

**Adaptive Multiresolution  
Discontinuous Galerkin Schemes for  
Conservation Laws:  
Multi-Dimensional Case**

**Nils Gerhard and Siegfried Müller**

**Bericht Nr. 375**

**September 2013**

Key words: discontinuous Galerkin methods, grid adaptivity,  
multiwavelets, multiresolution analysis, conservation laws

AMS Subject Classifications: 35L65, 65M12, 65M60, 65T60, 74S05

**Institut für Geometrie und Praktische Mathematik  
RWTH Aachen**

**Templergraben 55, D-52056 Aachen (Germany)**

# Adaptive multiresolution Discontinuous Galerkin Schemes for Conservation Laws: Multi-dimensional case

Nils Gerhard      Siegfried Müller

September 30, 2013

## Abstract

The concept of multiresolution-based adaptive DG schemes for nonlinear one-dimensional hyperbolic conservation laws has been developed and investigated analytically and numerically in *N. Hovhannisyan, S. Müller, R. Schäfer, Adaptive multiresolution Discontinuous Galerkin Schemes for Conservation Laws, Math. Comp., 2013*. The key idea is to perform a multiresolution analysis using multiwavelets on a hierarchy of nested grids for the data given on a uniformly refined mesh. 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. The focus of the present work lies on the extension of the originally one-dimensional concept to higher dimensions and the verification of the choice for the threshold value by means of parameter studies performed for linear and nonlinear scalar conservation laws.

## 1 Introduction

Nowadays discontinuous Galerkin (DG) methods are used very often in numerical simulations. The original method was introduced by Reed and Hill in 1973 [40] for a linear transport equation. In the end of the 1980s Cockburn and Shu started to develop Runge-Kutta discontinuous Galerkin (RKDG) methods for nonlinear conservation laws [14, 1, 12, 15] applying the method of lines: the partial differential equation (PDE) at hand is first discretized in space by a discontinuous Galerkin ansatz and then the resulting system of ordinary differential equations in time is solved by a Runge-Kutta (RK) method. Nowadays DG methods are widely applied not only for hyperbolic PDEs such as transport problems but also for elliptic and parabolic problems as well as ordinary differential equations. For an overview we refer to [13, 6]. A review on RKDG schemes for convection-dominated problems can be found in [16].

DG methods are very attractive because they offer the possibility to increase the order of accuracy without enlarging the discretization stencil. Thus the locality property allows for an efficient parallelization. However, DG schemes suffer from a large number of degrees of freedom per elements that significantly

inflates with increasing spatial dimension and polynomial degree of the elements. Since the solution of conservation laws typically is very heterogeneous, i.e., there exist large regions where the solution is smoothly varying and local regions exhibiting discontinuities, the underlying discretization may be coarse in smooth regions and a locally high resolution is only needed at discontinuities. Therefore it is evident to combine DG methods applied to conservation laws with grid adaptation techniques to reduce significantly the number of degrees and to realize a high accuracy of the approximation.

In order to control local grid refinement numerous refinement indicators have been developed so far that are based on interpolation error estimates of some key quantity using a priori knowledge of the solution. 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 [8], Flaherty et al. [2, 41], Houston et al. [29, 28, 30], Dedner et al. [21], and recently Mavriplis et al. [46]. These approaches are aiming at estimating the error of the solution. However, in general there are no mathematical rigorous a priori or a posteriori error estimates available for nonlinear systems of conservation laws. Therefore the *multiresolution-based grid adaptation concept* has been proposed that does *not* rely on error estimates. The rationale behind its design is to *accelerate* a given scheme (reference scheme), e.g., DG scheme or a finite volume (FV) scheme, on a uniformly refined mesh (reference mesh) through computing actually only on a locally refined adapted subgrid, while *preserving* the accuracy of the discretization on the full uniform grid. For this purpose, a multiresolution analysis (MRA) is performed, where the data corresponding to the current solution are represented as data on some coarse level and the fine scale information is encoded as arrays of detail coefficients of ascending resolution. The new data format reveals insight into the local behavior of the solution. It can be shown that the details become small with increasing refinement level when the underlying function is smooth. As suggested by this so-called *cancellation property*, we may determine a locally refined grid performing data compression on the array of detail coefficients using hard thresholding. This significantly reduces the complexity of the data. Based on the thresholded array local grid adaptation is performed, where we refine an element whenever there exists a significant detail. Of course, the crux in this context is to arrange this procedure in such a way that at no stage of the computation there is ever made use of the fully refined mesh.

This concept originates from Harten's work in the early 1990s in the framework of finite volume schemes, cf. [27]. In his work Harten was not aiming at local grid adaptation but applied the MRA only to switch between expensive and cheap flux evaluations to design a cost-effective scheme while still working on a uniformly fine grid. In the late 1990s, the MRA was also used to design fully adaptive schemes performing local grid adaptation, cf. [24, 33, 42] In the following decade multiresolution-based mesh adaptation methods have been quite successful with FV solvers. A comprehensive review together with numerous references can be found in [38] and [19] provides some overview on recent trends.

For the realization of a MRA one can use Harten's discrete framework [26] or bi-orthogonal wavelets [18], respectively. The efficiency of multiresolution-based mesh adaptation schemes crucially relies on the compression rate. In the wavelet context it can be proven that the rate of decay of the details fastens

with increasing number of vanishing moments of the wavelets, i.e., more details may be discarded in smooth regions and the adaptive mesh becomes coarser. However, for biorthogonal wavelets to realize more vanishing moments requires to extend the support of the wavelet functions. Its construction, in particular, becomes even more complicate on unstructured grid hierarchies. Therefore it is natural to extend the multiresolution-based mesh adaptation concept to higher order DG discretizations using so-called multiwavelets [45, 34]. Multiwavelets allow for higher order vanishing moments, while being supported on a single mesh element. First work in this regard has been reported in [10, 44, 5, 31, 43]:

Calle et al. [10] consider the implicit diffusive DG scheme (IDDG) where the DG discretization is stabilized using streamline diffusion instead of a limiter. The IDDG scheme is combined with wavelet-based grid adaptation techniques where instead of multiwavelets the Haar wavelet is used in the multiscale analysis providing only *one* vanishing moment. Therefore compression rates are moderate and the adaptive grids are not as sparse as for higher order vanishing moments.

In [44] Shelton presents a multiresolution DG method for unsteady compressible flows, adopting Alpert's multiwavelets in the multiresolution analysis. There scale transition algorithms are used to transfer matrix operators of the reference scheme onto the multiresolution-based DG scheme. Such a transfer is not needed in our method, where the underlying scheme is a classical DG method, which acts on the single-scale coefficients and not on the details.

In [31, 43] an implementation for scalar one-dimensional conservation laws has been developed and tested. In particular, it has been rigorously proved that choosing an ideal threshold value the adaptive solution is of the same accuracy as the reference solution on a uniformly refined mesh, but at significantly reduced computational cost. A proof of concept for a solution methodology for compressible flows has been given in [32].

Main objective of this work is the extension of the method developed in [31, 43] to the multi-dimensional case. For this purpose we construct appropriate multiwavelets. Since there is no analysis available similar to the one-dimensional scalar case that provides us with the choice of an appropriate threshold value, we perform parameter studies to verify numerically that the one-dimensional result may extend to the multi-dimensional case.

The paper is organized as follows. In Section 2 we briefly summarize the key ingredients of RKDG schemes. In Section 3 we introduce the concept of MRA using multiwavelets and their construction on general multidimensional nested grid hierarchies and explain how to design a locally refined grid. Then in Section 4 we present the multiwavelet-based adaptive DG scheme and present a strategy to choose an appropriate threshold value. Finally, we consider in Section 5 some numerical examples where we apply the adaptive scheme to a linear transport equation, the inviscid Burgers' equation and two non-linear test cases with non-convex fluxes.

## 2 The reference DG scheme

For convenience of the reader and to fix the notation we briefly summarize the main ideas of RKDG schemes applied to multi-dimensional conservation laws

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f}(u) = 0 \quad (2.1)$$

in the space-time domain  $\Omega \times (0, T)$  with  $\Omega \subset \mathbb{R}^d$  and  $T > 0$  describing the evolution of the conserved quantity  $u : \Omega \times [0, T] \rightarrow \mathbb{R}$  with the flux vector  $\mathbf{f} : \mathbb{R} \rightarrow \mathbb{R}^d$ . This problem has to be supplemented with initial data

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega$$

and suitable boundary conditions, e.g., periodic conditions or Dirichlet conditions at the inflow part. In particular, choosing  $\Omega = \mathbb{R}^d$ , then there exists a unique entropy solution  $u \in L^\infty(0, T, L^1_{loc}(\mathbb{R}^d))$  for all  $T > 0$  provided that  $u_0 \in BV(\mathbb{R}^d) \cap L^\infty(\mathbb{R}^d) \cap L^1(\mathbb{R}^d)$  and  $f \in C^1(\mathbb{R}^d)$ , see [22].

### 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 scheme. Here we confine ourselves to the semi-discrete formulation of the DG method. A detailed description can be found in [12] and [14]:

Since computations are only performed on bounded domains, we consider in the following a uniform finite discretization. For this purpose we discretize the domain  $\Omega$  by a finite number of cells  $V_\lambda$  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

$$S := \{f \in L^2(\Omega) : f|_{V_\lambda} \in \Pi_{p-1}(V_\lambda) \forall \lambda \in \mathcal{I}\}. \quad (2.2)$$

where  $\Pi_{p-1}(V)$  denotes the space of polynomials on the element  $V$  of total degree less than  $p$ , i.e.,

$$\Pi_{p-1}(V) := \text{span} \{ \mathbf{x}^i|_V : i \in \mathcal{P} \} \quad \text{with} \quad \mathcal{P} := \{ i \in \mathbb{N}_0^d : \|i\|_1 \leq p-1 \}.$$

For the finite-dimensional space  $S$  we introduce two sets of basis functions,  $\Phi := \{\varphi_{\lambda,i}\}_{\lambda \in \mathcal{I}, i \in \mathcal{P}}$  and  $\tilde{\Phi} := \{\tilde{\varphi}_{\lambda,i}\}_{\lambda \in \mathcal{I}, i \in \mathcal{P}}$  such that  $S^p = \text{span} \Phi = \text{span} \tilde{\Phi}$ . We assume that these functions fulfill the biorthogonality relation

$$\langle \varphi_{\lambda',i'}, \tilde{\varphi}_{\lambda,i} \rangle_\Omega = \delta_{i,i'} \delta_{\lambda,\lambda'} \quad (2.3)$$

and are compactly supported, i.e.,  $\text{supp} \varphi_{\lambda,i} = \text{supp} \tilde{\varphi}_{\lambda,i} = V_\lambda$ ,  $i \in \mathcal{P}$ . Here  $\langle f, g \rangle_\Omega := \int_\Omega f(\mathbf{x}) g(\mathbf{x}) d\mathbf{x}$  denotes the standard  $L^2$ -inner product.

For the semi-discretization of (2.1), we now assume that the approximate solution can be written as an expansion of the basis  $\Phi$ , i.e.,

$$u_h(\cdot, t) = \sum_{\lambda \in \mathcal{I}} \sum_{i \in \mathcal{P}} v_{\lambda,i}(t) \varphi_{\lambda,i}(\cdot) \in S. \quad (2.4)$$

For ease of notation we distinguish the coefficients and the functions by  $v$  and  $u$ , respectively. The coefficients  $v_{\lambda,i}$  are determined by the biorthogonality relation (2.3) as

$$v_{\lambda,i}(t) = \langle u_h(\cdot, t), \tilde{\varphi}_{\lambda,i} \rangle_{\Omega}. \quad (2.5)$$

In order to derive an evolution equation for these coefficients we first rewrite (2.1) in a weak formulation. For this purpose we multiply (2.1) by a test function  $v_h \in S$  and integrate over a cell  $V_\lambda$ . Then we perform integration by parts and introduce numerical fluxes  $\hat{\mathbf{f}}$  approximating the fluxes  $\mathbf{f} \cdot \vec{n}$  in direction  $\vec{n}$ . This results in the semi-discrete DG-formulation:

Find  $u_h(\cdot, t) \in S, t \in [0, T]$ , such that for all  $v_h \in S$  and  $\lambda \in \mathcal{I}$ :

$$\int_{V_\lambda} \frac{\partial u_h}{\partial t} v_h d\mathbf{x} - \int_{V_\lambda} \mathbf{f}(u_h) \cdot \nabla v_h d\mathbf{x} + \int_{\partial V_\lambda} \hat{\mathbf{f}}(u_h^+, u_h^-, \vec{n}_\lambda) v_h dS = 0. \quad (2.6)$$

Here  $u_h^+$  denotes the inner and  $u_h^-$  the outer value of  $u_h$  at the boundary of  $V_\lambda$ . Furthermore  $\vec{n}_\lambda$  is the outward pointing unit normal vector corresponding to  $V_\lambda$ . In our computations we choose the local Lax-Friedrichs flux [14]:

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

with

$$\alpha_\lambda = s \max_{u \in I(u^+, u^-)} |\mathbf{f}'(a) \cdot \vec{n}|,$$

where  $I(u^+, u^-) := [\min(u^+, u^-), \max(u^+, u^-)]$  and  $s \in \{-1, 1\}$  depends on the orientation of  $\vec{n}$ . If  $f$  is convex, this reduces to  $\alpha_\lambda = s \max(|\mathbf{f}'(u^-) \cdot \vec{n}|, |\mathbf{f}'(u^+) \cdot \vec{n}|)$ . Other numerical fluxes, known from the context of finite volume schemes could be chosen as well provided the flux is consistent and acts stabilizing to the scheme.

If we choose in (2.6) for the test functions the basis functions collected in  $\tilde{\Phi}$ , i.e.,  $v_h = \tilde{\varphi}_{\lambda,i}$ , then we obtain a system of ordinary differential equations in time for the coefficients in the basis representation (2.4) of the solution  $u_h$

$$\frac{\partial v_{\lambda,i}}{\partial t} = \int_{V_\lambda} \mathbf{f}(u_h) \cdot \nabla \tilde{\varphi}_{\lambda,i} d\mathbf{x} - \int_{\partial V_\lambda} \hat{\mathbf{f}}(u_h^+, u_h^-, \vec{n}_\lambda) \tilde{\varphi}_{\lambda,i} dS =: G_{\lambda,i} - B_{\lambda,i}, \quad (2.7)$$

where the right-hand side is composed of a volume integral  $G_{\lambda,i}$  and a boundary integral  $B_{\lambda,i}$ , respectively. Here we employ the biorthogonality property (2.3) and the local support of the basis functions.

To obtain a fully-discrete method, we have to discretize the system in time. For this purpose we use strong-stability-preserving Runge-Kutta methods [23] in our computations in Section 5.

## 2.2 Local projection limiting

In order to suppress oscillations near discontinuities which typically arise in hyperbolic equations we have to stabilize the discretization by either adding artificial viscosity or applying a limiter to guarantee nonlinear stability of the DG scheme. Here we use the Barth-Jespersen limiter [7] that was originally developed in the context of finite volume schemes on unstructured meshes. Here we apply their idea to the  $\Pi_1$ -projection of  $u_h$  for a fixed time instance  $t_n \in [0, T]$ :

- (i) Project  $u_h$  to  $\Pi_1$ , then the  $\Pi_1$ -part (linear part) of  $u_h$  can be written locally as:

$$\begin{aligned} u_{\Pi_1}(\mathbf{x}, t^n) &:= \Pi_1 u_h(\mathbf{x}, t^n) \\ &= \bar{u}_\lambda(t^n) + \nabla u_{\Pi_1}(t^n) (\mathbf{x} - \bar{\mathbf{x}}_\lambda), \quad \mathbf{x} \in V_\lambda, \end{aligned}$$

where  $\bar{u}_\lambda$  is the mean value of  $u_h$  in  $V_\lambda$  and  $\bar{\mathbf{x}}_\lambda$  the centroid of  $V_\lambda$ .

- (ii) Compute the maximum and minimum mean value  $\bar{u}_{min}$  and  $\bar{u}_{max}$  over all elements  $V_\mu$  which share an edge with  $V_\lambda$  as well as  $V_\lambda$ .
- (iii) For each cell  $V_\lambda$  find the largest  $\alpha \in [0, 1]$  such that for some fixed control points  $\mathbf{x}_i \in \bar{V}_\lambda$ :

$$\bar{u}_{min} \leq \tilde{u}_{\Pi_1}(\mathbf{x}_i, t^n) \leq \bar{u}_{max}$$

holds, where

$$\tilde{u}_{\Pi_1}(\mathbf{x}, t^n) := \bar{u}_\lambda(t^n) + \alpha \nabla u_{\Pi_1}(t^n) (\mathbf{x} - \mathbf{x}_\lambda), \quad \mathbf{x} \in V_\lambda.$$

- (iv) If  $\tilde{u}_{\Pi_1} \neq u_{\Pi_1}$  the solution will be limited on  $V_\lambda$ , i.e., set all high order coefficients corresponding to the cell  $V_\lambda$  in  $(v_{\lambda,i})_{i \in \mathcal{P}}$  to zero and scale the linear coefficients according to  $\alpha$ .

The basic idea of this limiter is the assumption that spurious oscillations only occur in the approximate solution if they already occur in its  $\Pi_1$ -part. If limiting is necessary, the solution will be projected to its  $\Pi_1$ -part. Then the slopes will be limited and all higher order coefficients will be set to zero. Thus the approximate solution is corrected in the entire cell. Note that the limiter is formulated independently of the polynomial degree  $p$ .

In our computations the control points  $\mathbf{x}_i$  are chosen as the nodes of the Gaussian quadrature rule, which is used to approximate the integral on the boundary of  $V_\lambda$  in (2.6). The local projection limiting is applied to each stage of the Runge-Kutta method used for the time integration of (2.7).

### 3 Multiresolution analysis, data compression and grid adaptation

The DG discretization typically works on an array of coefficients, see (2.5) and (2.7). 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. [34]. This can be considered a natural extension of the MRA for cell averages corresponding to a FV discretization, i.e.,  $p = 1$ , where biorthogonal wavelets have been used to construct an appropriate MRA, cf. [20] and [37]. Here we will briefly summarize the basic ideas of the MRA concept. The core ingredients are (i) a hierarchy of nested grids, (ii) (orthogonal) multiwavelets and (iii) the multi-scale decomposition. In contrast to [31] we present here the MRA on general *multi-dimensional* grid hierarchies.

### 3.1 Nested grid hierarchy

The starting point for the multiresolution analysis is a coarse uniform 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  $\ell$ :

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

Obviously, the resulting grid hierarchy is nested:

$$V_\lambda = \bigcup_{\mu \in \mathcal{M}_\lambda} V_\mu, \quad \forall \lambda \in \mathcal{I}_\ell, \ell \in \mathbb{N}_0, \quad (3.1)$$

where  $\mathcal{M}_\lambda \subset \mathcal{I}_{\ell+1}$  denotes the *refinement set* of cell  $\lambda$ . In Figure 1 we exemplarily present two-dimensional grid hierarchies for triangular and quadrilateral elements.

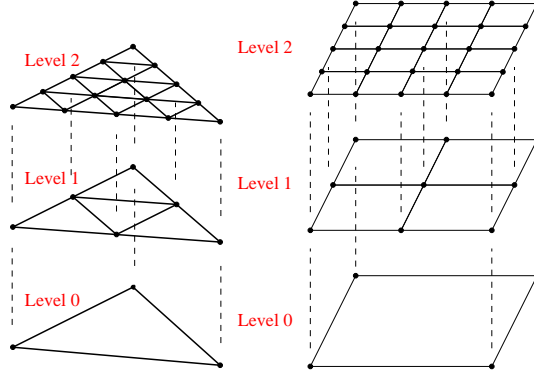


Figure 1: Two-dimensional grid hierarchies for triangular and quadrilateral grids with  $|\mathcal{M}_\lambda| = 4$ .

### 3.2 Multiresolution analysis

Similar to (2.2) 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* [36]. 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.,

$$S_0 \subset S_1 \subset \dots \subset S_{\ell+1} \subset \dots \subset L^2(\Omega). \quad (3.2)$$

such that  $\mathcal{S}$  is dense in  $L^2(\Omega)$ , i.e.,

$$\text{clos}_{L^2(\Omega)} \left( \bigcup_{\ell=0}^{\infty} S_\ell \right) = L^2(\Omega). \quad (3.3)$$



The nestedness of the discretization spaces implies that there exist orthogonal complement spaces  $W_\ell := S_{\ell+1} \setminus S_\ell$  such that  $S_{\ell+1} = S_\ell \oplus W_\ell$ . Recursively applying this two-scale relation we obtain:

$$S_L = S_0 \oplus W_0 \oplus \dots \oplus W_{L-1} \quad (3.4)$$

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 behaviour of the solution that can be exploited to trigger the grid refinement process.

### 3.3 Choice of basis

To perform the multi-scale decomposition of the single-scale space  $S_L$  in (3.4) we need to specify a bases for the single-scale spaces  $S_\ell$  and the complement spaces  $W_\ell$ . 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:

$$\begin{aligned} S_\ell &= \text{span}_{\lambda \in \mathcal{I}_\ell, i \in \mathcal{P}} \varphi_{\lambda, i}, & \text{supp}(\varphi_{\lambda, i}) &= V_\lambda, \\ W_\ell &= \text{span}_{\lambda \in \mathcal{I}_\ell, i \in \mathcal{P}, e \in \mathcal{E}} \psi_{\lambda, i, e}, & \text{supp}(\psi_{\lambda, i, e}) &= V_\lambda, \end{aligned}$$

where  $\mathcal{P}$  characterizes the local degrees of freedom, see Section 2.1, and  $\mathcal{E} := \{1, \dots, \#\mathcal{M}_\lambda - 1\}$  the additional degrees of freedom of the complement spaces  $W_\ell$ . The functions  $\varphi_{\lambda, i}$  and  $\psi_{\lambda, i}$  are referred to as *scaling functions* and *multi-wavelets*, respectively. For convenience of the reader we simplify the presentation and introduce the set  $\mathcal{P}^* := \mathcal{P} \times \mathcal{E}$  characterizing the total local degrees of freedom of the complement spaces. Furthermore, we define the 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  $W_\ell$  and  $S_\ell$ , respectively. Then we may rewrite these spaces as

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

In addition to the compact support, we want the basis to be orthogonal:

$$\langle \varphi_{\mathbf{i}}, \varphi_{\mathbf{j}} \rangle_\Omega = 0, \quad \mathbf{i}, \mathbf{j} \in \mathcal{I}_\ell^S, \quad \mathbf{i} \neq \mathbf{j}, \quad (3.5)$$

$$\langle \psi_{\mathbf{i}}, \psi_{\mathbf{j}} \rangle_\Omega = 0, \quad \mathbf{i}, \mathbf{j} \in \mathcal{I}_\ell^W, \quad \mathbf{i} \neq \mathbf{j}. \quad (3.6)$$

By definition of the complement spaces  $W^\ell$  the wavelets are orthogonal to the scaling functions:

$$\langle \psi_{\mathbf{i}}, \varphi_{\mathbf{j}} \rangle_\Omega = 0 \quad \mathbf{i} \in \mathcal{I}_\ell^W, \quad \mathbf{j} \in \mathcal{I}_\ell^S, \quad \ell \in \mathbb{N}_0. \quad (3.7)$$

The scaling of the basis functions is chosen such that:

$$\|\varphi_{\mathbf{i}}\|_{L^\infty(\Omega)} \leq C \quad \text{and} \quad \|\psi_{\mathbf{j}}\|_{L^\infty(\Omega)} \leq C \quad \mathbf{i} \in \mathcal{I}_\ell^S, \mathbf{j} \in \mathcal{I}_\ell^W, \quad (3.8)$$

where the constant  $C$  is independent of  $\ell$ . Due to the uniform boundedness in  $L^\infty$  the basis is not orthonormal. Therefore we introduce bi-orthogonal bases

$\{\tilde{\varphi}_{\mathbf{i}}\}_{\mathbf{i} \in \mathcal{I}_\ell^S}$  for  $S_\ell$  and  $\{\tilde{\psi}_{\mathbf{i}}\}_{\mathbf{i} \in \mathcal{I}_\ell^W}$  for  $W_\ell$  such that:

$$\begin{aligned} \langle \varphi_{\mathbf{i}}, \tilde{\varphi}_{\mathbf{j}} \rangle_\Omega &= \delta_{\mathbf{ij}}, \quad \mathbf{i}, \mathbf{j} \in \mathcal{I}_\ell^S, & \langle \psi_{\mathbf{i}}, \tilde{\psi}_{\mathbf{j}} \rangle_\Omega &= \delta_{\mathbf{ij}}, \quad \mathbf{i}, \mathbf{j} \in \mathcal{I}_\ell^W, \\ \langle \varphi_{\mathbf{i}}, \tilde{\psi}_{\mathbf{j}} \rangle_\Omega &= 0, \quad \mathbf{i} \in \mathcal{I}_\ell^S, \mathbf{j} \in \mathcal{I}_\ell^W, & \langle \psi_{\mathbf{i}}, \tilde{\varphi}_{\mathbf{j}} \rangle_\Omega &= 0, \quad \mathbf{i} \in \mathcal{I}_\ell^W, \mathbf{j} \in \mathcal{I}_\ell^S. \end{aligned} \quad (3.9)$$

These dual basis functions can easily be constructed by scaling the primal basis functions such that they are uniformly bounded in  $L^1(\Omega)$ .

Finally we need that each multiwavelet provides vanishing moments of order  $M \geq p$ :

$$\langle P, \psi_{\mathbf{i}} \rangle_\Omega = 0, \quad \forall P \in \Pi_{M-1}. \quad (3.10)$$

Since the scaling functions are assumed to be piecewise polynomials, the orthogonality relation (3.7) implies that the minimal number of vanishing moments is larger or equal to  $p$ , i.e.,  $M \geq p$ . From the nestedness of the spaces and required properties of the basis functions we can derive two-scale relations for the scaling functions and multiwavelets:

**Proposition 1** *For the scaling functions and multiwavelets the two-scale relations hold:*

$$\begin{aligned} (i) \quad \tilde{\varphi}_{\lambda, \mathbf{i}} &= \sum_{\mu \in \mathcal{M}_\lambda} \sum_{\mathbf{i}' \in \mathcal{P}} \tilde{\varphi}_{\mu, \mathbf{i}'} \langle \varphi_{\mu, \mathbf{i}'}, \tilde{\varphi}_{\lambda, \mathbf{i}} \rangle_{V_\mu}, \quad \lambda \in \mathcal{I}_\ell, \mathbf{i} \in \mathcal{P}, \ell = 0, \dots, L-1, \\ (ii) \quad \tilde{\psi}_{\lambda, \mathbf{i}} &= \sum_{\mu \in \mathcal{M}_\lambda} \sum_{\mathbf{i}' \in \mathcal{P}} \tilde{\varphi}_{\mu, \mathbf{i}'} \langle \varphi_{\mu, \mathbf{i}'}, \tilde{\psi}_{\lambda, \mathbf{i}} \rangle_{V_\mu}, \quad \lambda \in \mathcal{I}_\ell, \mathbf{i} \in \mathcal{P}^*, \ell = 0, \dots, L-1, \\ (iii) \quad \tilde{\varphi}_{\mu, \mathbf{i}} &= \sum_{\mathbf{i}' \in \mathcal{P}} \tilde{\varphi}_{\lambda, \mathbf{i}'} \langle \varphi_{\lambda, \mathbf{i}'}, \tilde{\varphi}_{\mu, \mathbf{i}} \rangle_{V_\mu} + \sum_{\mathbf{i}' \in \mathcal{P}^*} \tilde{\psi}_{\lambda, \mathbf{i}'} \langle \psi_{\lambda, \mathbf{i}'}, \tilde{\varphi}_{\mu, \mathbf{i}} \rangle_{V_\mu}, \quad \mu \in \mathcal{M}_\lambda, \lambda \in \mathcal{I}_\ell, \mathbf{i} \in \mathcal{P}, \ell = 1, \dots, L. \end{aligned}$$

### 3.4 Construction of multiwavelets and scaling functions

For a dyadic grid hierarchy on a one-dimensional interval, i.e.,  $\Omega = [a, b]$ , a prominent example are the Legendre polynomials and Alpert's *multiwavelets* [3, 4] for the scaling functions and the multiwavelets, respectively. Then  $\varphi_{\lambda, i}$  and  $\psi_{\lambda, i}$  are determined by shifts and translates, see [31]. Whereas wavelets are in general continuous functions, this regularity is lost when passing to Alpert's multiwavelets. This loss of regularity is necessary, since the complement spaces contain discontinuous functions.

In the case of a Cartesian grid and a tensor-product based polynomial space, which is characterized by  $\overline{\mathcal{P}} := \{i \in \mathbb{N}_0^d : \|i\|_\infty \leq p-1\}$  instead of  $\mathcal{P}$ , one can use the tensor products of the one-dimensional basis functions in the different directions. However, the tensor-product ansatz inflates the degrees of freedom because  $\#\mathcal{P} < \#\overline{\mathcal{P}}$ . Therefore we construct genuinely multi-dimensional scaling functions and multiwavelets. The scaling functions  $\varphi_{\lambda, i}$ ,  $i \in \mathcal{P}$  can be constructed by applying the Gram-Schmidt process to a monomial basis on  $V_\lambda$  and a following normalization with respect to  $L^\infty(V_\lambda)$ . For the construction of the multiwavelets we generalize Alpert's construction principle [3] for arbitrary nested grid hierarchies:

**Algorithm 1 (Construction of multiwavelets on cell  $V_\lambda$ )**

*For given  $\varphi_{\lambda, j}$ ,  $j \in \mathcal{P}$ , and  $M_\lambda = \{\mu_1, \dots, \mu_{|\mathcal{M}_\lambda|}\}$  the multiwavelets  $\psi_{\lambda, i}$ ,  $i \in \mathcal{P}^*$ , can be constructed by the following steps:*

(i) Initialize  $\psi_{\lambda,i}$ ,  $i \in \mathcal{P}^*$ , such that the following conditions are satisfied:

- a)  $\psi_{\lambda,i}|_{V_\mu} \in \Pi_{p-1}(V_\mu)$ ,  $\mu \in M_\lambda$ ,
- b)  $\text{span}_{i \in \mathcal{P}^*} \psi_{\lambda,i}|_{V_\lambda} \notin \Pi_{p-1}(V_\lambda)$ ,
- c)  $\{\psi_{\lambda,i}\}_{i \in \mathcal{P}^*}$  is a linearly independent family,
- d)  $\text{supp } \psi_{\lambda,i} = V_\lambda$ .

(ii) Apply the Gram-Schmidt process to orthogonalize  $\psi_{\lambda,i}$  with respect to  $\varphi_{\lambda,j}$ ,  $j \in \mathcal{P}$ .

(iii) Orthogonalize  $\psi_{\lambda,i}$ ,  $i \in \mathcal{P}^*$  by a Gram-Schmidt process.

(iv) Normalize  $\psi_{\lambda,i,e}$ ,  $i \in \mathcal{P}$ ,  $e \in \mathcal{E}$  with respect to  $L^\infty(V_\lambda)$ .

Note that by construction the wavelets are (i) locally supported, (ii) orthogonal and (iii) have vanishing moments of order  $M = p$ . In step (i) one might choose:

$$\psi_{\lambda,i,e} := \begin{cases} \mathbf{x}^i & \text{if } \mathbf{x} \in V_{\mu_e} \\ -\mathbf{x}^i & \text{if } \mathbf{x} \in V_\lambda \setminus V_{\mu_e} \\ 0 & \text{else} \end{cases}$$

For quadrilateral or triangular grids with a uniform refinement, cf. Figure 1, the multiwavelets can be constructed on a reference element and shifted to the local elements in the grid. In this case the fourth step in Algorithm 1 can be skipped, since the level independent boundedness of the multiwavelets in (3.8) is automatically satisfied. For instance, for Cartesian grid hierarchies we apply Algorithm 1 on  $[0, 1]^2$ . In Figure 2 examples of these multiwavelets are shown. We list a set of constructed multiwavelets for  $p = 1, 2, 3$  in the Appendix. These multiwavelets are used for the computations in Section 5. For details on the

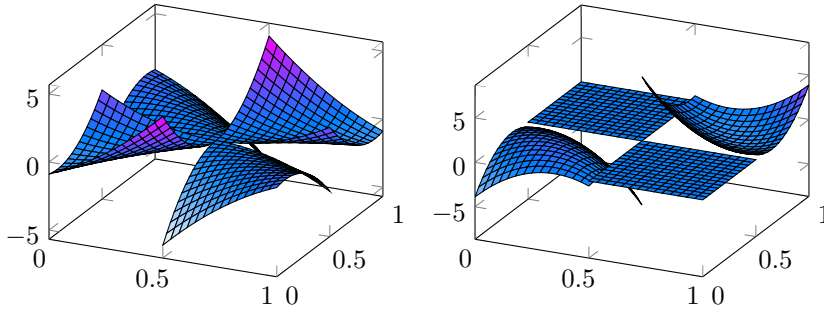


Figure 2: Examples of two-dimensional multiwavelets on a dyadic Cartesian grid hierarchy for  $p = 3$

construction on triangular grids we refer to [47].

### 3.5 Cancellation property

The vanishing moments (3.10) and the normalization (3.8) imply that the details become small with increasing refinement level when the underlying function is

locally smooth, i.e.,

$$|\langle u, \psi_{\mathbf{i}} \rangle_{\Omega}| \leq \inf_{P \in \Pi_{M-1}} |\langle u - P, \psi_{\mathbf{i}} \rangle_{\Omega}| \lesssim 2^{-\ell M d/2} \|u\|_{H^M(V_{\lambda})} \quad \mathbf{i} \in \mathcal{I}_{\ell}^W \quad (3.11)$$

holds for any function  $u$  in the Sobolev space  $H^M(V_{\lambda})$ . Here we assume that the grid hierarchy is quasi-uniform in the sense that the diameters and the volumes of the cells on each level  $\ell$  are proportional to  $2^{-\ell}$  and  $2^{-d\ell}$ , respectively, i.e.,

$$\text{diam } V_{\lambda} \sim 2^{-\ell} \quad \text{and} \quad |V_{\lambda}| \sim 2^{-d\ell},$$

where “ $\sim$ ” denotes an upper and a lower bound up to some constant independent of the level and the position encoded by the index  $\lambda$ . More precisely, the details decay at a rate of at least  $2^{-\ell M d/2}$ , provided that the function  $u$  has sufficient regularity on the support of the multiwavelet, and the multiwavelet has vanishing moments of order  $M$ . In fact, the higher  $M$  the more details may be discarded in smooth regions. Therefore, opposite to biorthogonal wavelets, the number of vanishing moments can be easily improved by increasing the order  $p$  *without* enlarging the support. The multiwavelets constructed with Algorithm 1 have vanishing moments of order  $p$ . Note that the algorithm can be extended by one additional step to obtain higher vanishing moments.

### 3.6 Multiscale transformation

In order to exploit the potential of data compression motivated by the cancellation property (3.11) we make use of the multi-scale transformation (3.4), and determine the relation between the basis coefficients in the single-scale and the multi-scale representation of  $u_L \in S_L$ :

$$u_L = \sum_{\mathbf{i} \in \mathcal{I}_L^S} v_{\mathbf{i}} \varphi_{\mathbf{i}} = \sum_{\mathbf{i} \in \mathcal{I}_0^S} v_{\mathbf{i}} \varphi_{\mathbf{i}} + \sum_{\ell=0}^{L-1} \sum_{\mathbf{i} \in \mathcal{I}_{\ell}^W} d_{\mathbf{i}} \psi_{\mathbf{i}}. \quad (3.12)$$

Similar to (2.5) we conclude from the bi-orthogonality relation (3.9) that the single-scale coefficients  $v_{\mathbf{i}}$  and the detail coefficients  $d_{\mathbf{i}}$  are determined as the inner products of  $u_L$  with the dual basis functions, i.e.,

$$v_{\mathbf{i}} = \langle u_L, \tilde{\varphi}_{\mathbf{i}} \rangle \quad \text{and} \quad d_{\mathbf{i}} = \langle u_L, \tilde{\psi}_{\mathbf{i}} \rangle.$$

Similar to the two-scale relation of the scaling functions and multiwavelets in Proposition 1 we can derive from (3.12) two-scale relations for these coefficients:

**Proposition 2** *For the single-scale coefficients and the detail coefficients the two-scale relations hold:*

$$\begin{aligned} (i) \quad v_{\lambda, \mathbf{i}} &= \sum_{\mu \in \mathcal{M}_{\lambda}} \sum_{\mathbf{i}' \in \mathcal{P}} v_{\mu, \mathbf{i}'} \langle \varphi_{\mu, \mathbf{i}'}, \tilde{\varphi}_{\lambda, \mathbf{i}} \rangle_{V_{\mu}}, \quad \lambda \in \mathcal{I}_{\ell}, \mathbf{i} \in \mathcal{P}, \ell = 0, \dots, L-1, \\ (ii) \quad d_{\lambda, \mathbf{i}} &= \sum_{\mu \in \mathcal{M}_{\lambda}} \sum_{\mathbf{i}' \in \mathcal{P}} v_{\mu, \mathbf{i}'} \langle \varphi_{\mu, \mathbf{i}'}, \tilde{\psi}_{\lambda, \mathbf{i}} \rangle_{V_{\mu}}, \quad \lambda \in \mathcal{I}_{\ell}, \mathbf{i} \in \mathcal{P}^*, \ell = 0, \dots, L-1, \\ (iii) \quad v_{\mu, \mathbf{i}} &= \sum_{\mathbf{i}' \in \mathcal{P}} v_{\lambda, \mathbf{i}'} \langle \varphi_{\lambda, \mathbf{i}'}, \tilde{\varphi}_{\mu, \mathbf{i}} \rangle_{V_{\mu}} + \sum_{\mathbf{i}' \in \mathcal{P}^*} d_{\lambda, \mathbf{i}'} \langle \psi_{\lambda, \mathbf{i}'}, \tilde{\varphi}_{\mu, \mathbf{i}} \rangle_{V_{\mu}}, \quad \mu \in \mathcal{M}_{\lambda}, \lambda \in \mathcal{I}_{\ell}, \mathbf{i} \in \mathcal{P}, \ell = 1, \dots, L. \end{aligned}$$

Recursively applying the two-scale relations (i) and (ii) of Proposition 2 the array of single-scale coefficients corresponding to the finest discretization level

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

can be transformed into a sequence of coarse grid data  $\mathbf{u}_0$  and details  $\mathbf{d}_\ell$

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

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*  $\mathcal{M}_L : \mathbf{v}_L \rightarrow (\mathbf{v}_0, \mathbf{d}_0, \dots, \mathbf{d}_{L-1})$ . It is reversed by recursively applying the two-scale relation (iii) in Proposition 2. The resulting *inverse multi-scale transformation* is described by the *inverse multi-scale operator*  $\mathcal{M}_L^{-1} : (\mathbf{v}_0, \mathbf{d}_0, \dots, \mathbf{d}_{L-1}) \rightarrow \mathbf{v}_L$ . In case of quadrilateral or triangular elements these transformations can be realized in a very efficient way, because the inner products in the relations in Proposition 2 have to be computed only once and to be stored for the reference element. Then during run-time, the transformations reduce to a local application of matrix-vector products to the local vectors of coefficients corresponding to one cell.

### 3.7 Thresholding

Due to the cancellation property (3.11), detail coefficients may become negligibly small whenever the underlying function is locally smooth. This gives rise to *hard thresholding* characterized by the index set

$$\mathcal{D}_\varepsilon := \{\mathbf{i} : |d_{\mathbf{i}}| > \varepsilon_\ell, \mathbf{i} \in \mathcal{I}_\ell^W, \ell = 0, \dots, L-1, \}.$$

The level-dependent threshold values  $\varepsilon_\ell$  are determined by an appropriate scaling of the threshold value  $\varepsilon \geq 0$  that is to be discussed later.

For an arbitrary index set  $\mathcal{D}$  of multi-scale coefficients the threshold operator  $\mathcal{T}_{\mathcal{D}} : (\mathbf{v}_0, \mathbf{d}_0, \dots, \mathbf{d}_{L-1}) \rightarrow (\tilde{\mathbf{v}}_0, \tilde{\mathbf{d}}_0, \dots, \tilde{\mathbf{d}}_{L-1})$  is defined element-wise by

$$\tilde{d}_{\mathbf{i}} := \begin{cases} d_{\mathbf{i}}, & \mathbf{i} \in \mathcal{D} \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad \tilde{v}_{\mathbf{i}} := v_{\mathbf{i}}, \quad \mathbf{i} \in \mathcal{I}_0^S.$$

From this operator we define the approximation operator

$$\mathcal{A}_{\mathcal{D}} := \mathcal{M}_L^{-1} \mathcal{T}_{\mathcal{D}} \mathcal{M}_L, \quad (3.13)$$

and, in particular for  $\mathcal{D} = \mathcal{D}_\varepsilon$ ,

$$\mathcal{A}_\varepsilon := \mathcal{M}_L^{-1} \mathcal{T}_{\mathcal{D}_\varepsilon} \mathcal{M}_L. \quad (3.14)$$

For any  $u_L \in S_L$  we then can define the approximation  $u_{L,\varepsilon}$  corresponding to (3.14):

$$u_{L,\mathcal{D}_\varepsilon} = \sum_{\mathbf{i} \in \mathcal{I}_0^S} v_{\mathbf{i}} \varphi_{\mathbf{i}} + \sum_{\ell=0}^{L-1} \sum_{\mathbf{i} \in \mathcal{D}_\varepsilon^\ell} d_{\mathbf{i}} \psi_{\mathbf{i}}, \quad (3.15)$$

where  $\mathcal{D}_\varepsilon^\ell := \mathcal{I}_\ell^W \cap \mathcal{D}_\varepsilon$ . The error introduced by thresholding can be estimated proceeding in analogy to classical wavelet analysis, cf. [17].

**Proposition 3** (*Approximation error*)

Let  $\Omega$  be a bounded domain,  $L \in \mathbb{N}$ ,  $u_L \in S_L$  and  $\varepsilon_\ell = a^{\ell-L}\varepsilon$  with  $a > 1$ . Then 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.,

$$\|u_L - u_{L, \mathcal{D}_\varepsilon}\|_{L^q(\Omega)} \leq \frac{C|\mathcal{P}^*||\Omega|^{\frac{1}{q}}}{a-1}\varepsilon, \quad (3.16)$$

where  $C$  is the uniform upper bound of the primal basis functions with respect to  $L^\infty(\Omega)$  given in (3.8).

This proposition implies that we may approximate any function  $u_L \in S_L$  by the approximation  $u_{L, \mathcal{D}_\varepsilon}$  up to an error that is proportional to the threshold value but at the expense of less degrees of freedom, because  $\mathcal{D}_\varepsilon$  can be significantly smaller than the full set  $\mathcal{D}_0 = \bigcup_{\ell=0}^{L-1} \mathcal{I}_\ell^W$ . In particular, Proposition 3 suggests how to choose the level-dependent threshold values  $\varepsilon_\ell$  for a given  $\varepsilon \geq 0$ .

### 3.8 Grid adaptation

From the index set  $\mathcal{D}$  we now determine a locally refined grid. This will be characterized by an index set  $\mathcal{G} \subset \{\lambda : \lambda \in \mathcal{I}_\ell, \ell = 0, \dots, L\}$  satisfying  $\Omega = \bigcup_{(\lambda) \in \mathcal{G}} V_\lambda$  with  $|V_\lambda \cap V_{\lambda'}| = 0$  for  $\lambda \neq \lambda'$ . How to determine  $\mathcal{G}$  is detailed in the sequel: First of all, we assume that  $\mathcal{D}$  is a tree, i.e., the relation

$$(\mu, i) \in \mathcal{D} \Rightarrow (\mu, j) \in \mathcal{D} \text{ and } (\lambda, j) \in \mathcal{D} \quad \forall i, j \in \mathcal{P}^* \quad (3.17)$$

holds for any  $\mu \in \mathcal{M}_\lambda$ ,  $\lambda \in \mathcal{I}_\ell$ ,  $\ell \in \{1, \dots, L-1\}$ . Note that in case of adaptive MR-FV schemes [37] the tree needs to be graded of degree  $q \geq 1$ . This leads

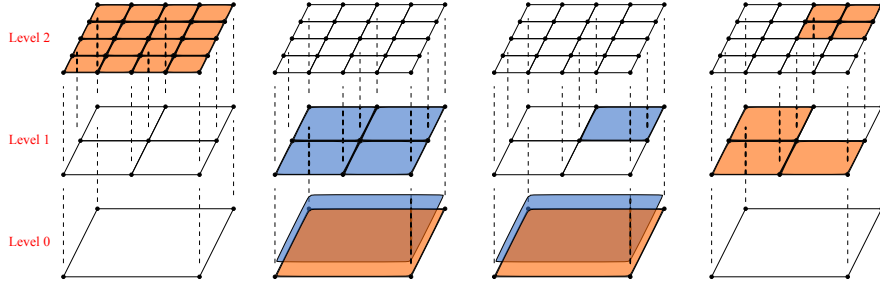


Figure 3: Multi-scale transformation, thresholding and the adaptive grid.

to larger trees and, hence, degrades the efficiency. Then  $\mathcal{G}$  can be determined recursively. For this purpose, the index set  $\mathcal{G}$  is initialized by all indices of the coarsest discretization. Then, traversing through the levels from coarse to fine we proceed as follows: if  $(\lambda, i) \in \mathcal{D}$  for some  $i \in \mathcal{P}^*$ , then the cell  $V_\lambda$  is locally refined, i.e., the index  $\lambda$  is removed from  $\mathcal{G}$  and the indices  $\mu \in \mathcal{M}_\lambda$  of the sub cells on the finer level are added to  $\mathcal{G}$ . Finally, we obtain a locally adapted grid which naturally corresponds to the outer leaves of the tree of significant details, i.e.,

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

For an illustration of the multi-scale transformation and the adaptive grid corresponding to the set  $\mathcal{D}$ , computed by the thresholding operation, see Figure 3.

## 4 Adaptive multiresolution DG scheme

In order to derive our adaptive scheme we now combine the MRA in Section 3 with the reference DG scheme presented in Section 2. Here again we will confine ourselves to the semi-discretization. The extension to the full discretization in space *and* time is straightforward. The idea is first to derive evolution equations for the single-scale and the multi-scale coefficients, respectively, and then to apply thresholding to discard non-significant contributions and, hence, to reduce the complexity of the evolution system. Starting point is a given DG discretization that is defined on the uniform highest resolution level  $L$ , i.e.,  $u_L = u_h$  with coefficients  $v_{\lambda,i}$ ,  $\lambda \in \mathcal{I}_L$ ,  $i \in \mathcal{P}$ , in (2.4). This will be referred to as the *reference scheme* on the *reference mesh*.

### 4.1 Evolution equations for single-scale and multi-scale coefficients

In a first step we now apply the two-scale relation (i) in Proposition 2 for the single-scale coefficients to the evolution equation (2.7). Proceeding level-wise from fine to coarse we then recursively obtain for any  $\lambda \in \mathcal{I}_\ell$ ,  $\mathbf{i} \in \mathcal{P}$ , the two-scale evolution equations for the single-scale coefficients

$$\frac{\partial v_{\lambda,i}}{\partial t} = G_{\lambda,i} - B_{\lambda,i}, \quad (4.1)$$

where the terms on the right-hand side correspond to the volume integral and the boundary integral in (2.7) that are recursively determined for  $\ell = 0, \dots, L-1$  and  $(\lambda, i) \in \mathcal{I}_\ell^S$  by

$$B_{\lambda,i} := \sum_{\mu \in \mathcal{M}_\lambda} \sum_{i' \in \mathcal{P}} \langle \varphi_{\mu,i'}, \tilde{\varphi}_{\lambda,i} \rangle_{V_\mu} B_{\mu,i'} = \int_{\partial V_\lambda} \hat{\mathbf{f}}(u_L^+, u_L^-, \vec{n}_\lambda) \tilde{\varphi}_{\lambda,i} dS \quad (4.2)$$

$$G_{\lambda,i} := \sum_{\mu \in \mathcal{M}_\lambda} \sum_{i' \in \mathcal{P}} \langle \varphi_{\mu,i'}, \tilde{\varphi}_{\lambda,i} \rangle_{V_\mu} G_{\mu,i'} = \sum_{\mu \in \mathcal{M}_\lambda} \int_{V_\mu} \mathbf{f}(u_L) \cdot \nabla \tilde{\varphi}_{\lambda,i} d\mathbf{x}. \quad (4.3)$$

Here we use the two-scale relations for the scaling functions and coefficients in Proposition 1 and 2.

Note that for the boundary integral (4.2) only those terms on the left-hand side have to be computed that have an intersection with the boundary  $\partial V_\lambda$  of the coarse cell whereas all internal boundaries cancel due to the nestedness of the grid hierarchy and the consistency property of the numerical fluxes.

In a second step, we similarly derive evolution equations for the details where we recursively apply the two-scale relation (ii) in Proposition 2 for the detail coefficients to the evolution equation (4.1) of the single-scale coefficients for  $\ell = L-1, \dots, 0$ . Then the evolution process (2.7) on the uniform reference mesh is equivalent to the evolution of the multi-scale coefficients, i.e., coarse-

scale coefficients and details,

$$\frac{\partial v_{\lambda,i}}{\partial t} = G_{\lambda,i} - B_{\lambda,i}, \quad (\lambda, i) \in \mathcal{I}_0^S \quad (4.4)$$

$$\frac{\partial d_{\lambda,i}}{\partial t} = \bar{G}_{\lambda,i} - \bar{B}_{\lambda,i}, \quad (\lambda, i) \in \mathcal{I}_\ell^W, \ell = 0, \dots, L-1. \quad (4.5)$$

Here the numerical flux balance, the flux integral and the source integral differ from those in (4.1):

$$\bar{B}_{\lambda,i} := \sum_{\mu \in \mathcal{M}_\lambda} \sum_{i' \in \mathcal{P}} \langle \varphi_{\mu,i'}, \tilde{\psi}_{\lambda,i} \rangle_{V_\mu} B_{\mu,i'} \quad (4.6)$$

$$\bar{G}_{\lambda,i}^n := \sum_{\mu \in \mathcal{M}_\lambda} \sum_{i' \in \mathcal{P}} \langle \varphi_{\mu,i'}, \tilde{\psi}_{\lambda,i} \rangle_{V_\mu} G_{\mu,i'}. \quad (4.7)$$

Since the multiwavelets  $\psi_{\lambda,i}$  typically are discontinuous inside the support  $V_\lambda$ , the numerical fluxes at internal edges corresponding to subcells of  $V_\lambda$  do *not* cancel in (4.6).

According to the change of basis (3.12) applied to (2.4), the evolution equation (2.7) of the reference scheme can be rewritten equivalently as

$$\frac{\partial u_L}{\partial t} = \sum_{\mathbf{i} \in \mathcal{I}_L^S} \frac{\partial v_{\mathbf{i}}}{\partial t} \varphi_{\mathbf{i}} = \sum_{\mathbf{i} \in \mathcal{I}_0^S} \frac{\partial v_{\mathbf{i}}}{\partial t} \varphi_{\mathbf{i}} + \sum_{\ell=0}^{L-1} \sum_{\mathbf{i} \in \mathcal{I}_\ell^W} \frac{\partial d_{\mathbf{i}}}{\partial t} \psi_{\mathbf{i}}, \quad (4.8)$$

where the derivatives of the multi-scale coefficients are determined by (4.4) and (4.5). We emphasize that up to now we only derived an equivalent representation of the reference DG Scheme (2.7) without having reduced the overall complexity that is still proportional to the reference mesh. How to reduce the costs without losing significantly in accuracy is explained in the following.

## 4.2 Data compression

As is motivated by the cancellation property (3.11), we may reduce the degrees of freedom by first applying *data compression* to the multi-scale coefficients where the approximation error can be controlled according to Proposition 3. Then the compressed set of equations is solved on a locally refined *adaptive grid* characterized by some index set  $\mathcal{G}$ . According to Section 3.8 this set is determined by means of an index set  $\mathcal{D}$  corresponding to significant detail coefficients. Note that, if  $\mathcal{D}$  is a tree, the system of evolution equations for the single-scale coefficients (4.1) for  $\mathcal{G}$  is equivalent to the system of evolution equations for the multi-scale coefficients (4.4) and (4.5), i.e., by the sets  $\mathcal{D}$  and  $\mathcal{G}$  a local change of basis is performed that, in analogy to (3.12), reads

$$\frac{\partial u_{L,\mathcal{D}}}{\partial t} = \sum_{(\lambda,i) \in \mathcal{G} \times \mathcal{P}} \frac{\partial v_{\mathbf{i}}}{\partial t} \varphi_{\lambda,i} = \sum_{\mathbf{i} \in \mathcal{I}_0^S} \frac{\partial v_{\mathbf{i}}}{\partial t} \varphi_{\mathbf{i}} + \sum_{\mathbf{i} \in \mathcal{D}} \frac{\partial d_{\mathbf{i}}}{\partial t} \psi_{\mathbf{i}}. \quad (4.9)$$

The transformation between the two representations characterized by the local single-scale coefficients  $\mathbf{v}_{\mathcal{G}} = (v_{\lambda,i})_{\lambda \in \mathcal{G}, i \in \mathcal{P}}$  and the compressed multi-scale coefficients  $\mathbf{v}_{\mathcal{D}} = ((v_{\mathbf{i}})_{\mathbf{i} \in \mathcal{I}_0^S}, (d_{\mathbf{i}})_{\mathbf{i} \in \mathcal{D}})$ , respectively, can be realized efficiently, i.e., the



number of operations is proportional to the number of cells  $\#\mathcal{G}$  and the number of significant detail coefficients  $\#\mathcal{D}$ , respectively. In [31] the transformation algorithms are explicitly given in the one-dimensional case.

Obviously, the number of evolution equations corresponding to the multi-scale coefficients has been reduced in (4.9). Due to the recursive definition of the integrals in (4.2) and (4.3) the coefficients are determined by the sum of *all* integrals on the *finest* scale  $L$ , although (4.1) is the evolution equation of the coefficients on the coarse scale  $\ell$ . Thus we still have the complexity of the reference grid. However, if the integrals can be computed *exactly*, then the sums can be replaced by one integral on the *local* scale  $\ell$ . Otherwise, we approximate the integrals on the right-hand side on the local scale by quadrature rules of *higher* order than those used in the reference scheme to avoid loss of accuracy and to reduce the complexity of the reference mesh to that of the adaptive grid. In particular, one has to ensure that the error introduced by the quadrature rules is smaller than the error resulting from the thresholding procedure. In the one-dimensional case this has been investigated analytically in [43].

So far we have only considered the semi-discrete evolution equations, i.e., the sets  $\mathcal{G}$  and  $\mathcal{D}$  are time-dependent, since the underlying data by which we perform grid adaptation according to Section 3 depend on time. As we do for the reference scheme we now apply an explicit RK scheme for the time discretization in (4.9) and (4.1), (4.4), (4.5), respectively. Then for each discrete time  $t^n$  we perform grid adaptation with respect to the coefficients of this time level. Note that by means of the thresholding strategy we may only *coarsen* a given grid according to  $\mathcal{D}_\varepsilon^n$  at time  $t_n$ . However if we would apply the DG scheme on the coarsen grid, we cannot make sure that the grid is suitable for the solution at the next time level  $t_{n+1}$ . Therefore we need a refinement strategy to approximate  $\mathcal{D}_\varepsilon^{n+1}$ .

### 4.3 Prediction

We have to *predict* which details will be significant at the new time level  $t^{n+1}$ . Prediction can only be based on the information available at the old time level  $t^n$ , i.e., on the set  $\mathcal{D}_\varepsilon^n$  of significant details at time level  $t^n$ . In view of reliability an appropriate prediction set  $\tilde{\mathcal{D}}_\varepsilon^{n+1}$  has to satisfy the condition

$$\mathcal{D}_\varepsilon^n \cup \mathcal{D}_\varepsilon^{n+1} \subset \tilde{\mathcal{D}}_\varepsilon^{n+1}. \quad (4.10)$$

For the design of such a prediction set we follow the idea of Harten's strategy [27] developed in the finite volume context and, thus, operates on cell averages only. Instead of operating on zeroth order coefficients only, we consider also higher order coefficients:

- (i) 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.,

$$\max_{i \in \mathcal{P}^*} |d_{\lambda,i}| > \varepsilon_\ell \quad \Rightarrow \quad \{(\mu, i) : \mu \text{ shares an edge with } \lambda, i \in \mathcal{P}^*\} \subset \tilde{\mathcal{D}}_\varepsilon^{n+1}, \quad (4.11)$$

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

$$\max_{i \in \mathcal{P}^*} |d_{\lambda,i}| \geq 2^{\bar{M}+1} \varepsilon_\ell \quad \Rightarrow \quad \{(\mu, i) : \mu \in \mathcal{M}_\lambda, i \in \mathcal{P}^*\} \subset \tilde{\mathcal{D}}_\varepsilon^{n+1}, \quad (4.12)$$

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

In the finite volume context this turned out to give satisfactory results, cf. [9, 38], although the reliability condition (4.10) has never been proven to hold.

#### 4.4 Adaptive MR-DG scheme

Finally we summarize the adaptive MR-DG scheme. Note that because of (4.9) we may perform equivalently the evolution with respect to the single-scale representation and the multi-scale representation. However, from a practical point of view we prefer to evolve the single-scale according to (4.1) on a locally refined grid for two reasons: (i) the evaluation of  $u_{L, \mathcal{D}_\varepsilon}$  that has to be performed when computing the fluxes is more efficient and in the single-scale basis than the multiwavelet basis (ii) the single-scale coefficients enter explicitly the limiting process.

In Algorithm 2 we summarize the main steps of our scheme. Therein prediction and thresholding are performed on the multiscale coefficients, while all the other operations are carried out on the single-scale coefficients.

**Algorithm 2** *One time-step of the adaptive scheme:*

(1) **Grid refinement:**

- (a) Apply multi-scale transformation to determine the detail coefficients,
- (b) perform prediction of detail coefficients that may become significant,
- (c) determine refined adaptive grid by applying the inverse multi-scale transformation.

(2) **Time evolution:**

*Perform the Runge-Kutta time evolution where at each stage of the RK scheme the single-scale coefficients are limited.*

(3) **Grid coarsening:**

- (a) Apply multi-scale transformation to determine the detail coefficients,
- (b) perform thresholding of detail coefficients,
- (c) determine adaptive grid by applying the inverse multi-scale transformation.

We emphasize that the time evolution in step (2) of the adaptive MR-DG scheme cannot be interpreted as a time step performed with a DG scheme where the adaptive grid is considered to be an *unstructured* grid. According to the definition of (4.2) and (4.3) these terms are strongly intertwined with the highest refinement level  $L$  by means of the multiscale transformation as has been discussed above.

## 4.5 Choice of the threshold value

The efficiency as well as the accuracy of adaptive MR-DG scheme is significantly influenced by only one parameter that is the threshold value  $\varepsilon$ . In the following we discuss a strategy how to choose this parameter. For this reason we focus our attention on the perturbation analysis that aims at controlling the threshold error which is introduced by the adaptive process. According to Proposition 3 the thresholding process introduces an error of order  $\varepsilon$  to the solution. In addition,  $\mathcal{D}^{n+1}$  cannot be computed exactly for the next time-step, but is predicted by a heuristic strategy instead. These additional errors may spoil the discretization error of the reference scheme, i.e., the non-adaptive DG scheme applied to a fully-refined grid on level  $L$ . The main idea behind our adaptive approach is to accelerate the convergence of this reference DG scheme without losing accuracy. For this purpose one has to balance the *discretization error* of the reference scheme, i.e., the difference between the exact solution and the reference scheme, and the *perturbation error*, i.e., the difference between the reference DG scheme and the adaptive scheme. To quantify this, we estimate the errors in one time step.

For any discrete time  $t^n \in [0, T]$  let  $u^n$  be the exact solution,  $u_L^n \in S_L$  be the solution of the reference scheme and  $u_{L,\varepsilon}^n \in S_L$  be the solution of the adaptive DG scheme with threshold value  $\varepsilon$ . Then we can estimate the *full error* of the adaptive scheme  $e^n := u^n - u_{L,\varepsilon}^n$  by the sum of the discretization error  $\eta^n := u^n - u_L^n$  of the reference scheme and the perturbation error  $\nu^n := u_L^n - u_{L,\varepsilon}^n$ :

$$\|u^n - u_{L,\varepsilon}^n\|_{L^1(\Omega)} \leq \|\eta^n\|_{L^1(\Omega)} + \|\nu^n\|_{L^1(\Omega)} \leq TOL.$$

If there is an a priori estimate available for the discretization error of the reference scheme, i.e.,  $\|\eta^n\|_{L^1(\Omega)} \sim h^\beta$  where  $h$  denotes the spatial step size and  $\beta$  the order of convergence, then we would choose  $h$  such that  $h^\beta \sim TOL$ . In order to preserve the accuracy of the reference scheme we may then admit a perturbation error that is proportional to the discretization error, i.e.,

$$\|\eta^n\|_{L^1(\Omega)} \sim \|\nu^n\|_{L^1(\Omega)}. \quad (4.13)$$

This implies that  $h = h(TOL, \beta)$  and  $\varepsilon = \varepsilon(h)$ . Let denote the threshold value satisfying (4.13) by  $\varepsilon_{\text{ideal}}$ .

In the one-dimensional case it was proven in [31, 43] that for a uniform dyadic refined grid hierarchy the  $L^1$ -norm of the perturbation error *in the mean* can be bounded *uniformly* up to a constant by  $(\varepsilon + h_L^\alpha + h_L^\gamma)/\tau_L$ . Here  $\alpha > 1$  is a constant coming from stability assumptions on the reference scheme and  $\gamma > 1$  corresponds to the order of the quadrature rules used to approximate the integrals on the right-hand side of (4.2) and (4.3). Due to a CFL constraint the time step size  $\tau_L$  associated to the finest refinement level is proportional to  $h_L$ . Division by  $\tau_L$  thus accounts for the fact that in each time step we may introduce in the worst case a threshold error of order  $\varepsilon$  that accumulates over all time steps. For  $\alpha \geq \beta + 1$  and  $\gamma \geq \beta + 1$  we then may choose  $\varepsilon \sim h_L^{\beta+1}$  to ensure that (4.13) holds. Since the stability analysis in [31, 43] relies on a priori estimates of the reference scheme, in general this choice will be too pessimistic, i.e., the adaptive grid will be too dense and, thus, spoiling the efficiency of the adaptive grid without improving the accuracy of the reference scheme. On the other hand, for real computations it will not be possible to determine  $\varepsilon_{\text{ideal}}$

exactly. Motivated by numerous computations in [43] it was suggested to use the empirical estimate

$$\varepsilon \sim h_L^\beta, \quad (4.14)$$

that gives good results, where the power  $\beta$  can be chosen as 1.

## 5 Numerical results

To demonstrate the ability of reducing the computational cost of the reference DG scheme (2.7) while preserving its accuracy using multiwavelet-based grid adaptation, we apply our scheme to four test cases: a linear test case, where all discontinuities are contact discontinuities, and three non-linear test cases, where we have shocks. For the first two test cases the exact solution is known. Therefore we can identify the ideal threshold value  $\varepsilon_{\text{ideal}}$ .

Here we confine ourselves to two-dimensional problems. For the discretization we use quadrilateral grids, where the grid hierarchy is obtained by a uniform dyadic partition of  $\Omega$ , i.e.,  $|\mathcal{M}_\lambda| = 4$  for all  $\lambda \in \mathcal{I}_\ell$ ,  $\ell = 0, \dots, L-1$ . Then the size of the cells  $V_\lambda$ ,  $\lambda \in \mathcal{I}_\ell$ , on a fixed level  $\ell$  is constant. Motivated by Proposition 3, we choose  $\varepsilon_\ell = 2^{\ell-L}\varepsilon$  for the level-dependent threshold values in our computations. All computations are performed using quadratic polynomials, i.e.,  $p = 3$ , and the SSPRK(3,3) method is used for the discretization in time. The time step size is computed in each time step with respect to the highest refinement level in the grid and a constant CFL number of 0.08. The integrals are approximated with Gaussian quadrature rules, such that polynomials with degree less or equal than 7 are integrated exactly.

### 5.1 Linear transport equation

In this section we consider a two-dimensional linear transport equation:

$$\frac{\partial u}{\partial t} + \nabla \cdot (\vec{b}u) = 0$$

on the domain  $\Omega = [0, 1]^2$  with periodic boundary conditions. For this equation the exact (entropy) solution is known. We verify the accuracy of our reference DG scheme by computing the numerical order of accuracy for a smooth test case. For this purpose we choose

$$u_0(\mathbf{x}) = \sin(2\pi x_1) \sin(2\pi x_2) \quad \text{and} \quad \vec{b} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

Then the solution of (2.1) is given by

$$u(\mathbf{x}, t) = \sin(2\pi x_1 - 2\pi t) \sin(2\pi x_2).$$

Since this solution is smooth no limiting is needed. We performed several computations on uniform grids up to  $T = 0.5$  and determined the empirical order of convergence (EoC). The errors as well as the EoC are recorded in Table 1. The results confirm that our reference scheme is third order accurate.

Next we consider a discontinuous solution of (2.1) imposing non-smooth initial data:

$$u_0(\mathbf{x}) = \begin{cases} q(\sqrt{x_1^2 + x_2^2}), & \text{if } x_1^2 + x_2^2 \leq 1 \text{ and } x_1 > -0.2 \\ 0, & \text{else} \end{cases},$$

where  $q$  is defined as:

$$q(r) := 1 - r^4 + 4r^4(r - 1) - 10r^4(r - 1)^2 + 20r^4(r - 1)^3.$$

The function  $q$  is designed in such a way that  $u_0$  is smooth almost everywhere, but exhibits a discontinuity at  $x_1 = 0.2$  for  $x_1^2 + x_2^2 \leq 1$ . They are advected in the direction  $\vec{b} = (1, 0.2)^T$ . Since the problem is linear, the solution for  $t = 2.5$  is given by

$$u(\mathbf{x}, t) = u_0(x_1, x_2 - 0.2t).$$

Computations are performed on  $\Omega = [-1.25, 1.25] \times [-1.05, 1.45]$  using periodic boundary conditions. Since the solution exhibits a discontinuity we need limiting.

| Number of Cells    | smooth test case |                | non-smooth test case |                |
|--------------------|------------------|----------------|----------------------|----------------|
|                    | Error in $L_1$   | Order in $L_1$ | Error in $L_1$       | Order in $L_1$ |
| $4 \times 4$       | 4.322e-02        |                | 7.662e-1             |                |
| $8 \times 8$       | 7.431e-03        | 2.540          | 5.245e-1             | 0.5470         |
| $16 \times 16$     | 1.136e-03        | 2.710          | 2.246e-1             | 1.224          |
| $32 \times 32$     | 1.596e-04        | 2.831          | 7.822e-2             | 1.521          |
| $64 \times 64$     | 2.128e-05        | 2.906          | 3.099e-2             | 1.336          |
| $128 \times 128$   | 2.750e-06        | 2.952          | 1.460e-2             | 1.086          |
| $256 \times 256$   | 3.496e-07        | 2.976          | 7.416e-3             | 0.977          |
| $512 \times 512$   | 4.407e-08        | 2.988          | 3.923e-3             | 0.919          |
| $1024 \times 1024$ | 5.532e-09        | 2.993          | 2.087e-3             | 0.911          |

Table 1: Empirical order of convergence for the linear transport equation.

To verify the capability of the reference scheme to deal with discontinuities we performed several solutions on different uniform grids and determined the EoC. For this case the EoC is one in the  $L_1$ -norm that is optimal for a discontinuous solution. The results are shown in Table 1.

For the adaptive computations we use  $2 \times 2$  cells on the coarsest level  $\ell = 0$  and  $L = 8$  refinement levels. This corresponds to a grid with 262,144 cells on the finest level on which we perform the non-adaptive reference scheme. In order to compare the adaptive solutions with the solution of the reference scheme, we need the discretization error of the reference scheme, which can be found in Table 1.

Exemplarily, we present in Figure 4 the adaptive solution and the corresponding grid for  $\varepsilon = 0.003$  at  $T = 0.5$ . We conclude from the adaptive grid that in regions where the solution is smooth, the grid is very coarse and it is fully refined only in a small area around the discontinuity. In regions where the solution is smooth but not constant, the grid refinement is moderate.

To determine the ideal threshold value  $\varepsilon_{\text{ideal}}$ , we need to balance the perturbation error with the discretization error of the reference scheme. For this

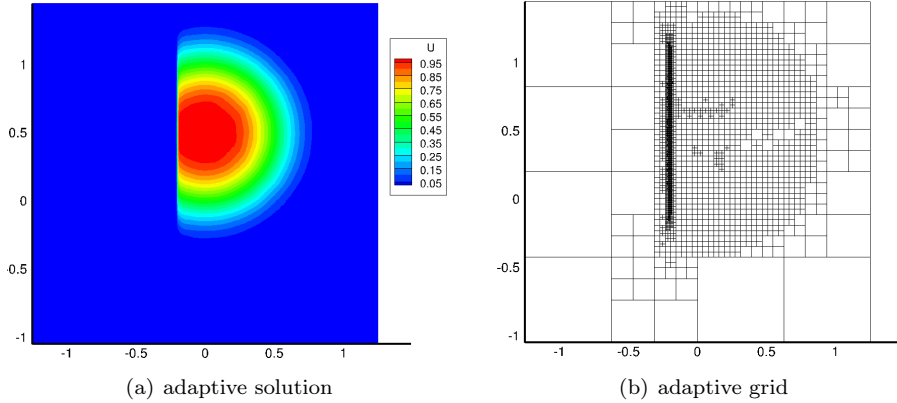


Figure 4: Adaptive solution and corresponding grid for the linear test case with non-smooth initial data for  $L = 8$ .

purpose we perform a parameter study where we fix  $L$  and vary  $\varepsilon$ . Then we compute the perturbation error  $\|u_{L,\varepsilon}^n - u_L^n\|_{L^1(\Omega)}$  and can determine the ideal threshold value  $\varepsilon_{\text{ideal}}$ .

The behavior of the perturbation and the full error is depicted in Figure 5(a). For  $\varepsilon$  tending to zero the perturbation error tends to zero. The perturbation error is monotonically decreasing in  $\varepsilon$ . For all considered values of  $\varepsilon$  the full error is in the same order of magnitude, i.e., the method is not very sensitive in the choice of the threshold value  $\varepsilon$ . Here the ideal threshold value can be depicted approximately from Figure 5(a) to be  $\varepsilon_{\text{ideal}} \approx 3.21 \cdot 10^{-3}$ . The estimate (4.14) suggests to use  $\varepsilon \approx 4.88 \cdot 10^{-3}$ . So the estimated  $\varepsilon$  is a slightly larger than  $\varepsilon_{\text{ideal}}$ , but nevertheless it is close to  $\varepsilon_{\text{ideal}}$ . Therefore it is a reliable estimate in this test case.

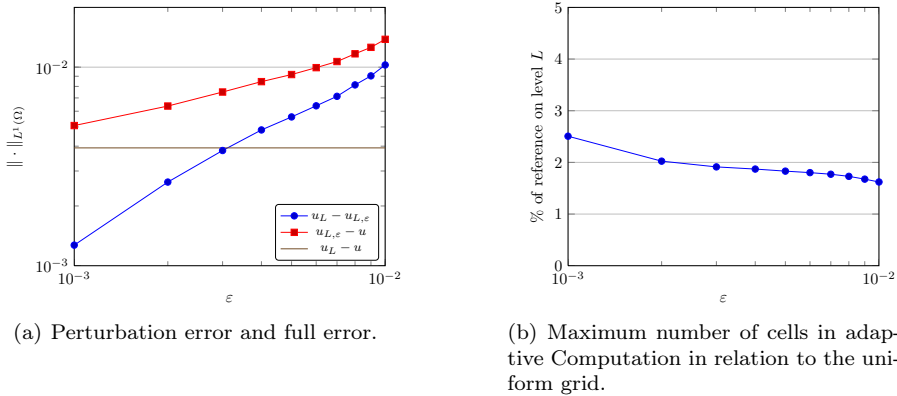


Figure 5: Error analysis and efficiency for the linear test case with non-smooth initial data for  $L = 8$ .

In Figure 5(b) the behavior of the maximal number of cells in the adaptive grid is shown. For all considered values of  $\varepsilon$  the maximal number of cells in the corresponding grid is about 2 percent of the reference grid. For this test

case our method is very efficient and accurate: for  $\varepsilon = 0.003$  the full error of the adaptive scheme is  $7.496 \cdot 10^{-3}$  and the error of the reference scheme  $3.923 \cdot 10^{-3}$ , but the maximum number of cells in the adaptive grid is only 1.9 percent of number of cells of the reference scheme. If we allow for a full error that is twice the discretization error of the reference scheme, we slash the number of cells enormously. On the other hand we conclude from Figure 5(b): If our guess of the threshold value is not sharp but choose a threshold value that is smaller than  $\varepsilon_{ideal}$ , then the number of cells in the adaptive grid is not increasing very much. In this sense the strategy is robust.

## 5.2 Inviscid Burgers' equation

To demonstrate the ability to deal with non-linear problems, we consider the two-dimensional Burgers' equation on  $\Omega = [0, 1]^2$ :

$$\frac{\partial u}{\partial t} + \frac{1}{2} \nabla \cdot \left( u^2 \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right) = 0,$$

that is subject to the initial data

$$u_0(x_1, x_2) = \begin{cases} -0.2 & , \text{ if } x_1 < 0.5 \text{ and } x_2 > 0.5 \\ -1 & , \text{ if } x_1 \geq 0.5 \text{ and } x_2 > 0.5 \\ 0.5 & , \text{ if } x_1 < 0.5 \text{ and } x_2 \leq 0.5 \\ 0.8 & , \text{ if } x_1 \geq 0.5 \text{ and } x_2 \leq 0.5 \end{cases}.$$

For this configuration an exact solution is given in [25]. We compute the solution at  $T = 0.5$ .

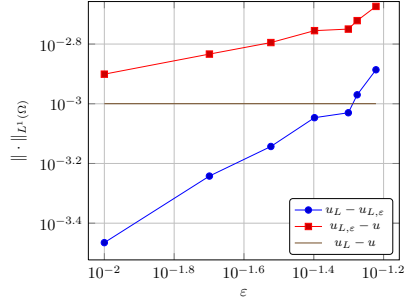
In order to verify our reference scheme, we have computed non-adaptive solutions for different grid sizes and thereby calculated the numerical order of accuracy. Since discontinuities are present in the solution, the reference scheme cannot be expected to have more than first order accuracy in  $L_1$ , which is confirmed in Table 2.

| L | Number of Cells  | Error in $L_1$ | Order in $L_1$ |
|---|------------------|----------------|----------------|
| 2 | $12 \times 12$   | 5.636e-2       |                |
| 3 | $24 \times 24$   | 2.977e-2       | 0.921          |
| 4 | $48 \times 48$   | 1.479e-2       | 1.01           |
| 5 | $96 \times 96$   | 7.719e-3       | 0.938          |
| 6 | $192 \times 192$ | 3.954e-3       | 0.954          |
| 7 | $384 \times 384$ | 2.027e-3       | 0.964          |
| 8 | $768 \times 768$ | 1.001e-3       | 1.018          |

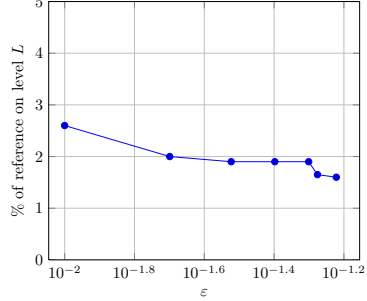
Table 2: Empirical order of convergence for the inviscid Burgers' equation at  $T = 0.5$ .

For the adaptive computations we consider different values for  $\varepsilon$  in the thresholding procedure. On the coarsest level  $\ell = 0$  we take  $3 \times 3$  cells and consider  $L = 8$  levels of refinement, this corresponds to 589,824 cells on the uniform reference grid. The discretization error of the reference scheme can be found in Table 2. Again we investigate the behavior of the perturbation error

and the full error. In Figure 6(a) we observe that for this configuration the ideal threshold value is about  $\varepsilon_{\text{ideal}} \approx 0.058$ .



(a) Perturbation error and full error



(b) Maximum number of cells in adaptive Computation in relation to the uniform grid.

Figure 6: Error analysis and efficiency for Burgers' equation for  $L = 8$ .

In this test case the estimate (4.14) suggests to use  $\varepsilon \approx 1.30 \cdot 10^{-3}$ . Obviously, the estimated value for  $\varepsilon$  is not in the same order of magnitude, but is significantly smaller. Thus the estimate is not sharp in this case. Nevertheless the estimated threshold value is acceptable, since it is smaller than  $\varepsilon_{\text{ideal}}$  and does not spoil the accuracy. However the computational costs for the corresponding computation would not be optimal. In this sense estimate (4.14) can be considered to be reliable, but not optimal.

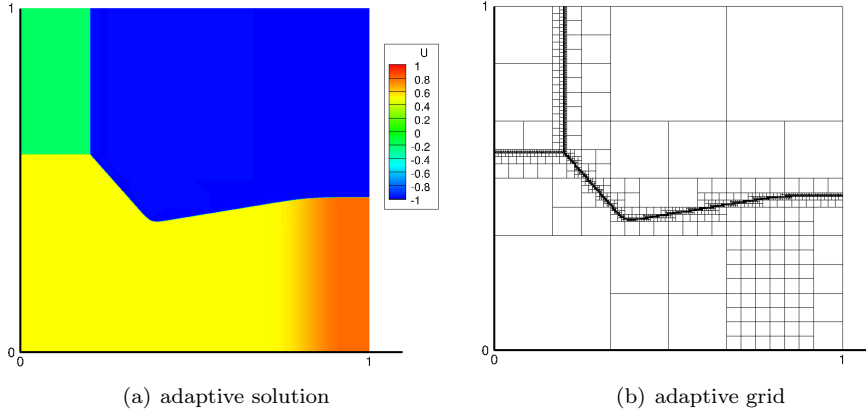


Figure 7: Burger's equation: Adaptive solution and corresponding grid for  $\varepsilon \approx 0.058$  and  $L = 8$  at  $T = 0.5$ .

In Figure 6(b) the behavior of the maximal number of cells in the adaptive grid is shown. For all considered values of  $\varepsilon$  the maximal number of cells in the corresponding grid is about 3 percent of the reference grid. Again our method is very efficient and accurate: for  $\varepsilon = 0.053$  the full error of the adaptive scheme is  $1.899 \cdot 10^{-3}$  and the error of the reference scheme  $1.001 \cdot 10^{-3}$ , but the maximum number of cells in the adaptive grid is only 1.9 percent of number of cells of the reference scheme.



In Figure 7 the adaptive solution and the corresponding grid for  $\varepsilon = 0.058$  at  $T = 0.5$  are shown. Again, the resolution of the grid is well adapted to the local features of the solution.

### 5.3 Rotating wave

For the two test cases considered above, the exact solution was available, and therefore we can determine the ideal threshold value  $\varepsilon_{\text{ideal}}$ . In the following two test cases the entropy solution is not known explicitly, so we cannot determine the ideal threshold value and we have to rely on an estimate for it. We consider a two-dimensional non-linear equation with non-convex fluxes on  $\Omega = [-2.5, 2.5] \times [-3, 2]$ :

$$\frac{\partial u}{\partial t} + \nabla \cdot \begin{pmatrix} \sin(u - \frac{\pi}{8}) \\ \cos(u - \frac{\pi}{8}) \end{pmatrix} = 0,$$

that is subject to the initial data

$$u_0(x_1, x_2) = \begin{cases} \frac{7}{2}\pi, & \text{if } x_1^2 + x_2^2 < 1 \\ \frac{1}{4}\pi, & \text{else} \end{cases}.$$

Starting from the initial data, the solution develops a complex wave structure. This is a modification of a challenging test case which was proposed in [35]. The solution is computed up to  $T = 1$ .

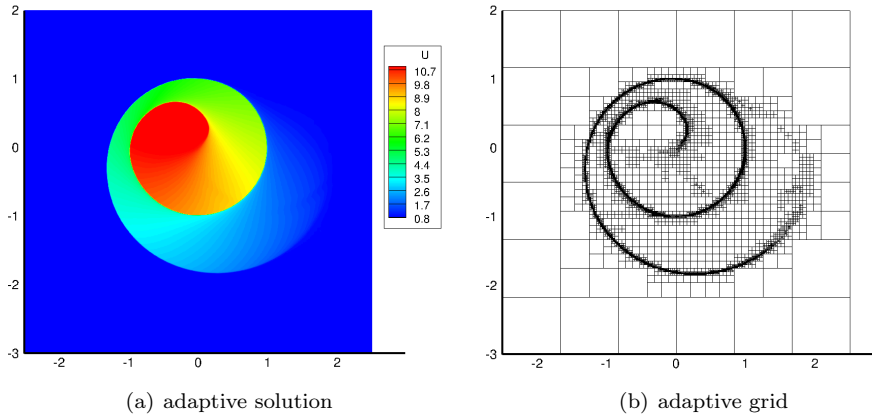


Figure 8: Rotating wave: Adaptive solution and corresponding grid for  $\varepsilon = 0.005$  and  $L = 8$  at  $T = 1$ .

On the coarsest level  $\ell = 0$  we have  $3 \times 3$  cells and consider  $L = 8$  levels of refinement. This corresponds to 589,824 cells on the uniform reference grid. According to (4.14) we choose  $\varepsilon = 0.005 \approx h_L$  in the thresholding procedure. In Figure 8 we present the adaptive solution and the corresponding grid. In this case the maximum number of cells in the adaptive grid is 26,856. We conclude from the adaptive grid that in regions where the solution is smooth, the grid is very coarse and it is fully refined only in a small area around the discontinuity. Here we obtain an acceptable grid, where the compression rate is 4.6 % in comparison to the reference mesh.

## 5.4 Buckley-Leverett

Finally we consider the two-dimensional Buckley-Leverett equation with gravitational effects in  $x_2$ -direction on  $\Omega = [-1, 1.7] \times [-1.1, 1.5]$ :

$$\frac{\partial u}{\partial t} + \nabla \cdot \left( \begin{array}{c} \frac{u^2}{u^2 + (1-u)^2} \\ \frac{u^2(1-5(1-u)^2)}{u^2 + (1-u)^2} \end{array} \right) = 0$$

that is subject to the initial data

$$u_0(x_1, x_2) = \begin{cases} 1.0, & \text{if } x_1^2 + x_2^2 < 0.5 \\ 0.0, & \text{else} \end{cases}.$$

Again, the fluxes are non-convex and starting from the initial data, the solution develops a complex wave structure. This test case was introduced in [11]. The solution is computed up to  $T = 0.5$  and, thus, can be compared with the results in [11, 25].

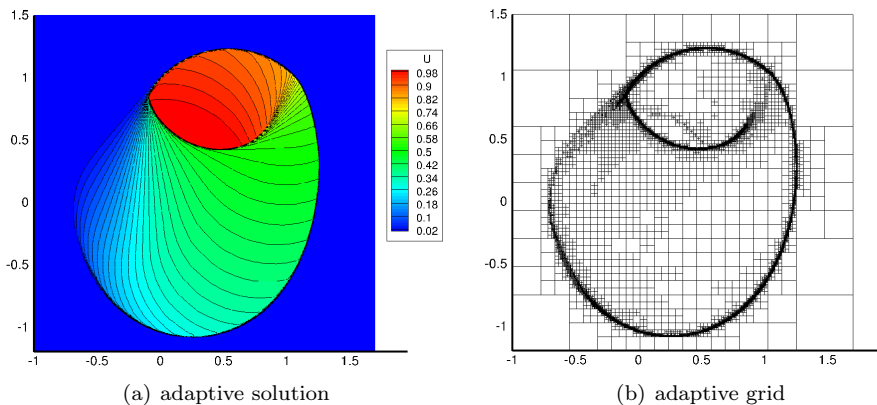


Figure 9: Buckley-Leverett: Adaptive solution and corresponding grid for  $\varepsilon = 0.003$  and  $L = 8$  at  $T = 0.5$ .

On the coarsest level  $\ell = 0$  we have  $3 \times 3$  cells and consider  $L = 8$  levels of refinement. This corresponds to 589,824 cells on the uniform reference grid. According to (4.14) we choose  $\varepsilon = 0.003 \approx h_L$  in the thresholding procedure. Here the maximum number of cells in the adaptive grid is 19,830. In Figure 9 we present the adaptive solution and the corresponding grid. Again, we observe that in regions where the solution is smooth, the grid is very coarse and it is locally refined up to the finest level only in a small area around the discontinuity. We obtain an appropriately refined grid with less than 3.4 % of the cells in the reference mesh.

## 6 Discussion and Conclusion

The concept of adaptive MR-DG schemes originally developed in one space dimension and investigated in [31, 43] has been extended to the multi-dimensional

case. In particular, we construct genuinely multi-dimensional multiwavelets following Alpert's general construction principle [3]. The computational performance and the accuracy of the scheme essentially relies on the choice of the threshold value. Here we give an empirical rule that is validated by means of linear and non-linear test cases using convex and non-convex fluxes. The reduction in the number of DOFs and computational time is shown to be very favorable. Even though a rigorous analysis as the one for the one-dimensional case [31, 43] is not yet available for the multi-dimensional case, the empirical rule for the choice of the thresholding tolerance gives satisfactory results.

#### Acknowledgements

Financial support has been provided by the German Research Foundation (Deutsche Forschungsgemeinschaft — DFG) in the framework of the Sonderforschungsbereich Transregio 40.

## References

- [1] B.C., Lin, S.Y., Shu, C.W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. *J. Comput. Phys.* **84**, 90–113 (1989)
- [2] Adjerid, S., Devine, K., Flaherty, J., Krivodonova, L.: A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems. *Comput. Methods Appl. Mech. Eng.* **191**, 1097–1112 (2002)
- [3] Alpert, B.: A class of bases in  $l^2$  for the sparse representation of integral operators. *SIAM J. Math. Anal.* **24**, 246–262 (1993)
- [4] Alpert, B., Beylkin, G., Gines, D., Vozovoi, L.: Adaptive solution of partial differential equation in multiwavelet bases. *J. Comp. Phys.* **182**, 149–190 (2002)
- [5] Archibald, R., Fann, G., Shelton, W.: Adaptive discontinuous Galerkin methods in multiwavelets bases. *Applied Numer. Math.* **61**, 879–890 (2011)
- [6] Arnold, D.N., Brezzi, F., Cockburn, B., Marini, L.D.: Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.* **39**(5), 1749–1779 (2002)
- [7] Barth, T., Jepserson, D.: The design and application of upwind schemes on unstructured meshes. *AIAA, 27th Aerospace Sciences Meeting* (1989)
- [8] Bey, K., Oden, J.:  $hp$ -version discontinuous Galerkin methods for hyperbolic conservation laws. *Comput. Methods Appl. Mech. Eng.* **133**(3-4), 259–286 (1996)
- [9] Bramkamp, F., Lamby, P., Müller, S.: An adaptive multiscale finite volume solver for unsteady and steady state flow computations. *J. Comp. Phys.* **197**(2), 460–490 (2004)
- [10] Calle, J., Devloo, P., Gomes, S.: Wavelets and adaptive grids for the discontinuous Galerkin method. *Numer. Algorithms* **39**(1-3), 143–154 (2005)

- [11] Christov, I., Popov, B.: New non-oscillatory central schemes on unstructured triangulations for hyperbolic systems of conservation laws. *J. Comp. Phys.* **227**, 5736–5757 (2008)
- [12] Cockburn, B., Hou, S., Shu, C.W.: The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV: The multidimensional case. *Math. Comp.* **54**(190), 545–581 (1990)
- [13] Cockburn, B., Karniadakis, G., Shu, C.W.: The development of discontinuous Galerkin methods. Cockburn, Bernardo (ed.) et al., *Discontinuous Galerkin methods. Theory, computation and applications*. 1st international symposium on DGM, Newport, RI, USA, May 24-26, 1999. Berlin: Springer. *Lect. Notes Comput. Sci. Eng.* 11, 3-50 (2000).
- [14] Cockburn, B., Shu, C.W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Math. Comp.* **52**(186), 411–435 (1989)
- [15] Cockburn, B., Shu, C.W.: The Runge-Kutta discontinuous Galerkin method for conservation laws v: Multidimensional systems. *J. Comput. Phys.* **141**, 199–244 (1998)
- [16] Cockburn, B., Shu, C.W.: Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *J. Sci. Computing* **16**(3), 173–261 (2001)
- [17] Cohen, A.: Numerical analysis of wavelet methods, *Studies in Mathematics and its Applications*, vol. 32. North-Holland Publishing Co., Amsterdam (2003)
- [18] Cohen, A., Daubechies, I., Feauveau, J.C.: Biorthogonal bases of compactly supported wavelets. *Comm. Pure Appl. Math.* **45**, 485–560 (1992)
- [19] Coquel, F., Maday, Y., Müller, S., Postel, M., Tran, Q.: New trends in multiresolution and adaptive methods for convection-dominated problems. *ESAIM, Proc.* **29**, 1–7 (2009)
- [20] Dahmen, W., Gottschlich–Müller, B., Müller, S.: Multiresolution schemes for conservation laws. *Numer. Math.* **88**(3), 399–443 (2000)
- [21] Dedner, A., Makridakis, C., Ohlberger, M.: Error control for a class of Runge Kutta discontinuous Galerkin methods for nonlinear conservation laws. *SIAM J. Numer. Anal.* **45**, 514–538 (2007)
- [22] Godlewski, E., Raviart, P.A.: Hyperbolic systems of conservation laws, *Mathématiques & Applications (Paris) [Mathematics and Applications]*, vol. 3/4. Ellipses, Paris (1991)
- [23] Gottlieb, S., Shu, C.W., Tadmor, E.: Strong stability preserving high-order time discretization methods. *SIAM Review* **43**(1), 89–112 (2001)
- [24] Gottschlich–Müller, B., Müller, S.: 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, pp. 385–394. Basel: Birkhäuser (1999)

- [25] Guermond, J., Pasquetti, R., Popov, B.: Entropy viscosity method for nonlinear conservation laws. *J. Comput. Phys.* **230**, 4248–4267 (2011)
- [26] Harten, A.: Discrete multi-resolution analysis and generalized wavelets. *Applied Numer. Math.* **12**, 153–192 (1993)
- [27] Harten, A.: Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm. Pure Appl. Math.* **48**, 1305–1342 (1995)
- [28] Hartmann, R., Houston, P.: Adaptive discontinuous Galerkin finite element methods for nonlinear hyperbolic conservation laws. *SIAM J. Sci. Comput.* **24**, 979–1004 (2002)
- [29] Hartmann, R., Houston, P.: Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comp. Phys.* **183**, 508–532 (2002)
- [30] Houston, P., Senior, B., Süli, E.: *hp*-discontinuous Galerkin finite element methods for hyperbolic problems: Error analysis and adaptivity. *Int. J. Numer. Methods Fluids* **40**(1-2), 153–169 (2002)
- [31] Hovhannisyan, N., Müller, S., Schäfer, R.: Adaptive multiresolution discontinuous galerkin schemes for conservation laws. *Math. Comp.* (2013). DOI: <http://dx.doi.org/10.1090/S0025-5718-2013-02732-9>
- [32] Iacono, F., May, G., Müller, S., Schäfer, R.: A discontinuous Galerkin discretization with multiwavelet-based grid adaptation for compressible flows. In: 49th AIAA Aerospace Sciences Meeting, 2011-0200 (2011)
- [33] Kaibara, M., Gomes, S.: A fully adaptive multiresolution scheme for shock computations. In: Godunov methods. Theory and applications. International conference, Oxford, GB, October 1999, pp. 497–503. New York, NY: Kluwer Academic/ Plenum Publishers (2001)
- [34] Keinert, F.: Wavelets and multiwavelets. *Studies in Advanced Mathematics*. Chapman & Hall/CRC, Boca Raton, FL (2004)
- [35] Kurganov, A., Petrova, G., Popov, B.: Adaptive semidiscrete central-upwind schemes for nonconvex hyperbolic conservation laws. *SIAM J. Sci. Comput.* **29**(6), 1064–8275 (2007)
- [36] Mallat, S.G.: A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. Pattern Anal. Machine Intell.* **11**(7), 674–693 (1989)
- [37] Müller, S.: Adaptive multiscale schemes for conservation laws, *Lecture Notes in Computational Science and Engineering*, vol. 27. Springer-Verlag, Berlin (2003)
- [38] Müller, S.: 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)

- [39] Müller, T.: Construction of genuinely multi-dimensional multiwavelets (2013). Bachelor thesis, RWTH Aachen University
- [40] Reed, W.H., Hill, T.R.: Triangular mesh methods for the neutron transport equation. Los Alamos Scientific Laboratory Report (LA-UR-73-479) (1973)
- [41] Remacle, J.F., Flaherty, J., Shephard, M.: An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems. *SIAM Review* **45**(1), 53–72 (2003)
- [42] Roussel, O., Schneider, K.: A fully adaptive multiresolution scheme for 3D reaction-diffusion equations. In: Finite volumes for complex applications III. Problems and perspectives. Papers from the 3rd symposium of finite volumes for complex applications, Porquerolles, France, June 24–28, 2002., pp. 833–840. London: Hermes Penton Science (2002)
- [43] Schäfer, R.: Adaptive multiresolution discontinuous Galerkin schemes for conservation laws. Ph.D. thesis, RWTH Aachen University (2011)
- [44] Shelton, A.B.: A multi-resolution discontinuous Galerkin method for unsteady compressible flows. Ph.D. thesis, Georgia Institute of Technology (2008)
- [45] Strela, V.: Multiwavelets: Theory and applications. Ph.D. thesis, Massachusetts Institute of Technology (1996)
- [46] Wang, L., Mavriplis, D.: Adjoint-based *hp* adaptive discontinuous Galerkin methods for the 2d compressible Euler equations. *J. Comp. Phys.* **228**(20), 7643–7661 (2009)
- [47] Yu, T., Kolarov, K., Lynch, W.: Barysymmetric multiwavelets on triangles. IRC Report 1997-006, Stanford University (1997)

## Appendix: Multiwavelets on the reference element $[0, 1]^2$

In this section we give some examples of multiwavelets which are constructed by Algorithm 1 and used for the computations in Section 5. For a detailed description of the construction we refer to [39] To simplify their representation

we use the following initialization in step (i) of Algorithm 1:

$$\begin{aligned}\psi_{\lambda,i,0}(\mathbf{x}) &:= \begin{cases} (\mathbf{x} - \mathbf{x}_c)^i & \text{if } \mathbf{x} \in (0.5, 1.0)^2 \\ -(\mathbf{x} - \mathbf{x}_c)^i & \text{if } \mathbf{x} \in (0, 0.5)^2 \\ 0 & \text{else} \end{cases}, \\ \psi_{\lambda,i,1}(\mathbf{x}) &:= \begin{cases} (\mathbf{x} - \mathbf{x}_c)^i & \text{if } \mathbf{x} \in (0.5, 1) \times (0, 0.5) \\ -(\mathbf{x} - \mathbf{x}_c)^i & \text{if } \mathbf{x} \in (0, 0.5) \times (0.5, 1) \\ 0 & \text{else} \end{cases}, \\ \psi_{\lambda,i,2}(\mathbf{x}) &:= \begin{cases} (\mathbf{x} - \mathbf{x}_c)^i & \text{if } \mathbf{x} \in (0, 0.5)^2 \cup (0.5, 1)^2 \\ -(\mathbf{x} - \mathbf{x}_c)^i & \text{if } \mathbf{x} \in (0.5, 1) \times (0, 0.5) \cup (0, 0.5) \times (0.5, 1) \\ 0 & \text{else} \end{cases},\end{aligned}$$

with  $\mathbf{x}_c = (0.5, 0.5)^t$ .

Since the construction is done on a reference element, the uniform boundedness of the multiwavelets in (3.8) is satisfied for any scaling of the multiwavelets on the reference element. The multiwavelets we use in our computations in Section 5 are normalized with respect to  $L_2((0, 1)^2)$  on the reference element and then shifted to the local cells in the grid.

In Table 3 and Table 4 we list all multiwavelets for  $p = 1, 2, 3$ . The multiwavelets are shown here without normalization to keep them clearly arranged. For simplicity we name the multiwavelets on the reference element  $V_\lambda = (0, 1)^2$  by  $\hat{\psi}_i$ . The multiwavelets are enumerated from 1 to  $3|\mathcal{P}|$ . Then a multiwavelet is uniquely defined by four polynomials:

$$\hat{\psi}_i(x, y) := \begin{cases} \hat{\psi}_{i,1}(x, y) & \text{if } (x, y) \in [0.5, 1] \times [0.5, 1], \\ \hat{\psi}_{i,2}(x, y) & \text{if } (x, y) \in (0.5, 1) \times (0, 0.5), \\ \hat{\psi}_{i,3}(x, y) & \text{if } (x, y) \in (0, 0.5) \times (0, 0.5), \\ \hat{\psi}_{i,4}(x, y) & \text{if } (x, y) \in (0, 0.5) \times (0.5, 1), \\ 0 & \text{else.} \end{cases}$$

|     |     | $p = 1$            | $p = 2$            | $p = 3$                                   |
|-----|-----|--------------------|--------------------|---|
| $i$ | $j$ | $\hat{\psi}_{i,j}$ | $\hat{\psi}_{i,j}$ | $\hat{\psi}_{i,j}$                        |
| 1   | 1   | 1                  | $19 - 12(x + y)$   | $115 - 150(x + y) + 60y^2 + 72xy + 60x^2$ |
|     | 2   | 0                  | 0                  | 0   |
|     | 3   | -1                 | $5 - 12(x + y)$    | $-7 + 42(x + y) - 60y^2 - 72xy - 60x^2$   |
|     | 4   | 0                  | 0                  | 0   |
| 2   | 1   | 0                  | $4y - 3$           | $220y - 95 - 120y^2 - 48xy + 36x$         |
|     | 2   | 1                  | 0                  | 0   |
|     | 3   | 0                  | $-4y + 1$          | $68y - 7 - 120y^2 - 48xy + 12x$           |
|     | 4   | -1                 | 0                  | 0   |
| 3   | 1   | 1                  | $4x - 3$           | $42y^2 - 53y + 9 + 25x - 12xy - 10x^2$    |
|     | 2   | -1                 | 0                  | 0   |
|     | 3   | 1                  | $-4x + 1$          | $-42y^2 + 19y - 1 - x + 12xy + 10x^2$     |
|     | 4   | -1                 | 0                  | 0   |
| 4   | 1   |                    | 0                  | $509x - 190 - 96xy + 3y - 285x^2 + 45y^2$ |
|     | 2   |                    | $7 + 12y - 12x$    | 0   |
|     | 3   |                    | 0                  | $157x - 14 - 96xy + 3y - 285x^2 + 45y^2$  |
|     | 4   |                    | $-7 + 12y - 12x$   | 0   |
| 5   | 1   |                    | 0                  | $16xy + 4x - 11y + 2 - 10x^2$             |
|     | 2   |                    | $4y - 1$           | 0   |
|     | 3   |                    | 0                  | $-16xy + 5y - 1 + 10x^2$                  |
|     | 4   |                    | $-4y + 3$          | 0   |
| 6   | 1   |                    | 0                  | $14x^2 - 20x + 6 + 1y$                    |
|     | 2   |                    | $4x - 3$           | 0   |
|     | 3   |                    | 0                  | $-14x^2 + 8x - 1 + 1y$                    |
|     | 4   |                    | $-4x + 1$          | 0   |
| 7   | 1   |                    | 1                  | 0   |
|     | 2   |                    | -1                 | $25 + 30y - 78x + 60(x^2 + y^2) - 72xy$   |
|     | 3   |                    | 1                  | 0   |
|     | 4   |                    | -1                 | $-25 + 78y - 30x - 60(x^2 + y^2) + 72xy$  |
| 8   | 1   |                    | $8y - 1 - 6x$      | 0   |
|     | 2   |                    | $-8y + 7 - 6x$     | $-20y - 5 + 120y^2 - 48xy + 12x$          |
|     | 3   |                    | $8y - 1 - 6x$      | 0   |
|     | 4   |                    | $-8y + 7 - 6x$     | $-172y + 59 + 120y^2 - 48xy + 36x$        |
| 9   | 1   |                    | $8x - 1 - 6y$      | 0   |
|     | 2   |                    | $-8x + 7 - 6y$     | $42y^2 - 31y - 2 + 13x + 12xy - 10x^2$    |
|     | 3   |                    | $8x - 1 - 6y$      | 0   |
|     | 4   |                    | $-8x + 7 - 6y$     | $-42y^2 + 65y - 24 + 5x - 12xy + 10x^2$   |

Table 3: Wavelets for  $p = 1, 2, 3$  - part I



|     |     | $p = 1$            | $p = 2$            | $p = 3$                                      |
|-----|-----|--------------------|--------------------|--|
| $i$ | $j$ | $\hat{\psi}_{i,j}$ | $\hat{\psi}_{i,j}$ | $\hat{\psi}_{i,j}$                           |
| 10  | 1   |                    |                    | 0  |
|     | 2   |                    |                    | $413x - 142 + 96xy - 93y - 285x^2 + 45y^2$   |
|     | 3   |                    |                    | 0  |
|     | 4   |                    |                    | $61x + 34 + 96xy - 93y - 285x^2 + 45y^2$     |
| 11  | 1   |                    |                    | 0  |
|     | 2   |                    |                    | $96xy - 120x - 66y + 54 + 60x^2$             |
|     | 3   |                    |                    | 0  |
|     | 4   |                    |                    | $-96xy + 96x + 30y - 24 - 60x^2$             |
| 12  | 1   |                    |                    | 0  |
|     | 2   |                    |                    | $28x^2 - 40x + 14 - 2y$                      |
|     | 3   |                    |                    | 0  |
|     | 4   |                    |                    | $-28x^2 + 16x - 2y$                          |
| 13  | 1   |                    |                    | $-35 + 288xy + 156(x + y) - 300(x^2 + y^2)$  |
|     | 2   |                    |                    | $179 + 288xy - 444(x + y) + 300(x^2 + y^2)$  |
|     | 3   |                    |                    | $-35 + 288xy + 156(x + y) - 300(x^2 + y^2)$  |
|     | 4   |                    |                    | $179 + 288xy - 444(x + y) + 300(x^2 + y^2)$  |
| 14  | 1   |                    |                    | $8y - 1 - 6x$                                |
|     | 2   |                    |                    | $-8y + 7 - 6x$                               |
|     | 3   |                    |                    | $8y - 1 - 6x$                                |
|     | 4   |                    |                    | $-8y + 7 - 6x$                               |
| 15  | 1   |                    |                    | $59x + 171y^2 - 27y - 8 - 288xy + 85x^2$     |
|     | 2   |                    |                    | $229x - 171y^2 + 315y - 136 - 288xy - 85x^2$ |
|     | 3   |                    |                    | $59x + 171y^2 - 27y - 8 - 288xy + 85x^2$     |
|     | 4   |                    |                    | $229x - 171y^2 + 315y - 136 - 288xy - 85x^2$ |
| 16  | 1   |                    |                    | $8x - 1 - 6y$                                |
|     | 2   |                    |                    | $-8x + 7 - 6y$                               |
|     | 3   |                    |                    | $8x - 1 - 6y$                                |
|     | 4   |                    |                    | $-8x + 7 - 6y$                               |
| 17  | 1   |                    |                    | $32xy - 1(x + y + 1) - 15(x^2 + y^2)$        |
|     | 2   |                    |                    | $-32xy + 31(x + y) - 15(1 + x^2 + y^2)$      |
|     | 3   |                    |                    | $32xy - 1(x + y + 1) - 15(x^2 + y^2)$        |
|     | 4   |                    |                    | $-32xy + 31(x + y) - 15(1 + x^2 + y^2)$      |
| 18  | 1   |                    |                    | $32x^2 - 14x - 1 - 36xy + 18y$               |
|     | 2   |                    |                    | $-32x^2 + 50x - 17 - 36xy + 18y$             |
|     | 3   |                    |                    | $32x^2 - 14x - 1 - 36xy + 18y$               |
|     | 4   |                    |                    | $-32x^2 + 50x - 17 - 36xy + 18y$             |

Table 4: Wavelets for  $p = 1, 2, 3$  - part II