinuous Galerkin methods for the 2d compressible Euler equations, J. Comp. Phys. 228 (20) (2009) 7643–7661.
[39] A. Harten, Multiresolution algorithms for the numerical solution of hyperbolic conservation laws, Comm. Pure Appl. Math. 48 (1995) 1305–1342.
[40] B. Gottschlich-Müller, S. Müller, Adaptive finite volume schemes for conservation laws based on local multiresolution techniques., in: Hyperbolic problems: Theory, numerics, applications. Proceedings of the 7th international conference, Zürich, Switzerland, February 1998. Vol. I, Basel: Birkhäuser, 1999, pp. 385–394.

[41] S. G. Mallat, A theory for multiresolution signal decomposition: The wavelet representation, IEEE Trans. Pattern Anal. Machine Intell. 11 (7) (1989) 674–693.
[42] I. Daubechies, Ten Lectures on Wavelets, SIAM, Philadelphia, Pennsylvania, 1992.
[43] A. Harten, Discrete multi-resolution analysis and generalized wavelets, Applied Numer. Math. 12 (1993) 153–192.
[44] A. Harten, Multiresolution representation of data: A general framework, SIAM J. Numer. Anal. 33 (3) (1996) 1205–1256.
[45] A. Cohen, S. M. Kaber, S. Müller, M. Postel, Fully adaptive multiresolution finite volume schemes for conservation laws, Math. Comput. 72 (241) (2003) 183–225.
[46] S. Müller, Adaptive multiscale schemes for conservation laws, Vol. 27 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, Berlin, 2003.
[47] F. Bramkamp, P. Lamby, S. Müller, An adaptive multiscale finite volume solver for unsteady and steady state flow computations, J. Comp. Phys. 197 (2) (2004) 460–490.
[48] S. Müller, Multiresolution schemes for conservation laws., DeVore, Ronald (ed.) et al., Multiscale, nonlinear and adaptive approximation. Dedicated to Wolfgang Dahmen on the occasion of his 60th birthday. Springer, Berlin. 379-4080 (2009).
[49] R. Schäfer, Adaptive multiresolution discontinuous Galerkin schemes for conservation laws, Ph.D. thesis, RWTH Aachen University (2011).
[50] N. Hovhannisyan, S. Müller, R. Schäfer, Adaptive multiresolution discontinuous Galerkin schemes for conservation laws, Math. Comp. 83 (285) (2014) 113–151.
[51] N. Gerhard, F. Iacono, G. May, S. Müller, R. Schäfer, A high-order discontinuous Galerkin discretization with multiwavelet-based grid adaptation for compressible flows, Journal of Scientific Computing (DOI: 10.1007/s10915-014-9846-9).
[52] N. Gerhard, S. Müller, Adaptive multiresolution discontinuous Galerkin schemes for conservation laws: Multi-dimensional case, Computational and Applied Mathematics (DOI: 10.1007/s40314-014-0134-y).
[53] R. Donat, M. C. Marti, A. Martinez-Gavara, P. Mulet, Well-balanced adaptive mesh refinement for shallow water flows, Journal of Computational Physics 257, Part A (0) (2014) 937 – 953.
[54] B. Cockburn, S. Hou, C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV: The multidimensional case., Math. Comp. 54 (190) (1990) 545–581.
[55] B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework, Math. Comp. 52 (186) (1989) 411–435.
[56] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability preserving high-order time discretization methods, SIAM Review 43 (1) (2001) 89–112.
[57] B. C. S.-Y. Lin, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems, J. Comput. Phys. 84 (1989) 90–113.
[58] B. Cockburn, C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems, J. Comput. Phys. 141 (1998) 199–244.
[59] Y. Xing, C.-W. Shu, 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, Communications in Computational Physics 1 (1) (2006) 100–134.
[60] Y. Xing, X. Zhang, C.-W. Shu, Positivity-preserving high order well-balanced discontinuous Galerkin methods for the shallow water equations, Advances in Water Resources 33 (12) (2010) 1476 – 1493.
[61] R. Gandham, D. Medina, T. Warburton, GPU accelerated discontinuous Galerkin mathods for shallow water equations, arXiv:1403.1661 [math.NA].
[62] F. Keinert, Wavelets and multiwavelets, Studies in Advanced Mathematics, Chapman & Hall/CRC, Boca Raton, FL, 2004.
[63] B. Alpert, A class of bases in l^2 for the sparse representation of integral operators, SIAM J. Math. Anal. 24 (1993) 246–262.
[64] Y. Xing, C.-W. Shu, S. Noelle, On the advantage of well-balanced schemes for moving-water equilibria of the shallow water equations, Journal of Scientific Computing 48 (1-3) (2011) 339–349.
[65] M. Morales-Hernández, P. García-Navarro, J. Murillo, A large time step 1D upwind explicit scheme (CFL>1): Application to shallow water equations, Journal of Computational Physics 231 (19) (2012) 6532 – 6557.
[66] R. J. LeVeque, Balancing source terms and flux gradients in high-resolution Godunov methods: The quasi-steady wave-propagation algorithm, Journal of Computational Physics 146 (1) (1998) 346 – 365.
[67] M. Hubbard, P. Garcia-Navarro, Flux difference splitting and the balancing of source terms and flux gradients, Journal of Computational Physics 165 (1) (2000) 89–125.
[68] R. LeVeque, Balancing source terms and flux gradients in high-resolution Godunov methods: The quasi-steady wave-propagation algorithm, J. Comput. Phys. 146 (1998) 346–365.