Limiting discontinuous Galerkin solutions using multiwavelets

Thea Vuik
Literature review

March 12, 2012

Contents

Introduction

This research project concentrates on constructing a multiwavelet limiter. This is based on the need for a reliable limiting technique for hyperbolic partial differential equations. Many areas such as climate modeling, shallow water equations, and computational fluid dynamics use hyperbolic partial differential equations (PDE's) to describe the behavior of some unknown quantity. To solve PDE's, various types of numerical methods can be used, such as finite difference, finite volume and finite element methods. In this literature review the discontinuous Galerkin method (DG) is used to solve the linear advection equation in one (Section 1) or two dimensions (Section 2).
Hyperbolic equations often go hand in hand with discontinuities or shocks. To efficiently apply DG in this case, limiting techniques are used to reduce the spurious oscillations that develop near discontinuities. Some properties of limiters are described in Section 4. Besides that, some examples of existing limiters are given, such as the minmod limiter, projection limiter, moment limiter, WENO limiter and the multiwavelet limiter. Unfortunately, these limiters do not work well for higher order approximations, or multidimensional cases.
As this project focuses on using another type of multiwavelet limiting, necessary information about multiwavelets is given in Section 3.
The project that follows this literature study will explore the possibility of using multiwavelets as a limiter, both to detect discontinuities and to reduce oscillations in the neighbourhood of a shock, while maintaining high accuracy away from a shock. A multidimensional case will also be considered, along with typical one-dimensional test cases.

The following research questions are the motivation for this research project:

- How might multiwavelets help to overcome the current problems (limiters give low order, and are only useful in 1D), and what is the difference between multiwavelet limiting and the existing limiters?
- Is there a relation between the degree of the multiwavelet basis and the number of levels which are needed to approximate a function?
- How can multiwavelets be used in discontinuous functions? Is it possible to detect discontinuities using multiwavelets?

In the course of this MSc thesis, these questions will be addressed and hopefully answered. The purpose of this literature study is to provide the background necessary to proceed in answering these questions.

1 Linear advection equation in one dimension

In this section the linear advection equation is considered on the domain $[-1,1]$. This is given by

$$u_t + u_x = 0, \quad x \in [-1,1], t \geq 0, \tag{1}$$

$$u(x,0) = u^0(x), \quad x \in [-1,1], \tag{2}$$

where $u = u(x,t)$ and periodic boundary conditions are assumed.

1.1 The discontinuous Galerkin space discretization

1.1.1 The weak formulation

To discretize in space a mesh is defined. Let N be such that the number of elements in $[-1,1]$ is $N+1$ and define $\Delta x = \frac{2}{N+1}$. The cell centers are given by

$$x_j = -1 + (j + \frac{1}{2})\Delta x, \quad j = 0, \ldots, N.$$

The elements, I_j, in $[-1,1]$ are defined as

$$I_j = (x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}) = (x_j - \frac{\Delta x}{2}, x_j + \frac{\Delta x}{2}), \quad j = 0, \ldots, N.$$

To construct the DG method, the approximation space is defined by

$$V_h = \{v : v \in \Phi^\ell(I_j), \ell = 0, \ldots, k, j = 0, \ldots, N\},$$

where $\Phi^\ell(I_j)$ denotes the space of scaled Legendre polynomials on I_j of degree $\ell \leq k$. For degree 0 to 4, the scaled Legendre polynomials on I_j are given by

$$\phi_0(\xi) = \sqrt{\frac{1}{2}}, \tag{3}$$

$$\phi_1(\xi) = \sqrt{\frac{3}{2}}\xi, \tag{4}$$

$$\phi_2(\xi) = \frac{1}{2}\sqrt{\frac{5}{2}}(3\xi^2 - 1), \tag{5}$$

$$\phi_3(\xi) = \frac{1}{2}\sqrt{\frac{7}{2}}(5\xi^3 - 3\xi), \tag{6}$$

$$\phi_4(\xi) = \frac{1}{8}\sqrt{\frac{9}{2}}(35\xi^4 - 30\xi^2 + 3), \tag{7}$$

where $\xi = \frac{2}{\Delta x}(x - x_j)$, such that $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ (global coordinates) transforms into $[-1,1]$ (local coordinates). Functions $\phi_0, \ldots, \phi_4$ form an orthonormal basis for Φ^4, such that

$$\langle \phi_\ell, \phi_m \rangle = \int_{-1}^{1} \phi_\ell(\xi)\phi_m(\xi)d\xi = \delta_{\ell m}. \tag{8}$$

The (unscaled) Legendre polynomials are given by $P^{(m)}(\xi) = \sqrt{\frac{2}{2m+1}}\phi_m(\xi)$.

Now that a mesh and approximating basis have been defined, an approximation $u_h(x,t)$ is found, such that for each time t, $u_h \in V_h$ and $u_h \approx u$.
For this, a weak formulation is used, as described in Cockburn [7]. Equation (1) is multiplied by an arbitrary, smooth function $v \in V_h(I_j)$ and integrated over I_j, $j \in \{0, \ldots, N\}$. Using integration by parts, this gives

$$\int_{I_j} (u_t + u_x) v dx = 0,$$
$$\int_{I_j} u_t v dx + uv|_{I_j} - \int_{I_j} u v_x dx = 0,$$

such that u should satisfy

$$\int_{I_j} u_t v dx - \int_{I_j} u v_x dx + u(x_{j+\frac{1}{2}}) v(x_{j+\frac{1}{2}}) - u(x_{j-\frac{1}{2}}) v(x_{j-\frac{1}{2}}) = 0. \tag{9}$$

Next, v is replaced by a test function $v_h \in V_h$, and the exact solution u by the approximate solution $u_h \in V_h$. Using local coordinates $\xi = \frac{2}{\Delta x}(x - x_j)$,

$$v_h(x) = \phi_m(\xi), \qquad m \in \{0, \ldots, k\}, \tag{10}$$
$$u_h(x,t) = \sum_{\ell=0}^{k} u_j^{(\ell)}(t)\phi_\ell(\xi), \text{ on element } I_j, \tag{11}$$

where $u_j^{(\ell)}(t), \ell = 0, \ldots, k$, are the unknown DG coefficients. Using equations (10) and (11) in the weak formulation as given in (9), gives

$$\begin{aligned} &\int_{I_j} \left(\sum_{\ell=0}^{k} \frac{du_j^{(\ell)}}{dt} \phi_\ell \left(\frac{2}{\Delta x}(x - x_j) \right) \right) \phi_m \left(\frac{2}{\Delta x}(x - x_j) \right) dx \; + \\ -&\int_{I_j} \left(\sum_{\ell=0}^{k} u_j^{(\ell)} \phi_\ell \left(\frac{2}{\Delta x}(x - x_j) \right) \right) \frac{d}{dx} \phi_m \left(\frac{2}{\Delta x}(x - x_j) \right) dx \; + \\ &+ \; \hat{u}_h(x_{j+\frac{1}{2}}) v_h(x^-_{j+\frac{1}{2}}) - \hat{u}_h(x_{j-\frac{1}{2}}) v_h(x^+_{j-\frac{1}{2}}) = 0. \end{aligned} \tag{12}$$

for every $m \in \{0, \ldots, k\}$. More information about the boundary terms in equation (12) is given in Section 1.1.2.
Next, a change of coordinates is performed, such that $\xi = \frac{2}{\Delta x}(x - x_j)$. It follows that $\frac{\Delta x}{2} d\xi = dx$ and $\frac{d}{d\xi}\frac{d\xi}{dx} = \frac{2}{\Delta x}\frac{d}{d\xi}$.
Using this change of coordinates to rewrite equation (12) gives

$$\sum_{\ell=0}^{k} \frac{du_j^{(\ell)}}{dt} \frac{\Delta x}{2} \int_{-1}^{1} \phi_\ell(\xi)\phi_m(\xi) d\xi - \sum_{\ell=0}^{k} u_j^{(\ell)} \int_{-1}^{1} \phi_\ell(\xi) \frac{d}{d\xi} \phi_m(\xi) d\xi$$

$$+\ \hat{u}_h(x_{j+\frac{1}{2}})v_h(x^-_{j+\frac{1}{2}}) - \hat{u}_h(x_{j-\frac{1}{2}})v_h(x^+_{j-\frac{1}{2}}) = 0. \tag{13}$$

Another useful relation is given by,

$$\int_{-1}^{1} \phi_\ell(\xi)\frac{d}{d\xi}\phi_m(\xi)d\xi = \begin{cases} 2\sqrt{(\ell+\frac{1}{2})(m+\frac{1}{2})} & \text{if } m > \ell \text{ and } m+\ell \text{ is odd,} \\ 0 & \text{else.} \end{cases}$$

Simplifying the first term in equation (13) gives

$$\sum_{\ell=0}^{k} \frac{du_j^{(\ell)}}{dt}\frac{\Delta x}{2}\int_{-1}^{1}\phi_\ell(\xi)\phi_m(\xi)d\xi = \frac{\Delta x}{2}\frac{du_j^{(m)}}{dt}. \tag{14}$$

1.1.2 Numerical fluxes

In the weak form given in equation (13), the fluxes, $\hat{u}_h$, at the left and right boundary of element I_j appear. These are, in general, unknown. However, in this special case (the linear advection equation) the exact solution of equation (1) is known: the initial condition given in (2) is advected with constant speed along the characteristics $x-t=c$, with $c \in \mathbb{R}$, see Leveque [15]. Therefore, the flux can be approximated using an upwind flux.
At $x_{j+\frac{1}{2}}$ this approximation is given by using equation (11) at $\xi = 1$ on element I_j, and at $x_{j-\frac{1}{2}}$ by using this equation at $\xi = 1$ on element I_{j-1}. Such that

$$\begin{aligned} \hat{u}_h(x_{j+\frac{1}{2}}) &\approx (u_h)^-_{j+\frac{1}{2}} &&= \textstyle\sum_{\ell=0}^{k} u_j^{(\ell)}\phi_\ell(1), \\ \hat{u}_h(x_{j-\frac{1}{2}}) &\approx (u_h)^-_{j-\frac{1}{2}} &&= \textstyle\sum_{\ell=0}^{k} u_{j-1}^{(\ell)}\phi_\ell(1), \end{aligned}$$

where the minus sign refers to the left side of the boundaries $x_{j-\frac{1}{2}}$ and $x_{j+\frac{1}{2}}$, as can be seen in Figure 1.

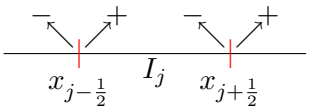


Figure 1: Boundaries of interval I_j, used for flux terms

The test function v is taken from inside the cell, given by $v^+_{j-\frac{1}{2}}$ and $v^-_{j+\frac{1}{2}}$.
The scaled Legendre polynomials are defined such that

$$\phi_m(1) = \sqrt{m+\frac{1}{2}} \text{ and } \phi_m(-1) = (-1)^m\sqrt{m+\frac{1}{2}}, \quad m = 0, 1, \ldots, k.$$

The boundary terms in equation (13) are replaced by

$$(u_h)^-_{j+\frac{1}{2}} v^-_{j+\frac{1}{2}} = \left(\sum_{\ell=0}^{k} u_j^{(\ell)} \phi_\ell(1) \right) \phi_m(1) \quad = \sqrt{m+\frac{1}{2}} \sum_{\ell=0}^{k} \sqrt{\ell+\frac{1}{2}} u_j^{(\ell)}, \tag{15}$$

$$(u_h)^-_{j-\frac{1}{2}} v^+_{j-\frac{1}{2}} = \left(\sum_{\ell=0}^{k} u_{j-1}^{(\ell)} \phi_\ell(1) \right) \phi_m(-1) = (-1)^m \sqrt{m+\frac{1}{2}} \left(\sum_{\ell=0}^{k} \sqrt{\ell+\frac{1}{2}} u_{j-1}^{(\ell)} \right). \tag{16}$$

1.1.3 Matrix-vector notation

Using equations (14), (15) and (16) in equation (13) gives the following differential equation:

$$\begin{aligned} \tfrac{\Delta x}{2} \tfrac{du_j^{(m)}}{dt} &= \sum_{\ell=0}^{k} u_j^{(\ell)} \textstyle\int_{-1}^{1} \phi_\ell(\xi) \tfrac{d}{d\xi} \phi_m(\xi) d\xi + \\ &- \sqrt{m+\tfrac{1}{2}} \left(\sum_{\ell=0}^{k} \sqrt{\ell+\tfrac{1}{2}} u_j^{(\ell)} \right) + (-1)^m \sqrt{m+\tfrac{1}{2}} \left(\sum_{\ell=0}^{k} \sqrt{\ell+\tfrac{1}{2}} u_{j-1}^{(\ell)} \right); \end{aligned}$$

where $u_j^{(m)}(t), j = 0, \ldots, N, m = 0, \ldots, k$ are unknown coefficients. Written in matrix-vector form, this looks like

$$M \frac{d}{dt} \mathbf{u}_j = S_1 \mathbf{u}_j + S_2 \mathbf{u}_{j-1}, j = 1, \ldots, N, \tag{17}$$

where

$$\begin{aligned} \mathbf{u}_j &= [u_j^{(0)}(t) \;\; u_j^{(1)}(t) \;\; \ldots \;\; u_j^{(k)}(t)]^\top, \\ M &= \operatorname{diag}\left(\tfrac{\Delta x}{2}\right), \\ S_1(m+1, \ell+1) &= -\sqrt{(m+\tfrac{1}{2})(\ell+\tfrac{1}{2})} + \textstyle\int_{-1}^{1} \phi_\ell(\xi) \tfrac{d}{d\xi} \phi_m(\xi) d\xi, \\ S_2(m+1, \ell+1) &= (-1)^m \sqrt{(m+\tfrac{1}{2})(\ell+\tfrac{1}{2})}. \end{aligned}$$

The three matrices all have the same size of $(k+1) \times (k+1)$.
To compute the solution for $j = 0$, a periodic boundary condition is used. Therefore, a virtual element I_{-1} is included, where $u_{-1}^{(\ell)}(t) = u_N^{(\ell)}(t), \ell = 0, \ldots, k$. This results in the following equation,

$$M \frac{d}{dt} \mathbf{u}_0 = S_1 \mathbf{u}_0 + S_2 \mathbf{u}_N. \tag{18}$$

After solving this system of equations (17) and (18) for $\mathbf{u}_j, j = 0, \ldots, N$, the DG solution is then formed by summing the product of the coefficients with the basis functions

$$u_h(x,t) = \sum_{\ell=0}^{k} u_j^{(\ell)}(t) \phi_\ell(\xi), x \in I_j, j = 0, \ldots, N.$$

1.2 Time stepping using TVD Runge Kutta

Equation (17) describes a system of differential equations in time. The time stepping part is done using a total variation diminishing (TVD) Runge Kutta. The third order TVD RK method has been developed by Gottlieb [11] and is also used by Qiu [16]. If equation (17) is written as $\frac{d}{dt}\mathbf{u}_j = L(\mathbf{u}_{j-1}, \mathbf{u}_j)$, $j = 0, \ldots, N$, and the approximation at time $t^n = n\Delta t, n \geq 0$, for element j is given by $\mathbf{w}_j^n$, then $\mathbf{w}_j^{n+1}$ at time t_{n+1} is computed by

$$\begin{aligned} \mathbf{w}_j^{(1)} &= \mathbf{w}_j^n + \Delta t L(\mathbf{w}_{j-1}^n, \mathbf{w}_j^n), \\ \mathbf{w}_j^{(2)} &= \tfrac{3}{4}\mathbf{w}_j^n + \tfrac{1}{4}\mathbf{w}_j^{(1)} + \tfrac{1}{4}\Delta t L(\mathbf{w}_{j-1}^{(1)}, \mathbf{w}_j^{(1)}), \\ \mathbf{w}_j^{n+1} &= \tfrac{1}{3}\mathbf{w}_j^n + \tfrac{2}{3}\mathbf{w}_j^{(2)} + \tfrac{2}{3}\Delta t L(\mathbf{w}_{j-1}^{(2)}, \mathbf{w}_j^{(2)}). \end{aligned} \tag{19}$$

As an example, the initial condition is given by $u^0(x) = \sin(2\pi x)$, $x \in [-1, 1]$.
Note that the DG coefficients $u_j^{(\ell)}(0), j = 0, \ldots, N, \ell = 0, \ldots, k$, are needed to act as initial conditions in equation (17). Therefore $u^0(x)$ should be projected onto V_h to find $u_h(x, 0)$.

1.2.1 Projection of initial condition

Let a projected initial condition $u_h(x, 0) \in V_h$ on element $I_j, j \in \{0, \ldots, N\}$, be given by

$$u_h(x, 0) = \sum_{\ell=0}^{k} u_j^{(\ell)}(0)\phi_\ell\left(\frac{2}{\Delta x}(x - x_j)\right) \tag{20}$$

(see equation (11)). Multiplying this equation by $\phi_m\left(\frac{2}{\Delta x}(x - x_j)\right)$ and integrating over I_j gives

$$\int_{I_j} u_h(x, 0)\phi_m\left(\frac{2}{\Delta x}(x - x_j)\right) dx =$$

$$\int_{I_j} \sum_{\ell=0}^{k} u_j^{(\ell)}(0)\phi_\ell\left(\frac{2}{\Delta x}(x - x_j)\right)\phi_m\left(\frac{2}{\Delta x}(x - x_j)\right) dx.$$

Using the change of coordinates $\xi = \frac{2}{\Delta x}(x - x_j)$ and the orthonormal property of the scaled Legendre polynomials given in (8) gives

$$\begin{aligned} \frac{\Delta x}{2}\int_{-1}^{1} u_h\left(x_j + \frac{\Delta x}{2}\xi, 0\right)\phi_m(\xi)d\xi &= \frac{\Delta x}{2}\sum_{\ell=0}^{k} u_j^{(\ell)}(0)\int_{-1}^{1}\phi_\ell(\xi)\phi_m(\xi)d\xi \\ &= \frac{\Delta x}{2}u_j^{(m)}(0). \end{aligned}$$

Therefore the DG coefficients are given by

$$\begin{aligned} u_j^{(m)}(0) &= \int_{-1}^{1} u_h\left(x_j + \frac{\Delta x}{2}\xi, 0\right)\phi_m(\xi)d\xi \\ &\approx \int_{-1}^{1} u^0\left(x_j + \frac{\Delta x}{2}\xi\right)\phi_m(\xi)d\xi. \end{aligned} \tag{21}$$

To compute the above stated integral numerically, Gauss-Legendre quadrature using six points[1] is used, see Abramowitz and Stegun, [1]. This rule is stated as

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=0}^{5} v_i f(\hat{x}_i),$$

where $\hat{x}_i, v_i, i = 0, \ldots, 5$ are as stated in Table 1.

i	0	1	2	3	4	5	6
	Six points						
$\hat{x}_i$	-0.93247	-0.66121	-0.23862	0.23862	0.66121	0.93247	-
v_i	0.17132	0.36076	0.46791	0.46791	0.36076	0.17132	-
	Seven points						
$\hat{y}_i$	-0.94911	-0.74153	-0.40585	0	0.40585	0.74153	0.94911
w_i	0.12948	0.27971	0.38183	0.41796	0.38183	0.27971	0.12948

Table 1: Points and weights using six and seven point Gauss-Legendre quadrature

The initial condition $u_h(x, 0)$ is evaluated in several points within each interval I_j, using formula (20) for $x \in I_j, j = 0, \ldots, N$.

[1]The associated formula is given at `http://pathfinder.scar.utoronto.ca/~dyer/csca57/book_P/node44.html`

1.2.2 The CFL condition

In order to use a time stepping method, a value for Δt should be chosen such that the CFL number $\nu = \frac{\Delta t}{\Delta x}$ is small enough. In that case, the time stepping errors can be neglected with respect to the spatial errors. If a TVD RK method of order three is used, then the stability condition using Φ^k, for $k = 0, 1, 2$, is

$$\nu \leq \frac{1}{2k+1},$$

Cockburn [7]. In practice, this bound is somewhat too restrictive. If $k \leq 2$ the used stability limit is as given in Table 2, so ν must be less than these numbers (Cockburn [9]). Note that in this case the time stepping error is $\mathcal{O}(\Delta t^3)$ whereas the spatial errors are of order $k+1 \leq 3$. Indeed the spatial errors dominate in this case.

V_h	Φ^0	Φ^1	Φ^2
$\nu_{\max}$	1.256	0.409	0.209

Table 2: Stability limit of using DG and TVD RK to solve $u_t + u_x = 0$

If $k \geq 3$ it is possible that the time stepping errors dominate. If Φ^3 is used it should hold that $\nu \leq 0.13$ and for Φ^4 it should hold that $\nu \leq 0.089$. If another number of elements is used, then the new CFL number $\widetilde{\nu}$ should be adapted such that the spatial errors still dominate. This is achieved by using

$$\widetilde{\nu} = \nu \left(\frac{N+1}{\widetilde{N}+1} \right)^{\frac{k+1}{\mu}}, \tag{22}$$

where ν is the old stability bound, $N+1$ is the number of elements in the old mesh, $\widetilde{N}+1$ is the number of elements on the current mesh, $k+1$ is the order of the DG scheme, and μ is the order of the time-stepping scheme (in this case $\mu = 3$).

If Φ^3 or Φ^4 is used, then the relation between Δt and Δx should be as given in Table 3. The most restrictive value for ν is chosen, such that for $k = 3$ it should hold that $\frac{\Delta t}{\Delta x} \leq 0.003 \leq 0.13 \cdot \left(\frac{1}{2}\right)^{\frac{16}{3}} \approx 0.0032$.
If $k = 4$ this gives $\frac{\Delta t}{\Delta x} \leq 8 \cdot 10^{-5} \leq 0.089 \cdot \left(\frac{1}{2}\right)^{\frac{20}{3}} \approx 8.7604 \cdot 10^{-5}$.
Note that the chosen value for ν depends on which mesh is started with. Relation (22) is an empirical formula which can not be proven thoroughly.

	Φ^3				
$N+1$	10	20	40	80	160
ν	0.13	$0.13 \cdot \left(\frac{10}{20}\right)^{\frac{4}{3}}$	$0.13 \cdot \left(\frac{1}{2}\right)^{\frac{8}{3}}$	$0.13 \cdot \left(\frac{1}{2}\right)^{\frac{12}{3}}$	$0.13 \cdot \left(\frac{1}{2}\right)^{\frac{16}{3}}$
	Φ^4				
$N+1$	10	20	40	80	160
ν	0.089	$0.089 \cdot \left(\frac{10}{20}\right)^{\frac{5}{3}}$	$0.089 \cdot \left(\frac{1}{2}\right)^{\frac{10}{3}}$	$0.089 \cdot \left(\frac{1}{2}\right)^{\frac{15}{3}}$	$0.089 \cdot \left(\frac{1}{2}\right)^{\frac{20}{3}}$

Table 3: Chosen CFL number

1.2.3 Accuracy of the numerical results

In this section two norms of the error at time T are considered: $||e(T)||_{L^2(-1,1)}$ and $||e(T)||_{L^\infty(-1,1)}$. The error $||e(T)||_{L^2(-1,1)}$ is given by

$$\begin{aligned} ||e(T)||^2_{L^2(-1,1)} &= \int_{-1}^{1} \left(u(x,T) - u_h(x,T)\right)^2 dx \\ &= \sum_{j=0}^{N} \int_{I_j} \left(u(x,T) - u_h(x,T)\right)^2 dx \\ &= \sum_{j=0}^{N} ||e(T)||^2_{L^2(x_{j-1/2},x_{j+1/2})}. \end{aligned}$$

For each $j \in \{0, 1, \ldots, N\}$ a seven point Gauss-Legendre quadrature is used to determine

$$||e(T)||^2_{L^2(x_{j-1/2},x_{j+1/2})} = \int_{x_{j-1/2}}^{x_{j+1/2}} \left(u(x,T) - u_h(x,T)\right)^2 dx. \tag{23}$$

To do this, a coordinate transformation[2] is needed such that $\int_{x_{j-1/2}}^{x_{j+1/2}} dx$ changes to $\int_{-1}^{1} dx$. If y is such that

$$y = -1 + \frac{2}{\Delta x}(x - x_{j-1/2}),$$

with $dx = \frac{\Delta x}{2} dy$ and $x = x_{j-1/2} + \frac{\Delta x}{2}(y+1)$ then equation (23) transforms into

$$\frac{\Delta x}{2} \int_{-1}^{1} \left(u\left(x_{j-1/2} + \frac{\Delta x}{2}(y+1), T\right) - u_h\left(x_{j-1/2} + \frac{\Delta x}{2}(y+1), T\right)\right)^2 dy,$$

which equals

$$\frac{\Delta x}{2} \int_{-1}^{1} \left(u\left(x_j + \frac{\Delta x}{2} y, T\right) - u_h\left(x_j + \frac{\Delta x}{2} y, T\right)\right)^2 dy.$$

Using the seven point Gauss-Legendre quadrature with points and weights as given in Table 1 results in

$$||e(T)||^2_{L^2(x_{j-1/2},x_{j+1/2})} \approx \frac{\Delta x}{2} \sum_{i=0}^{6} w_i \left(u\left(x_j + \frac{\Delta x}{2}\hat{y}_i, T\right) - u_h\left(x_j + \frac{\Delta x}{2}\hat{y}_i, T\right)\right)^2,$$

such that

$$||e(T)||^2_{L^2(-1,1)} \approx \frac{\Delta x}{2} \sum_{j=0}^{N} \left\{ \sum_{i=0}^{6} w_i \left(u\left(x_j + \frac{\Delta x}{2}\hat{y}_i, T\right) - u_h\left(x_j + \frac{\Delta x}{2}\hat{y}_i, T\right)\right)^2 \right\}.$$

[2]This idea can be found at `http://www.caam.rice.edu/~caam453/lecture26.pdf`

It is also possible to use the pointwise error norm, which is defined as

$$||e(T)||_\infty = \max\left\{|u(x,T) - u_h(x,T)|, x \in [-1,1]\right\}.$$

Following Theorem 2.2 in Cockburn [7], a regular initial condition (e.g. $u^0(x) = \sin(2\pi x)$) using approximation space Φ^k should give an approximation error $||e(T)|| = \mathcal{O}(\Delta x)^{k+1}$ if the correct CFL number is chosen. To verify this relation two different values of N are used. Note that if

$$\begin{aligned} ||e(T)|| &= C(\Delta x)^{k+1} = C\left(\tfrac{2}{N+1}\right)^{k+1}, \\ ||\widetilde{e}(T)|| &= C(\widetilde{\Delta x})^{k+1} = C\left(\tfrac{2}{\widetilde{N}+1}\right)^{k+1}, \end{aligned}$$

with $N, \widetilde{N} \in \mathbb{N}, C \in \mathbb{R}$, it should hold that

$$\frac{||e(T)||}{||\widetilde{e}(T)||} = \left(\frac{\widetilde{N}+1}{N+1}\right)^{k+1}.$$

Taking the logarithm of both sides gives

$$\log\left(\frac{||e(T)||}{||\widetilde{e}(T)||}\right) = (k+1)\log\left(\frac{\widetilde{N}+1}{N+1}\right),$$

such that the approximation is good enough if

$$\text{order} = \frac{\log\left(\frac{||e(T)||}{||\widetilde{e}(T)||}\right)}{\log\left(\frac{\widetilde{N}+1}{N+1}\right)} \geq k+1. \tag{24}$$

In Table 4 the errors are given for $T = 0.5$, for different values of $N+1, k$ and ν. The order of accuracy can also be seen: row j gives the order of the approximation using the mesh of row $j-1$ and row j. Note that the order equals $k+1$.
The L^2-norm averages the error and is therefore more regular than the pointwise norm. Due to the limitations of the processing chip, using polynomial basis Φ^5 or higher causes a dominating roundoff error. The order of the error is not equal to $k+1$ in that case.

In the case that the CFL number is not adapted for $k = 3, 4$, for the mesh, the order of the error is not equal to $k+1$, but equals approximately 3.5 if $k = 3$ and 3 if $k = 4$: the error is dominated by time stepping and not by the spatial error as wanted.

$N+1$	$\|\|e(0.5)\|\|_2$	order	$\|\|e(0.5)\|\|_\infty$	order
	$\Phi^0, \nu = 1$			
10	0.9282	-	1.0107	-
20	0.6548	0.5034	0.7040	0.5217
40	0.4003	0.7098	0.4221	0.7380
80	0.2234	0.8418	0.2326	0.8596
160	0.1183	0.9174	0.1223	0.9271
	$\Phi^1, \nu = 0.4$			
10	0.1277	-	0.1659	-
20	0.0257	2.3133	0.0435	1.9329
40	0.0062	2.0531	0.0130	1.7420
80	0.0015	2.0348	0.0035	1.8997
160	3.7599e-04	2.0065	8.9903e-04	1.9532
	$\Phi^2, \nu = 0.2$			
10	0.0101	-	0.0244	-
20	0.0012	3.0223	0.0030	3.0146
40	1.5498e-04	3.0063	3.9854e-04	2.9236
80	1.9350e-05	3.0017	5.0421e-05	2.9826
160	2.4181e-06	3.0004	6.3179e-06	2.9965
	$\Phi^3, \nu = 0.003$			
10	7.5257e-04	-	0.0017	-
20	4.6168e-05	4.0269	1.1501e-04	3.9017
40	2.9211e-06	3.9823	7.1912e-06	3.9993
80	1.8259e-07	3.9998	4.5587e-07	3.9795
160	1.1414e-08	3.9998	2.8542e-08	3.9975
	$\Phi^4, \nu = 8 \cdot 10^{-5}$			
10	4.7937e-05	-	8.5233e-05	-
20	1.4320e-06	5.0650	3.0349e-06	4.8117
40	4.4961e-08	4.9932	9.7045e-08	4.9668
80	1.4110e-09	4.9939	2.9729e-09	5.0287
160	5.0974e-11	4.7908	9.6787e-11	4.9409

Table 4: Errors and orders in 1D, $T = 0.5$

2 Linear advection equation in two dimensions

In two dimensions, the linear advection equation for $u = u(x, y, t)$ is given by

$$\begin{aligned} &u_t + u_x + u_y = 0, && x, y \in [-1, 1], t \geq 0, \\ &u(x, y, 0) = u^0(x, y), && x, y \in [-1, 1], \end{aligned}$$

using periodic boundary conditions.

2.1 The discontinuous Galerkin space discretization

2.1.1 The weak formulation

The space discretization is done by defining N_x and N_y such that $N_x + 1$ elements are found in the $x-$direction and $N_y + 1$ elements in the $y-$direction. The elements are defined as $I_{ij} = (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) \times (y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}), i = 0, \ldots, N_x, j = 0, \ldots, N_y$, following the definition in one dimension, see Section 1.1.1.
The polynomial approximation space is denoted by Φ^k, with basis given by a multiplication of the scaled Legendre polynomials $\phi_\ell, \ell = 0, \ldots, k$ as given in equations (3) to (7). A polynomial approximation of the solution is given by

$$u_h \in V_h = \{v : v \in \Phi^\ell(I_{ij}), \ell = 0, \ldots, k, i = 0, \ldots, N_x, j = 0, \ldots, N_y\},$$

using local coordinates $\xi = \frac{2}{\Delta x}(x - x_i)$ and $\eta = \frac{2}{\Delta y}(y - y_j)$, such that

$$u(x, y, t) \approx u_h(x, y, t) = \sum_{\ell_x=0}^{k} \sum_{\ell_y=0}^{k} u_{ij}^{(\ell_x,\ell_y)}(t)\phi_{\ell_x}(\xi)\phi_{\ell_y}(\eta), \tag{25}$$

which is a direct extension of Section 1.1.1. The weak form of the differential equation can be derived by multiplying by a test function

$$v(x, y) = \phi_{m_x}(\xi)\phi_{m_y}(\eta) \in V_h, \tag{26}$$

where ϕ_{m_x} is the scaled Legendre polynomial of degree $m_x \in \{0, \ldots, k\}$ and ϕ_{m_y} is the one of degree $m_y \in \{0, \ldots, k\}$.
Integrating over element I_{ij} gives, using partial integration and the same notation as in Section 1.1.2:

$$\begin{aligned} &\iint_{I_{ij}} (u_t + u_x + u_y) v dx dy = \\ &\iint_{I_{ij}} u_t v dx dy + \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \left(vu\big|_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} uv_x dx \right) dy \\ &\quad + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(vu\big|_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} uv_y dy \right) dx \\ &\quad \approx \iint_{I_{ij}} u_t v dx dy - \iint_{I_{ij}} u(v_x + v_y) dx dy \\ &\quad + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (v_{i,j+\frac{1}{2}}^- u_{i,j+\frac{1}{2}}^- - v_{i,j-\frac{1}{2}}^+ u_{i,j-\frac{1}{2}}^-) dx \\ &\quad + \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} (v_{i+\frac{1}{2},j}^- u_{i+\frac{1}{2},j}^- - v_{i-\frac{1}{2},j}^+ u_{i-\frac{1}{2},j}^-) dy = 0. \end{aligned}$$

Using definitions (25) and (26) and the fact that (from equations (15) and (16))

$$\begin{aligned} v^-_{i+\frac{1}{2},j}u^-_{i+\frac{1}{2},j} &= \sqrt{m_x+\tfrac{1}{2}}\sum_{\ell_x=0}^{k}\sum_{\ell_y=0}^{k}\sqrt{\ell_x+\tfrac{1}{2}}u_{ij}^{(\ell_x,\ell_y)}(t)\phi_{m_y}(\eta)\phi_{\ell_y}(\eta), \\ v^+_{i-\frac{1}{2},j}u^-_{i-\frac{1}{2},j} &= (-1)^{m_x}\sqrt{m_x+\tfrac{1}{2}}\sum_{\ell_x=0}^{k}\sum_{\ell_y=0}^{k}\sqrt{\ell_x+\tfrac{1}{2}}u_{i-1,j}^{(\ell_x,\ell_y)}(t)\phi_{m_y}(\eta)\phi_{\ell_y}(\eta), \end{aligned}$$

$y\in[y_{j-\frac{1}{2}},y_{j+\frac{1}{2}}]$, and the same structure for $v^-_{i,j+\frac{1}{2}}u^-_{i,j+\frac{1}{2}}$ and $v^+_{i,j-\frac{1}{2}}u^-_{i,j-\frac{1}{2}}$, this differential equation can be rewritten as

$$\begin{aligned} &\sum_{\ell_x=0}^{k}\sum_{\ell_y=0}^{k}\frac{d}{dt}u_{ij}^{(\ell_x,\ell_y)}(t)\iint\limits_{I_{ij}}\left(\phi_{\ell_x}(\xi)\phi_{\ell_y}(\eta)\right)\left(\phi_{m_x}(\xi)\phi_{m_y}(\eta)\right)dxdy = \\ &\sum_{\ell_x=0}^{k}\sum_{\ell_y=0}^{k}u_{ij}^{(\ell_x,\ell_y)}(t)\iint_{I_{ij}}\phi_{\ell_x}(\xi)\phi_{\ell_y}(\eta)\left(\phi_{m_y}(\eta)\frac{d}{dx}\phi_{m_x}(\xi)+\phi_{m_x}(\xi)\frac{d}{dy}\phi_{m_y}(\eta)\right)dxdy \\ &\quad-\sqrt{m_x+\frac{1}{2}}\sum_{\ell_x=0}^{k}\sum_{\ell_y=0}^{k}\sqrt{\ell_x+\frac{1}{2}}u_{ij}^{(\ell_x,\ell_y)}(t)\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\phi_{\ell_y}(\eta)\phi_{m_y}(\eta)dy \\ &\quad+(-1)^{m_x}\sqrt{m_x+\frac{1}{2}}\sum_{\ell_x=0}^{k}\sum_{\ell_y=0}^{k}\sqrt{\ell_x+\frac{1}{2}}u_{i-1,j}^{(\ell_x,\ell_y)}(t)\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}}\phi_{\ell_y}(\eta)\phi_{m_y}(\eta)dy \\ &\quad-\sqrt{m_y+\frac{1}{2}}\sum_{\ell_x=0}^{k}\sum_{\ell_y=0}^{k}\sqrt{\ell_y+\frac{1}{2}}u_{ij}^{(\ell_x,\ell_y)}(t)\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\phi_{\ell_x}(\xi)\phi_{m_x}(\xi)dx \\ &\quad+(-1)^{m_y}\sqrt{m_y+\frac{1}{2}}\sum_{\ell_x=0}^{k}\sum_{\ell_y=0}^{k}\sqrt{\ell_y+\frac{1}{2}}u_{i,j-1}^{(\ell_x,\ell_y)}(t)\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\phi_{\ell_x}(\xi)\phi_{m_x}(\xi)dx. \end{aligned} \tag{27}$$

Using the coordinate transformation to ξ and η and equation (8), equation (27) becomes

$$\begin{aligned} &\frac{\Delta x}{2}\cdot\frac{\Delta y}{2}\cdot\frac{d}{dt}u_{ij}^{(m_x,m_y)}(t) = \\ &\quad=\frac{\Delta y}{2}\sum_{\ell_x=0}^{k}u_{ij}^{(\ell_x,m_y)}(t)\int_{-1}^{1}\phi_{\ell_x}(\xi)\frac{d}{d\xi}\phi_{m_x}(\xi)d\xi \\ &\quad+\frac{\Delta x}{2}\sum_{\ell_y=0}^{k}u_{ij}^{(m_x,\ell_y)}(t)\int_{-1}^{1}\phi_{\ell_y}(\eta)\frac{d}{d\eta}\phi_{m_y}(\eta)d\eta \\ &\quad+\sqrt{m_x+\frac{1}{2}}\frac{\Delta y}{2}\sum_{\ell_x=0}^{k}\sqrt{\ell_x+\frac{1}{2}}\left(-u_{ij}^{(\ell_x,m_y)}(t)+(-1)^{m_x}u_{i-1,j}^{(\ell_x,m_y)}(t)\right) \\ &\quad+\sqrt{m_y+\frac{1}{2}}\frac{\Delta x}{2}\sum_{\ell_y=0}^{k}\sqrt{\ell_y+\frac{1}{2}}\left(-u_{ij}^{(m_x,\ell_y)}(t)+(-1)^{m_y}u_{i,j-1}^{(m_x,\ell_y)}(t)\right). \end{aligned} \tag{28}$$

2.1.2 Matrix-vector notation

For every element $I_{ij}, i = 0, \ldots, N_x, j = 0, \ldots, N_y$, the following $(k+1)\times(k+1)$ matrices are defined:

$$\mathbf{u}_{ij}(\ell_x+1, \ell_y+1) = u_{ij}^{(\ell_x,\ell_y)},\ \mathbf{u}_{i-1,j}(\ell_x+1, \ell_y+1) = u_{i-1,j}^{(\ell_x,\ell_y)},\ \mathbf{u}_{i,j-1}(\ell_x+1, \ell_y+1) = u_{i,j-1}^{(\ell_x,\ell_y)}.$$

Equation (28) can be written as

$$M\frac{d\mathbf{u}_{ij}}{dt} = A_i\mathbf{u}_{ij} + \mathbf{u}_{ij}A_j + c_{1,x}\mathbf{u}_{ij} + c_{2,x}\mathbf{u}_{i-1,j} + \mathbf{u}_{ij}c_{1,y} + \mathbf{u}_{i,j-1}c_{2,y}, \tag{29}$$

where

$$\begin{aligned}
M &= \operatorname{diag}\left(\tfrac{\Delta x\Delta y}{4}\right),\\
A_i(m_x+1, \ell_x+1) &= \tfrac{\Delta y}{2}\textstyle\int_{-1}^{1}\phi_{\ell_x}(\xi)\frac{d}{d\xi}\phi_{m_x}(\xi)d\xi,\\
A_j(\ell_y+1, m_y+1) &= \tfrac{\Delta x}{2}\textstyle\int_{-1}^{1}\phi_{\ell_y}(\eta)\frac{d}{d\eta}\phi_{m_y}(\eta)d\eta,\\
c_{1,x}(m_x+1, \ell_x+1) &= -\sqrt{m_x+\tfrac{1}{2}}\sqrt{\ell_x+\tfrac{1}{2}}\tfrac{\Delta y}{2},\\
c_{2,x}(m_x+1, \ell_x+1) &= (-1)^{m_x}\sqrt{m_x+\tfrac{1}{2}}\sqrt{\ell_x+\tfrac{1}{2}}\tfrac{\Delta y}{2},\\
c_{1,y}(\ell_y+1, m_y+1) &= -\sqrt{m_y+\tfrac{1}{2}}\sqrt{\ell_y+\tfrac{1}{2}}\tfrac{\Delta x}{2},\\
c_{2,y}(\ell_y+1, m_y+1) &= (-1)^{m_y}\sqrt{m_y+\tfrac{1}{2}}\sqrt{\ell_y+\tfrac{1}{2}}\tfrac{\Delta x}{2},
\end{aligned}$$

which are all matrices of size $(k+1)\times(k+1)$.

2.2 Time stepping using TVD Runge Kutta

The time stepping is again done using the TVD RK method of order three. Therefore equation (29) is written as

$$\frac{d}{dt}\mathbf{u}_{ij} = L(\mathbf{u}_{ij}, \mathbf{u}_{i-1,j}, \mathbf{u}_{i,j-1}), i = 0, \ldots, N_x; j = 0, \ldots, N_y.$$

The time stepping approximation, $\mathbf{w}_{ij}^{n+1}$, at time $t_{n+1}, n \geq 0$ in element I_{ij} is then given by

$$\begin{array}{rcl} \mathbf{w}_{ij}^{(1)} & = & \mathbf{w}_{ij}^{n} + \Delta t L(\mathbf{w}_{ij}^{n}, \mathbf{w}_{i-1,j}^{n}, \mathbf{w}_{i,j-1}^{n}), \\ \mathbf{w}_{ij}^{(2)} & = & \frac{3}{4}\mathbf{w}_{ij}^{n} + \frac{1}{4}\mathbf{w}_{ij}^{(1)} + \frac{1}{4}\Delta t L(\mathbf{w}_{ij}^{(1)}, \mathbf{w}_{i-1,j}^{(1)}, \mathbf{w}_{i,j-1}^{(1)}), \\ \mathbf{w}_{ij}^{n+1} & = & \frac{1}{3}\mathbf{w}_{ij}^{n} + \frac{2}{3}\mathbf{w}_{ij}^{(2)} + \frac{2}{3}\Delta t L(\mathbf{w}_{ij}^{(2)}, \mathbf{w}_{i-1,j}^{(2)}, \mathbf{w}_{i,j-1}^{(2)}), \end{array} \tag{30}$$

$i = 0, \ldots, N_x, j = 0, \ldots, N_y$.
The initial condition that is used is given by $u^0(x, y) = \sin(2\pi x) \cdot \sin(2\pi y)$.

2.2.1 Projection of the initial condition

The DG coefficients for the initial condition are found using the same procedure as in Section 1.2.1: for each element I_{ij} the initial condition,

$$u_h(x, y, 0) = \sum_{\ell_x=0}^{k} \sum_{\ell_y=0}^{k} u_{ij}^{(\ell_x,\ell_y)}(0)\phi_{\ell_x}\left(\frac{2}{\Delta x}(x - x_i)\right)\phi_{\ell_y}\left(\frac{2}{\Delta y}(y - y_j)\right),$$

is multiplied by $\phi_{m_x}\left(\frac{2}{\Delta x}(x - x_i)\right)\phi_{m_y}\left(\frac{2}{\Delta y}(y - y_j)\right), m_x, m_y \in \{0, \ldots, k\}$ and integrated over I_{ij}. Using coordinate transformations to ξ and η as well as the orthonormal property of the scaled Legendre polynomials, the coefficients are then given by

$$u_{ij}^{(m_x,m_y)}(0) = \int_{-1}^{1}\int_{-1}^{1} u_h(x_i + \frac{\Delta x}{2}\xi, y_j + \frac{\Delta y}{2}\eta, 0)\phi_{m_x}(\xi)\phi_{m_y}(\eta)d\xi d\eta.$$

The stated integral can be approximated using the Gauss Legendre quadrature. If two sets of quadrature points and weights are given by $\{\hat{x}_0, \ldots, \hat{x}_{q_x-1}, v_0, \ldots, v_{q_x-1}\}$ and $\{\hat{y}_0, \ldots, \hat{y}_{q_y-1}, w_0, \ldots, w_{q_y-1}\}$, then for an arbitrary function $f \in L^2[(-1, 1) \times (-1, 1)]$

$$\begin{array}{rcl} \int_{-1}^{1}\int_{-1}^{1} f(\xi, \eta)d\xi d\eta & = & \int_{-1}^{1}\left(\sum_{r_x=0}^{q_x-1} v_{r_x} f(\hat{x}_{r_x}, \eta)\right)d\eta \\ & = & \sum_{r_y=0}^{q_y-1} w_{r_y}\left(\sum_{r_x=0}^{q_x-1} v_{r_x} f(\hat{x}_{r_x}, \hat{y}_{r_y})\right) \\ & = & \sum_{r_y=0}^{q_y-1}\sum_{r_x=0}^{q_x-1} w_{r_y} v_{r_x} f(\hat{x}_{r_x}, \hat{y}_{r_y}). \end{array}$$

2.2.2 The CFL condition

The CFL condition in the two dimensional case is given by

$$\nu_{2D} = \frac{\Delta t}{\Delta x} + \frac{\Delta t}{\Delta y} \leq 1.$$

If $\Delta x = \Delta y$, this means that $\nu_{2D} = 2 \cdot \frac{\Delta t}{\Delta x} = 2\nu_{1D} \leq 1$, which is found if $\nu_{1D} \leq \frac{1}{2}$. This means that for every choice of N_x the value of Δt_{2D} is one half times the value of Δt_{1D}.

2.2.3 Accuracy of the numerical results

The accuracy of the initial condition can be measured using the same procedure as in Section 1.2.3 in two dimensions. Therefore the error $||e(T)||_{L^2[(-1,1)\times(-1,1)]}$ is written as

$$||e(T)||^2_{L^2[(-1,1)\times(-1,1)]} = \sum_{i=0}^{N_x}\sum_{j=0}^{N_y}\iint_{I_{ij}}(u(x,y,T)-u_h(x,y,T))^2dxdy = \sum_{i=0}^{N_x}\sum_{j=0}^{N_y}||e(T)||^2_{L^2(I_{ij})}.$$

For each element I_{ij}, the error can be approximated using a Gauss Legendre quadrature with points and weights given by

$$\{\hat{x}_0,\ldots,\hat{x}_{s_x-1},v_0,\ldots,v_{s_x-1}\} \text{ and } \{\hat{y}_0,\ldots,\hat{y}_{s_y-1},w_0,\ldots,w_{s_y-1}\},$$

such that $||e(T)||^2_{L^2(I_{ij})}$ can be approximated by

$$\frac{\Delta x\Delta y}{4}\sum_{r_y=0}^{s_y-1}\sum_{r_x=0}^{s_x-1}v_{r_x}w_{r_y}\left(u(x_i+\frac{\Delta x}{2}\hat{x}_{r_x},y_j+\frac{\Delta y}{2}\hat{y}_{r_y},T)- u_h(x_i+\frac{\Delta x}{2}\hat{x}_{r_x},y_j+\frac{\Delta y}{2}\hat{y}_{r_y},T)\right)^2.$$

Using the theory of Taylor series, the error of the initial condition can be expected to be ($(x,y)\in I_{ij}$)

$$\begin{aligned}||u^0(x,y)-u_h(x,y,0)|| &= ||u^0(x,y)-\sum_{\ell_x=0}^{k}\sum_{\ell_y=0}^{k}u_{ij}^{(\ell_x,\ell_y)}(0)\phi_{\ell_x}(\xi)\phi_{\ell_y}(\eta)||\\ &= \mathcal{O}((\Delta x)^{k+1}+(\Delta y)^{k+1}).\end{aligned}$$

If $\Delta x=\Delta y$, this reduces to $(\Delta x)^{k+1}$. In Table 5 the found values for the order are stated for $T=0.5$, using equation (24) and initial condition $u^0(x,y)=\sin(2\pi x)\cdot\sin(2\pi y)$. As can be seen in this table, the method is of order $k+1$ for $k\geq 2$. For $k=0$, the method is not so accurate: this is also found by Cockburn [7], where it is stated that the order is proven to be $k+1$ only for $k=1,2,3$. For $k=1$, the order of the error increases if the mesh is refined.

	$\Phi^0, \nu = 0.5$				
$N_x + 1$	10	20	40	80	160
$\|\|e(0.5)\|\|_\infty$	0.9991	0.8672	0.6448	0.4016	0.2246
order	-	0.2042	0.4275	0.6830	0.8384
	$\Phi^1, \nu = 0.2$				
$N_x + 1$	10	20	40	80	160
$\|\|e(0.5)\|\|_\infty$	0.2725	0.0883	0.0262	0.0070	0.0018
order	-	1.6258	1.7546	1.9016	1.9580
	$\Phi^2, \nu = 0.1$				
$N_x + 1$	10	20	40	80	160
$\|\|e(0.5)\|\|_\infty$	0.0406	0.0041	4.6935e-04	5.5158e-05	6.6163e-06
order	-	3.3079	3.1268	3.0890	3.0594
	$\Phi^3, \nu = 0.0015$				
$N_x + 1$	10	20	40	80	160
$\|\|e(0.5)\|\|_\infty$	0.0031	2.1985e-04	1.4234e-05	*	*
order	-	3.8274	3.9490	*	*
	$\Phi^4, \nu = 4 \cdot 10^{-5}$				
$N_x + 1$	10	20	40	80	160
$\|\|e(0.5)\|\|_\infty$	1.4628e-04	3.9688e-06	*	*	*
order	-	5.2038	*	*	*

Table 5: Errors and orders in 2D, $N_x = N_y, T = 0.5$ *:not computed

3 Multiwavelets

3.1 Introduction

The use of multiwavelets for DG is described by Archibald and Alpert [3, 4]. For $p = 0, 1, \ldots$, and $n = 0, 1, \ldots$, the space of piecewise polynomial functions, V_n^p, is defined as

$$V_n^p \quad = \quad \{f : f \in \mathbb{P}^{p+1}(I_j^n), j = 0, \ldots, 2^n - 1\},$$

where

$$I_j^n = (-1 + 2^{-n+1}j, -1 + 2^{-n+1}(j+1)], \tag{31}$$

and $\mathbb{P}^{p+1}(I_j^n)$ is the space of all polynomials of degree less than $p+1$ on interval I_j^n. A visualization of the intervals in $V_0^p, V_1^p, \ldots$ can be seen in Figure 2.

The space V_n^p has dimension $2^n(p+1)$ and the following nested property holds

$$V_0^p \subset V_1^p \subset \cdots \subset V_n^p \subset \cdots .$$

The scale of the basis functions for V_j^p is too coarse for the details of the functions in V_{j+1}^p: the supported intervals in V_{j+1}^p are smaller than the intervals in V_j^p such that more resolution is possible[3].

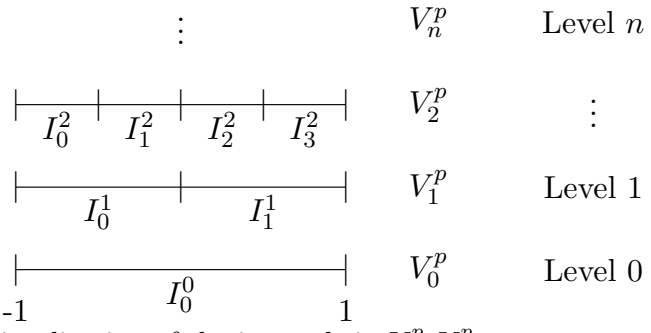


Figure 2: Visualization of the intervals in $V_0^p, V_1^p, \ldots$

In particular,

$$V_0^p = \{f \text{ is a polynomial of degree less than } p+1 \text{ on the interval } (-1, 1]\}.$$

Given an orthonormal basis $\phi_0, \ldots, \phi_p$ for V_0^p (for example ϕ_ℓ is a polynomial of degree ℓ, $\ell = 0, 1, \ldots, p$), the space V_n^p is spanned by $2^n(p+1)$ functions which are obtained from $\phi_0, \ldots, \phi_p$ by dilation and translation,

$$\phi_{\ell j}^n(x) = 2^{n/2}\phi_\ell(2^n(x+1) - 2j - 1), \ell = 0, \ldots, p, \ j = 0, \ldots, 2^n - 1, x \in I_j^n, \tag{32}$$

[3]See the website `http://fourier.eng.hmc.edu/e161/lectures/wavelets/wavelets.html`

where the coefficient j belongs to the various intervals I_j^n, see also Keinert [13]. The factor $2^{n/2}$ makes this an orthonormal basis for V_n^p. This can be proven using that $2^n(x+1)-2j-1 \in [-1,1]$ if $x \in [-1+2^{-n+1}j, -1+2^{-n+1}(j+1)]$:

$$\begin{aligned}\int_{-1}^{1}(\phi_{\ell j}^n(x))^2dx &= 2^n\int_{-1+2^{-n+1}j}^{-1+2^{-n+1}(j+1)}(\phi_\ell(2^n(x+1)-2j-1))^2dx\\ &= \int_{-1}^{1}(\phi_\ell(y))^2dy = 1.\end{aligned}$$

The functions $\phi_0,\ldots,\phi_p$ are called scaling functions. An example of these scaling functions is given in Section 3.1.1.

The multiwavelet subspace $W_n^p, n = 0,1,2,\ldots$, is defined as the orthogonal complement of V_n^p in V_{n+1}^p:

$$V_n^p \oplus W_n^p = V_{n+1}^p,\ W_n^p \perp V_n^p. \tag{33}$$

Note that therefore $W_n^p \subset V_{n+1}^p$. In equation (33) the idea of the orthogonal decomposition theorem from Linear Algebra is used.
To find a basis for W_0^p consisting of $p+1$ functions, it is necessary that these functions, $\psi_0,\ldots,\psi_p$, are piecewise polynomials. This is because $W_0^p \in V_1^p$, which has a basis consisting of $2(p+1)$ functions $\phi_{\ell j}^1, \ell = 0,\ldots,p, j = 0,1$. If the piecewise polynomial functions $\psi_0,\ldots,\psi_p$ form an orthonormal basis for W_0^p, then W_n^p is spanned by the functions

$$\psi_{\ell j}^n(x) = 2^{n/2}\psi_\ell(2^n(x+1)-2j-1), \ell = 0,\ldots,p,\ j = 0,\ldots,2^n-1, x \in I_j^n.$$

The functions $\psi_0,\ldots,\psi_p$ are the so-called multiwavelets. Section 3.1.1 contains an example of these multiwavelets.
Multiwavelets form a set of functions which, together with a set of scaling functions, can be used to approximate a function. The term multiwavelet refers to the fact that the bases for V_0^p and W_0^p contain multiple elements.

From definition (33) it follows that V_n^p can be split into $n+1$ orthogonal subspaces as

$$V_n^p = V_0^p \oplus W_0^p \oplus W_1^p \oplus \cdots \oplus W_{n-1}^p.$$

Note that following the definition given in (32), for $n \in \mathbb{N}, j \in \{0,\ldots,2^{n+1}-1\}$, $\ell \in \{0,\ldots,p\}$ the function $\phi_{\ell j}^{n+1}$ is narrower than $\phi_{\ell j}^n, j \in \{0,\ldots,2^n-1\}$ and is translated in smaller steps. Therefore, it is possible to represent finer detail.
The same discussion holds for the function $\psi_{\ell j}^{n+1}$ with respect to $\psi_{\ell j}^n$, see Burrus [6].

3.1.1 Example

As an example of the theory developed so far, the basis of scaling functions for V_0^4 is given by the scaled Legendre polynomials (see equations (3)-(7)), which can be seen in Figure 3a. The functions vanish outside the interval $[-1, 1]$.
The basis for V_1^4 can be seen in Figure 4. Functions $\phi_{\ell 0}^1$ and $\phi_{\ell 1}^1, \ell = 0, \dots 4$ vanish outside $[-1, 0]$ and $[0, 1]$, respectively.

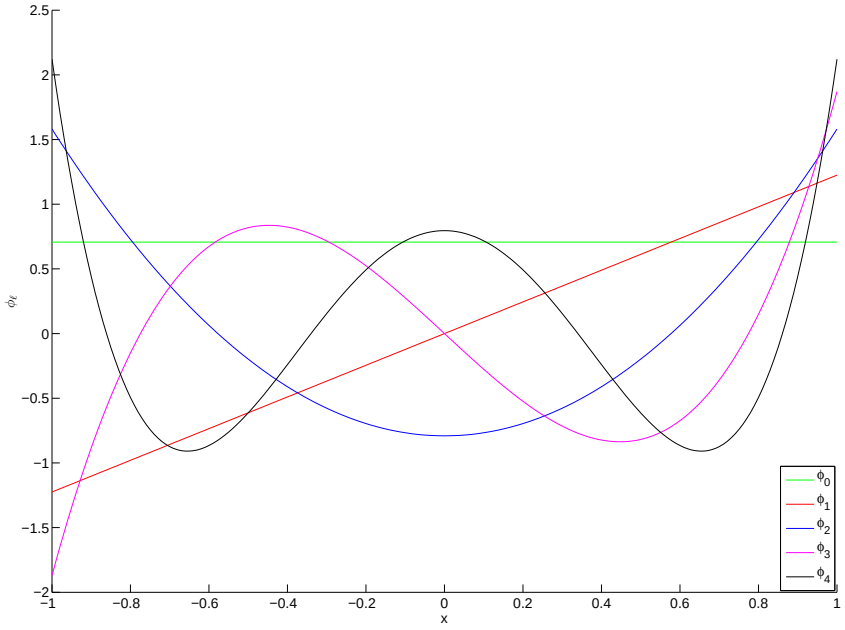
The multiwavelet basis $\{\psi_0, \dots, \psi_4\}$ for W_0^4 has the following properties ($\ell \in \{0, \dots, 4\}$):

1. If $\psi_\ell \in W_0^4$ then $\psi_\ell \in V_1^4$, but $W_0^4 \neq V_1^4$ ($V_0^4 \oplus W_0^4 = V_1^4$, see equation (33));
2. $\langle \psi_\ell, \psi_\ell \rangle = 1$: the system is orthonormal;
3. $\langle \psi_\ell, \phi_i \rangle = 0$, for all functions $\phi_i \in V_0^4$ using equation (33);
4. It is possible to define every basis function $\phi_{\ell j}^1 \in V_1^4, j = 0, 1$, using a linear combination of the basis functions $\phi_i \in V_0^4$ and $\psi_i \in W_0^4, i = 0, \dots, 4$.

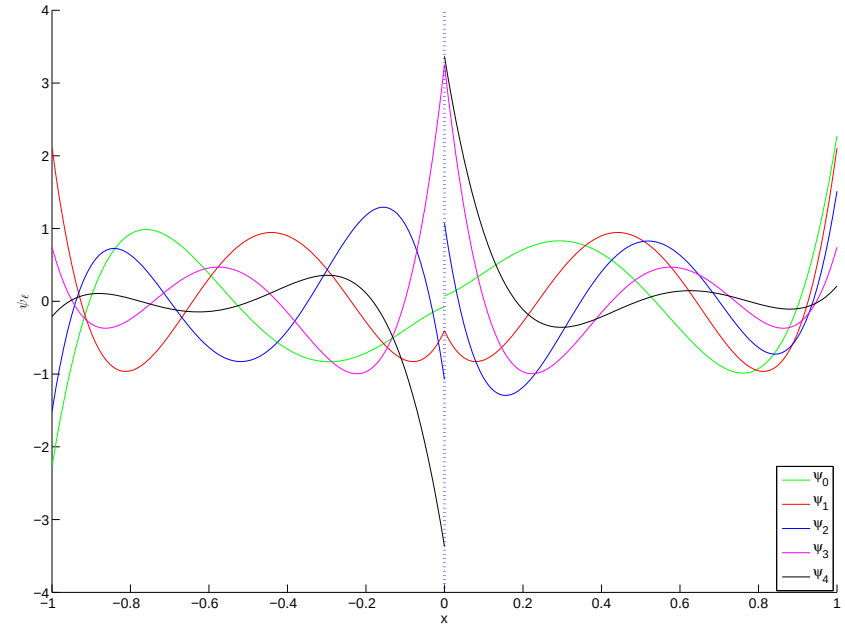
The multiwavelets that satisfy these requirements are developed by Alpert [2] and a good explanation is given in in Hovhannisyan [12].
The multiwavelet basis of Alpert that spans $W_0^p, p = 0, \dots, 4$ is given in Table 6. The functions $f_i, i = 0, \dots, p$ are given for $x \in (0, 1)$ and extended to the interval $(-1, 0)$ as an odd or even function, according to the formula $f_i(x) = (-1)^{i+p+1} f_i(-x)$. Outside the interval $(-1, 1)$, f_i is zero.

$p = 0$		
$f_0(x)$	$=$	$\sqrt{\frac{1}{2}}$
$p = 1$		
$f_0(x)$	$=$	$\sqrt{\frac{3}{2}}(-1 + 2x)$
$f_1(x)$	$=$	$\sqrt{\frac{1}{2}}(-2 + 3x)$
$p = 2$		
$f_0(x)$	$=$	$\frac{1}{3}\sqrt{\frac{1}{2}}(1 - 24x + 30x^2)$
$f_1(x)$	$=$	$\frac{1}{2}\sqrt{\frac{3}{2}}(3 - 16x + 15x^2)$
$f_2(x)$	$=$	$\frac{1}{3}\sqrt{\frac{5}{2}}(4 - 15x + 12x^2)$
$p = 3$		
$f_0(x)$	$=$	$\sqrt{\frac{15}{34}}(1 + 4x - 30x^2 + 28x^3)$
$f_1(x)$	$=$	$\sqrt{\frac{1}{42}}(-4 + 105x - 300x^2 + 210x^3)$
$f_2(x)$	$=$	$\frac{1}{2}\sqrt{\frac{35}{34}}(-5 + 48x - 105x^2 + 64x^3)$
$f_3(x)$	$=$	$\frac{1}{2}\sqrt{\frac{5}{42}}(-16 + 105x - 192x^2 + 105x^3)$
$p = 4$		
$f_0(x)$	$=$	$\sqrt{\frac{1}{186}}(1 + 30x + 210x^2 - 840x^3 + 630x^4)$
$f_1(x)$	$=$	$\frac{1}{2}\sqrt{\frac{1}{38}}(-5 - 144x + 1155x^2 - 2240x^3 + 1260x^4)$
$f_2(x)$	$=$	$\sqrt{\frac{35}{14694}}(22 - 735x + 3504x^2 - 5460x^3 + 2700x^4)$
$f_3(x)$	$=$	$\frac{1}{8}\sqrt{\frac{21}{38}}(35 - 512x + 1890x^2 - 2560x^3 + 1155x^4)$
$f_4(x)$	$=$	$\frac{1}{2}\sqrt{\frac{7}{158}}(32 - 315x + 960x^2 - 1155x^3 + 480x^4)$

Table 6: Alpert's multiwavelet construction, $x \in (0, 1)$



(a) Scaling functions $\phi_\ell, \ell = 0, \ldots, 4$



(b) Wavelets $\psi_\ell, \ell = 0, \ldots, 4$

Figure 3: Basis of scaling functions for V_0^4 and multiwavelet basis for W_0^4

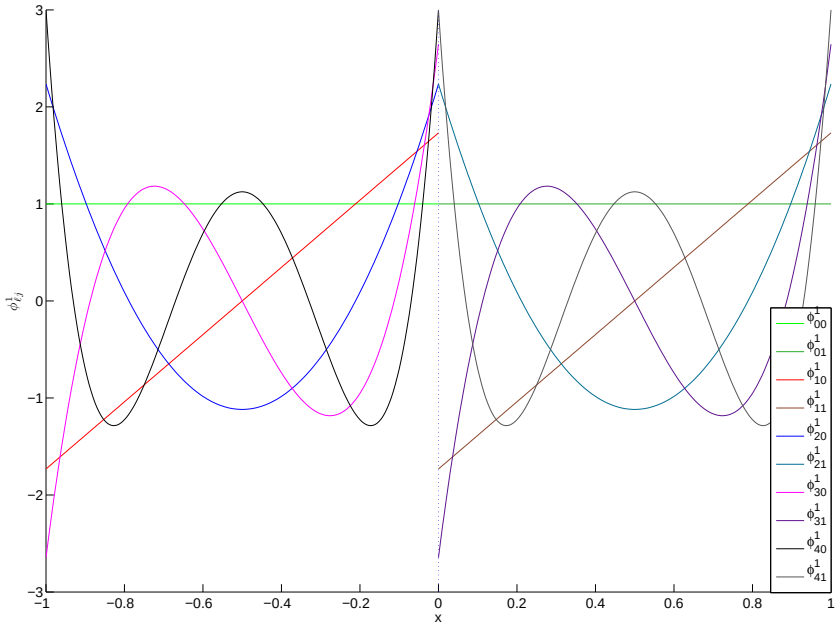


Figure 4: Basis of scaling functions for V_1^4

3.2 Approximating a function using multiwavelets

The orthogonal projection of an arbitrary function $f \in L^2(-1,1)$ onto $V_n^p, n \in \mathbb{N}$ is given by

$$P_n^p f(x) = \sum_{j=0}^{2^n-1} \sum_{\ell=0}^{p} s_{\ell j}^n \phi_{\ell j}^n(x), \tag{34}$$

which is called the single-scale decomposition of the approximate solution on level n. The coefficients $s_{\ell j}^n$ are

$$s_{\ell j}^n = \langle f, \phi_{\ell j}^n \rangle = \int_{-1+2^{-n+1}j}^{-1+2^{-n+1}(j+1)} f(x)\phi_{\ell j}^n(x)dx \tag{35}$$

$$= 2^{\frac{n}{2}} \int_{-1+2^{-n+1}j}^{-1+2^{-n+1}(j+1)} f(x)\phi_{\ell}(2^n(x+1)-2j-1)dx \tag{36}$$

$$= 2^{-\frac{n}{2}} \int_{-1}^{1} f(-1+2^{-n}(y+2j+1))\phi_{\ell}(y)dy, \tag{37}$$

the standard orthogonal projection onto an orthonormal basis.
In Keinert [13] it is proven that for any $f \in L^2(-1,1), P_n^p f \to f$ in $L^2(-1,1)$ as $n \to \infty$. Note that if $f \in V_n^p$, then $P_n^p f = f$.
The coefficients in equation (35) may be computed using a Gauss-Legendre quadrature with points $\hat{x}_0, \ldots, \hat{x}_{q-1}$ and weights $w_0, \ldots, w_{q-1}$:

$$s_{\ell j}^n \approx 2^{-\frac{n}{2}} \sum_{m=0}^{q-1} w_m f\left(-1+2^{-n}(\hat{x}_m+2j+1)\right)\phi_\ell\left(\hat{x}_m\right). \tag{38}$$

Similarly, the function f has a multiwavelet expansion, given by (using equation (33))

$$Q_n^p f(x) = P_{n+1}^p f(x) - P_n^p f(x) = \sum_{j=0}^{2^n-1} \sum_{\ell=0}^{p} d_{\ell j}^n \psi_{\ell j}^n(x), \tag{39}$$

which uses the multiwavelets $\psi_{\ell j}^n, \ell = 0, \ldots, p, j = 0, \ldots, 2^n - 1$. The coefficients are given by

$$d_{\ell j}^n = \langle f, \psi_{\ell j}^n \rangle = \int_{-1+2^{-n+1}j}^{-1+2^{-n+1}(j+1)} f(x)\psi_{\ell j}^n(x)dx.$$

Using equation (39) recursively, a relation between expansions at different levels can be

found:

$$\begin{aligned} P_{n+1}^p f(x) &= P_n^p f(x) + Q_n^p f(x) \\ &= P_{n-1}^p f(x) + Q_{n-1}^p f(x) + Q_n^p f(x) \\ &= \ldots = P_0^p f(x) + \sum_{m=0}^{n} Q_m^p f(x) \\ &= \sum_{\ell=0}^{p} s_{\ell 0}^0 \phi_\ell(x) + \sum_{m=0}^{n} \sum_{j=0}^{2^m-1} \sum_{\ell=0}^{p} d_{\ell j}^m \psi_{\ell j}^m(x). \end{aligned} \tag{40}$$

This representation of $P_{n+1}^p f(x)$ is called the multiscale decomposition. The coefficients $\{s_{\ell 0}^0\}_{\ell=0}^p$ represent the approximate solution on the coarsest level $n = 0$, and $\{d_{\ell j}^m\}$ carry the multiscale information. The detail coefficients can be seen as carriers of individual fluctuations of the solution, which, if added to the lowest-resolution information, enrich it up to the level $n + 1$ of resolution (see Iacono [10]).

3.2.1 Relation between DG and multiwavelets

As an application of the theory of multiwavelets as given in Section 3.2, the initial condition $u^0(x)$, which is used in the linear advection equation, is considered. This initial condition is written in the basis of scaled Legendre polynomials, using DG coefficients $u_i^{(\ell)}(0), \ell = 0, \ldots, k$, such that on element $i, i = 0, \ldots, N$, the approximation is given by

$$u_h(x,0) = \sum_{\ell=0}^{k} u_i^{(\ell)}(0)\phi_\ell(\xi),$$

where $\xi = \frac{2}{\Delta x}(x - x_i)$ is the corresponding local parameter and ϕ_ℓ is the scaled Legendre polynomial of degree ℓ (see Section 1.1.1).
Note that this sum of Legendre polynomials makes $u_h(x,0)$ a piecewise polynomial of degree k, such that $u_h(x,0)$ can be approximated using V_n^k for a suitable choice of n (using the definition as given in Section 3.1).

Therefore, each element $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}], i = 0, \ldots, N$, is divided into 2^n subintervals, with boundaries (using definition (31))

$$\left[x_i + \frac{\Delta x}{2}\left(-1 + 2^{-n+1}j\right), x_i + \frac{\Delta x}{2}\left(-1 + 2^{-n+1}(j+1)\right)\right], j = 0, \ldots, 2^n - 1.$$

An example of these nested grids can be seen in Figure 5.

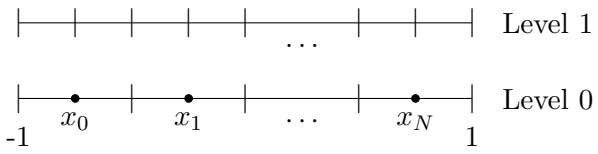


Figure 5: Example of nested grids, $n = 0, 1$

Projecting $u^0(x)$ onto the space V_n^k gives on element i:

$$\begin{aligned} P_n^k u^0(x) &= \sum_{j=0}^{2^n-1}\sum_{\ell=0}^{k} s_{\ell j}^n(i)\phi_{\ell j}^n\left(\frac{2}{\Delta x}(x - x_i)\right), \\ &= 2^{\frac{n}{2}}\sum_{j=0}^{2^n-1}\sum_{\ell=0}^{k} s_{\ell j}^n(i)\phi_\ell\left(2^n\left(\frac{2}{\Delta x}(x - x_i) + 1\right) - 2j - 1\right), \end{aligned}$$

where, using equations (35) to (37)

$$\begin{aligned} s_{\ell j}^n(i) = \langle u^0, \phi_{\ell j}^n\rangle &= \int_{-1}^{1} u^0\left(x_i + \frac{\Delta x}{2}\xi\right)\phi_{\ell j}^n(\xi)d\xi \\ &= 2^{\frac{n}{2}}\int_{-1+2^{-n+1}j}^{-1+2^{-n+1}(j+1)} u^0\left(x_i + \frac{\Delta x}{2}\xi\right)\phi_\ell(2^n(\xi+1) - 2j - 1)d\xi \\ &= 2^{-\frac{n}{2}}\int_{-1}^{1} u^0\left(x_i + \frac{\Delta x}{2}\left(-1 + 2^{-n}(y + 2j + 1)\right)\right)\phi_\ell(y)dy. \end{aligned} \tag{41}$$

Using Gauss-Legendre points $\hat{x}_0, \ldots, \hat{x}_{q-1}$ and weights $w_0, \ldots, w_{q-1}$, equation (41) can be approximated by

$$s^n_{\ell j}(i) \approx 2^{-\frac{n}{2}} \sum_{m=0}^{q-1} w_m u^0(x_i + \frac{\Delta x}{2}\left(-1 + 2^{-n}(\hat{x}_m + 2j + 1)\right))\phi_\ell\left(\hat{x}_m\right). \tag{42}$$

Note that on level 0 and element i

$$P^k_0 u^0(x) = \sum_{\ell=0}^{k} s^0_{\ell 0}(i)\phi_\ell(\xi) = \sum_{\ell=0}^{k} u_i^{(\ell)}(0)\phi_\ell(\xi) = u_h(x, 0),$$

such that $s^0_{\ell 0}(i) = u_i^{(\ell)}(0), \ell = 0, \ldots, k, i = 0, \ldots, N$.

3.3 Decomposition and reconstruction

In this section multiwavelet decomposition and reconstruction are described. The multiwavelet decomposition is a method to decompose coefficients $s^n_{\ell j}$,$\ell = 0, \ldots, p$, $n \in \mathbb{N}$, $j = 0, \ldots, 2^n - 1$ into $s^{n-1}_{\ell j}, d^{n-1}_{\ell j}, j = 0, \ldots, 2^{n-1} - 1$ recursively: it is possible to construct the scaling function and multiwavelet coefficients for all lower levels.
In Figure 6 a visualization can be seen. Decomposition is going top-down: starting in level n, all coefficients in the levels $n-1, \ldots, 0$ can be computed.

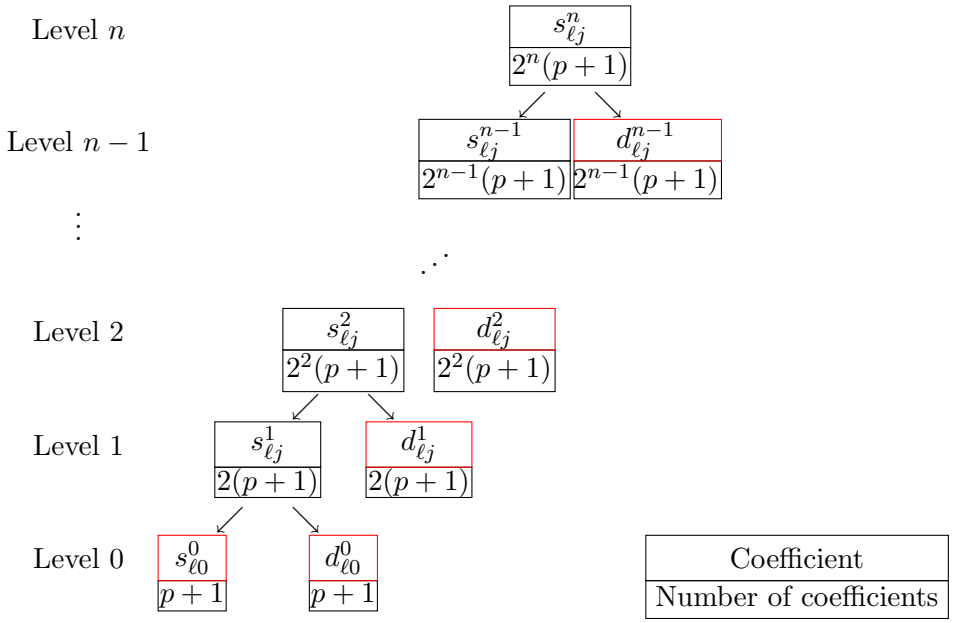


Figure 6: Visualization of decomposition and reconstruction

Multiwavelet reconstruction can be used to compute $s^n_{\ell j}$ from the coefficients $s^0_{\ell 0}, d^m_{\ell j}$, $m = 0, \ldots, n$, $j = 0, \ldots, 2^m - 1$, $\ell = 0, \ldots, p$. Reconstruction can therefore be used to construct scaling function coefficients for higher levels. All the red colored coefficients in Figure 6 together give the scaling function coefficients in level n.
In the decomposition and reconstruction steps so-called quadrature mirror filter coefficients are needed. In Section 3.3.1 these QMF coefficients are computed. In Section 3.3.2 the decomposition and reconstruction steps are described.

3.3.1 Quadrature mirror filter coefficients

In this section the QMF coefficients are computed, which are needed to decompose and reconstruct (Section 3.3.2). Two different types of coefficients are considered: lowpass filter coefficients (belonging to scaling functions), and highpass filter coefficients (belonging to multiwavelets).

Lowpass filter

Let basis function $\phi_\ell \in V_0^p$ be given, $\ell \in \{0, \ldots, p\}$. Note that $V_0^p \subset V_1^p$ such that also $\phi_\ell \in V_1^p$. This means that it is possible to write, using equation (34) and the idea in Ruch [17]

$$\phi_\ell(x) = P_1^p \phi_\ell(x) = \sum_{r=0}^{p} \langle \phi_\ell, \phi_{r0}^1 \rangle \phi_{r0}^1(x) + \sum_{r=0}^{p} \langle \phi_\ell, \phi_{r1}^1 \rangle \phi_{r1}^1(x). \tag{43}$$

Using equation (32), this becomes

$$\phi_\ell(x) = \sqrt{2} \left\{ \sum_{r=0}^{p} \langle \phi_\ell, \phi_{r0}^1 \rangle \phi_r(2(x+1)-1) + \sum_{r=0}^{p} \langle \phi_\ell, \phi_{r1}^1 \rangle \phi_r(2(x+1)-3) \right\} \tag{44}$$

$$= \sqrt{2} \sum_{j=0}^{1} \sum_{r=0}^{p} h_{\ell r}^{(j)} \phi_r(2(x-j)+1), \tag{45}$$

where $h_{\ell r}^{(j)} = \langle \phi_\ell, \phi_{rj}^1 \rangle, j = 0, 1$.

Consider $h_{\ell r}^{(0)}$. By definition,

$$h_{\ell r}^{(0)} = \langle \phi_\ell, \phi_{r0}^1 \rangle = \int_{-1}^{1} \phi_\ell(x) \phi_{r0}^1(x) dx = \sqrt{2} \int_{-1}^{1} \phi_\ell(x) \phi_r(2x+1) dx.$$

Note that $\phi_r(2x+1)$ is nonzero only if $(2x+1) \in (-1, 1]$ which means $x \in (-1, 0]$. This gives

$$h_{\ell r}^{(0)} = \sqrt{2} \int_{-1}^{0} \phi_\ell(x) \phi_r(2x+1) dx = \frac{1}{\sqrt{2}} \int_{-1}^{1} \phi_\ell \left(\frac{y-1}{2} \right) \phi_r(y) dy \tag{46}$$

$$\approx \frac{1}{\sqrt{2}} \sum_{m=0}^{q-1} w_m \phi_\ell \left(\frac{\hat{x}_m - 1}{2} \right) \phi_r(\hat{x}_m), \tag{47}$$

where the last equation uses a Gauss-Legendre quadrature with points $\hat{x}_0, \ldots, \hat{x}_{q-1}$ and weights $w_0, \ldots, w_{q-1}$.
Similarly, it is found that

$$\begin{aligned} h_{\ell r}^{(1)} &= \sqrt{2} \textstyle\int_0^1 \phi_\ell(x) \phi_r(2x-1) dx \\ &= \frac{1}{\sqrt{2}} \textstyle\int_{-1}^1 \phi_\ell \left(\frac{y+1}{2} \right) \phi_r(y) dy, \end{aligned}$$

such that $h_{\ell r}^{(1)}$ can be approximated using $\frac{1}{\sqrt{2}} \sum_{m=0}^{q-1} w_m \phi_\ell \left(\frac{\hat{x}_m+1}{2}\right) \phi_r(\hat{x}_m)$.
The coefficients $h_{\ell r}^{(j)}, j = 0, 1$, form the so-called lowpass scaling filter.

Highpass filter
The same approach can be used for the multiwavelets with respect to W_0^p. Let basis function $\psi_\ell \in W_0^p$ be given, $\ell \in \{0, \ldots, p\}$. Because $W_0^p \in V_1^p$ (see equation (33)), it holds that

$$\psi_\ell(x) = \sqrt{2} \sum_{j=0}^{1} \sum_{r=0}^{p} g_{\ell r}^{(j)} \phi_r(2(x-j)+1), \tag{48}$$

where $g_{\ell r}^{(j)} = \langle \psi_\ell, \phi_{rj}^1 \rangle, j = 0, 1$ (equations (43) to (45)).
Therefore the highpass filters are given as:

$$\begin{aligned} g_{\ell r}^{(0)} &\approx \frac{1}{\sqrt{2}} \sum_{m=0}^{q-1} w_m \psi_\ell \left(\frac{\hat{x}_m-1}{2}\right) \phi_r(\hat{x}_m), \\ g_{\ell r}^{(1)} &\approx \frac{1}{\sqrt{2}} \sum_{m=0}^{q-1} w_m \psi_\ell \left(\frac{\hat{x}_m+1}{2}\right) \phi_r(\hat{x}_m). \end{aligned}$$

3.3.2 Decomposition and reconstruction

Decomposition

As described in the introduction of Section 3.3, multiwavelet decomposition starts with scaling function coefficients $s^n_{\ell j}$,$\ell = 0, \ldots, p$, $n \in \mathbb{N}, j = 0, \ldots, 2^n - 1$ and constructs $s^{n-1}_{\ell j}, d^{n-1}_{\ell j}, j = 0, \ldots, 2^{n-1} - 1$. This can be done using equations (32), (35), and (45) for an arbitrary function $f \in L^2(-1, 1)$:

$$\begin{aligned}
s^{n-1}_{\ell j} &= \langle f, \phi^{n-1}_{\ell j} \rangle = \langle f, 2^{\frac{n-1}{2}} \phi_\ell(2^{n-1}(x+1) - 2j - 1) \rangle \\
&= \left\langle f, 2^{\frac{n-1}{2}} \sqrt{2} \sum_{r=0}^{p} \Big\{ h^{(0)}_{\ell r} \phi_r(2(2^{n-1}(x+1) - 2j - 1) + 1) \right. \\
&\quad \left. + h^{(1)}_{\ell r} \phi_r(2(2^{n-1}(x+1) - 2j - 1) - 1) \Big\} \right\rangle \\
&= \sum_{r=0}^{p} \Big(h^{(0)}_{\ell r} \left\langle f, 2^{\frac{n}{2}} \phi_r(2^n(x+1) - 2 \cdot 2j - 1) \right\rangle \\
&\quad + \; h^{(1)}_{\ell r} \left\langle f, 2^{\frac{n}{2}} \phi_r(2^n(x+1) - 2(2j+1) - 1) \right\rangle \Big) \\
&= \sum_{r=0}^{p} \left(h^{(0)}_{\ell r} s^n_{r,2j} + h^{(1)}_{\ell r} s^n_{r,2j+1} \right). \qquad (49)
\end{aligned}$$

Likewise, it holds that

$$d^{n-1}_{\ell j} = \sum_{r=0}^{p} \left(g^{(0)}_{\ell r} s^n_{r,2j} + g^{(1)}_{\ell r} s^n_{r,2j+1} \right). \qquad (50)$$

Thus, starting with $2^n(p+1)$ values for $s^n_{\ell j}$ and $d^n_{\ell j}$, the decomposition procedure can be applied repeatedly to compute the coefficients on coarser levels, $m = n-1, n-2, \ldots, 0$.

Reconstruction

Multiwavelet reconstruction can easily be found using matrix-vector products. Reconstruction can be used to compute $s^n_{\ell j}$ from the coefficients $s^0_{\ell 0}, d^m_{\ell j}$, $m = 0, \ldots, n$, $j = 0, \ldots, 2^m - 1$, $\ell = 0, \ldots, p$, using

$$s^n_{\ell,2j} = \sum_{r=0}^{p} \left(h^{(0)}_{r\ell} s^{n-1}_{rj} + g^{(0)}_{r\ell} d^{n-1}_{rj} \right), \qquad (51)$$

$$s^n_{\ell,2j+1} = \sum_{r=0}^{p} \left(h^{(1)}_{r\ell} s^{n-1}_{rj} + g^{(1)}_{r\ell} d^{n-1}_{rj} \right), \qquad (52)$$

which is proven in Appendix A.

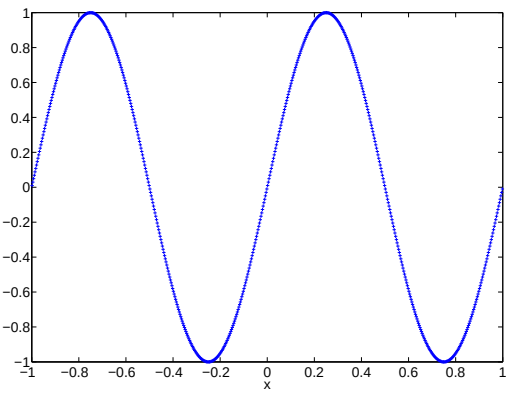
3.3.3 Application

In this section the application of Section 3.2.1 is used to give an example of the decomposition and reconstruction steps that are developed in Section 3.3.2. Recall that the values for $s_{\ell j}^n(i)$ were found in equation (42), where $i = 0, \ldots, N, \ell = 0, \ldots, k$, $n \in \mathbb{N}$ and $j = 0, \ldots, 2^n - 1$. Now the decomposition procedure as described in Section 3.3.2 is used to compute scaling function coefficients $s_{\ell j}^m(i)$ and multiwavelet coefficients $d_{\ell j}^m(i)$, for $m = n-1, n-2, \ldots, 0, j = 0, \ldots, 2^m - 1$. To do this, the matrices $H^{(0)}, H^{(1)}, G^{(0)}, G^{(1)}$ are needed (with elements as stated in Section 3.3.1), which are independent of grid cell i. Using this information, the approximation of the function on element i can be written as

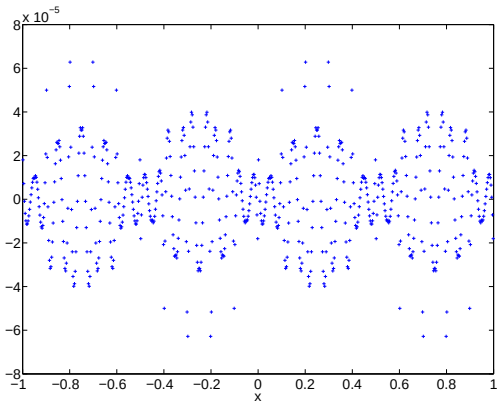
$$P_n^k u^0(x) = \sum_{\ell=0}^{k} s_{\ell 0}^0(i)\phi_\ell\left(\frac{2}{\Delta x}(x - x_i)\right) + \sum_{m=0}^{n-1}\sum_{j=0}^{2^m-1}\sum_{\ell=0}^{k} d_{\ell j}^m(i)\psi_{\ell j}^m\left(\frac{2}{\Delta x}(x - x_i)\right).$$

In Figure 7 the different parts of the projection $P_n^k u^0(x)$ are plotted, where initial condition $u^0(x) = \sin(2\pi x)$ is used.
Figure 7a contains $\sum_{\ell=0}^{k} s_{\ell 0}^0(i)\phi_\ell\left(\frac{2}{\Delta x}(x - x_i)\right)$ over each element i (the coarse approximation), where $s_{\ell 0}^0, \ell = 0, \ldots, k$ are found using equation (49) recursively. Figures 7b-7f are visualizations of $\sum_{j=0}^{2^m-1}\sum_{\ell=0}^{k} d_{\ell j}^m(i)\psi_{\ell j}^m\left(\frac{2}{\Delta x}(x - x_i)\right)$, $m = 0, \ldots, 4$ respectively (the finer details), where $d_{\ell j}^m, m = 0, \ldots, 4, \ell = 0, \ldots, k, j = 0, \ldots, 2^m - 1$ are found using equation (50) several times.

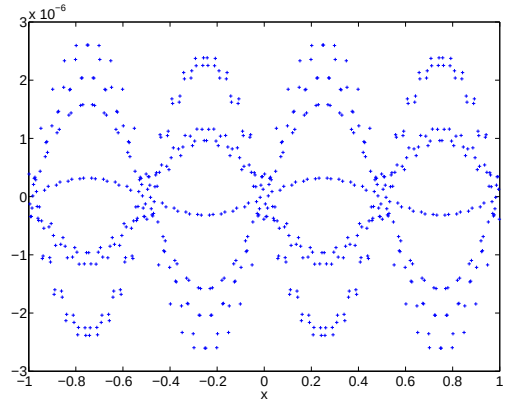
Because $u^0(x)$ is a continuous function, the contribution of the coefficients $d_{\ell j}^m$ is very small ($\mathcal{O}(10^{-4})$ for $m = 0, \mathcal{O}(10^{-10})$ for $m = 4$).



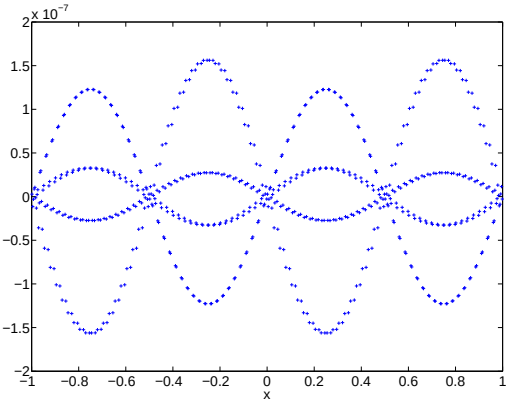
(a) First approximation, using $s^0_{\ell 0}(i)$



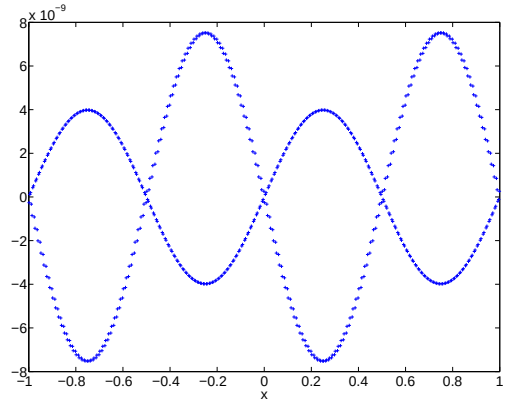
(b) Details using $d^0_{\ell j}(i)$



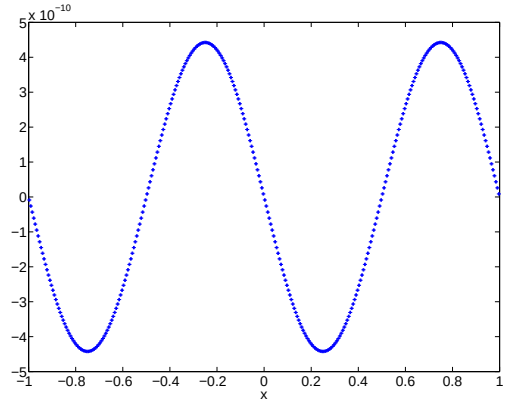
(c) Details using $d^1_{\ell j}(i)$



(d) Details using $d^2_{\ell j}(i)$



(e) Details using $d^3_{\ell j}(i)$



(f) Details using $d^4_{\ell j}(i)$

Figure 7: Decomposition procedure, $u^0(x) = \sin(2\pi x), N + 1 = 20, k = 3, n = 5$

4 Limiters

In practical applications initial conditions may contain discontinuities, or nonlinear equations with a smooth initial condition may develop a discontinuity. In that case, higher order accurate methods of order $s \geq 2$ fail near these discontinuities, and unphysical oscillations may be generated. Lower order methods ($s = 1$) have the advantage of keeping the solution monotonically varying in regions where the solution should be monotone, even though the accuracy is not very good, see Leveque [15].
A method is called monotonicity-preserving if

$$\overline{w}_j^n \geq \overline{w}_{j+1}^n, \forall j \in \{0, \ldots, N\},$$

implies that

$$\overline{w}_j^{n+1} \geq \overline{w}_{j+1}^{n+1}, \forall j \in \{0, \ldots, N\},$$

where $\overline{w}_j^n$ is the average of the approximation for $u_h(x, t^n), x \in I_j$.
This monotone property aids lower order methods in preventing unphysical oscillations that higher order methods produce.

To illustrate this point, consider the following example,

$$u^0(x) = \begin{cases} \frac{1}{6}(G(x, \beta, z - \delta) + G(x, \beta, z + \delta) + 4G(x, \beta, z)), & x \in [-0.8, -0.6], \\ 1, & x \in [-0.4, -0.2], \\ 1 - |10(x - 0.1)|, & x \in [0, 0.2], \\ \frac{1}{6}(F(x, \alpha, a - \delta) + F(x, \alpha, a + \delta) + 4F(x, \alpha, z)), & x \in [0.4, 0.6], \\ 0, & \text{else}, \end{cases}$$

where $G(x, \beta, z) = e^{-\beta(x-z)^2}, F(x, \alpha, a) = \sqrt{\max(1 - \alpha^2(x - a)^2, 0)}$, $a = 0.5$, $z = -0.7$, $\delta = 0.005, \alpha = 10$ and $\beta = \frac{\log(2)}{36\delta^2}$.
The numerical method described in Section 1.2 uses a piecewise polynomial approximation space, Φ^k, and is proven to be of order $k + 1$.
In Figure 8 the behavior of the solution using a lower ($k = 0$) and higher ($k \geq 1$) order method is shown. Clearly, the method of order 1 smears the solution and is not accurate, whereas the method of order 4 generates unphysical oscillations in the neighbourhood of the discontinuities.

The best features of both lower and higher order methods are combined in so-called high-resolution methods, such that the discontinuous portion of the solution remains nonoscillatory (lower order accurate) while the smooth portion remains accurate (higher order accurate). These high-resolution methods use a limiter that depend on how the solution is behaving. This technique is borrowed from finite volume methodology as described in Leveque [15].

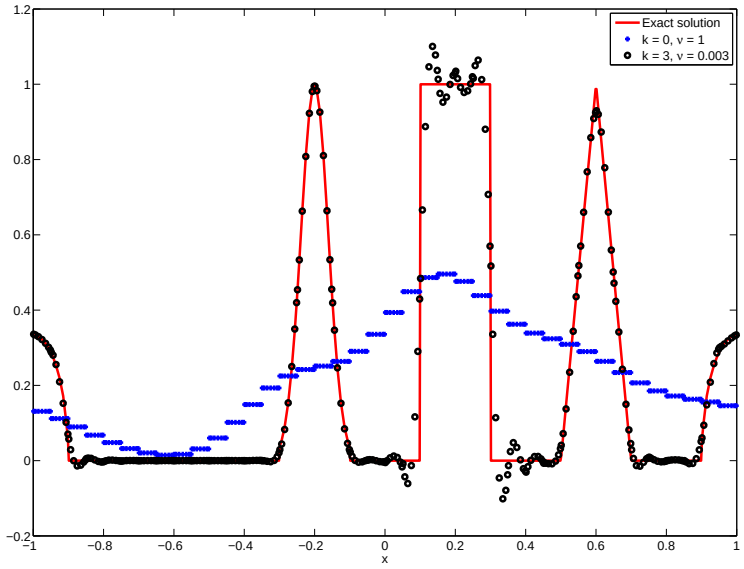


Figure 8: Results using 40 elements for $T = 0.5$

Oscillations in the solution are measured by the notion of total variation of a function. The total variation of the discrete function $\mathbf{w} = \{w_0, w_1, \ldots, w_N\}$ is given by

$$\mathrm{TV}(\mathbf{w}) = |w_0 - w_N| + \sum_{j=1}^{N} |w_j - w_{j-1}|,$$

where the first term arises from assumed periodic boundary conditions, see Leveque [15]. Note that the true solution to the advection equation simply propagates along characteristics with unchanged shape, such that the total variation must be constant in time. If a numerical method introduces oscillations, then it is expected that the total variation increases with time.
A method is called total variation diminishing (TVD) if, for any set of data $\mathbf{w}^n$ at time $t^n = n\Delta t$, the values $\mathbf{w}^{n+1}$ computed by the method satisfy $\mathrm{TV}(\mathbf{w}^{n+1}) \leq \mathrm{TV}(\mathbf{w}^n)$.
The stability of a method can also be considered using total variation boundedness. A numerical method is total variation bounded (TVB) if, for any initial data $\mathbf{w}^0$ with $\mathrm{TV}(\mathbf{w}^0) < \infty$ and for any time T, there is a constant $R > 0$ and a value $\Delta t_0 > 0$ such that $\mathrm{TV}(\mathbf{w}^n) \leq R$ for all $n\Delta t \leq T$ whenever $\Delta t < \Delta t_0$ (see [15]).

High-resolution methods that use limiters are distinguished as total variation diminishing in the mean (TVDM), and total variation bounded in the mean (TVBM). Hereby the mean $\overline{w}_j$ of w_j on interval $I_j, j = 0, \ldots, N$, is considered.

In Section 4.1 some examples of limiters are given.

4.1 Examples of limiters

The following algorithm is used to limit the approximation of the solution, using the third order TVD RK of equation (19) (N_t is the number of timesteps):

Algorithm 1 Calculate limited version of $\mathbf{w}^{n+1}$ using $\mathbf{w}^n$

$\mathbf{w}^0 = \mathbf{u}^0$
for $n = 0 : N_t - 1$ **do**
  Enforce periodic boundary conditions: $w_{-1}^{(\ell)}(t^n) = w_N^{(\ell)}(t^n), \ell = 0, \ldots, k$;
  Compute limited solution
  Enforce periodic boundary conditions: $w_{-1}^{(\ell)}(t^n) = w_N^{(\ell)}(t^n), \ell = 0, \ldots, k$;
  $\rightarrow \mathbf{w}^{(1)}$: the first RK step
  Enforce periodic boundary conditions
  Compute limited solution
  Enforce periodic boundary conditions
  $\rightarrow \mathbf{w}^{(2)}$: the second RK step
  Enforce periodic boundary conditions
  Compute limited solution
  Enforce periodic boundary conditions
  $\rightarrow \mathbf{w}^{n+1}$: the last RK step
end for
Enforce periodic boundary conditions
Compute limited solution
Enforce periodic boundary conditions

Examples of TVDM and TVBM slope limiters are given in Cockburn [7, 8]. Several groups of limiters are discussed in this report: minmod limiters, projection limiters, moment limiters, WENO limiters, and the multiwavelet limiters that are used today.

4.1.1 Minmod limiters

Minmod limiters use the so-called minmod function m, given by

$$m(a_1, \ldots, a_q) = \begin{cases} s \cdot \min_{1 \le r \le q} |a_r| & \text{if sign } (a_1) = \cdots = \text{sign}(a_q) = s, \\ 0 & \text{otherwise.} \end{cases}$$

An example is the monotonized central-difference limiter, proposed by Van Leer and described in Leveque [15]: the approximation $u_h(x, t^n)$ of the solution at I_j, $j = 0, \ldots, N$ and time t^n is written as a linear function of the form

$$u_h(x, t^n) = \overline{u}_j^n + \sigma_j(x - x_j). \tag{53}$$

The coefficient $\overline{u}_j^n$ denotes the average over element I_j, given by

$$\overline{u}_j^n = \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} u_h(x, t^n) dx.$$

Note that, using the orthonormal property of the scaled Legendre polynomials, for $m \in \{0, \ldots, k\}$

$$\int_{-1}^{1} \phi_m(x)dx = \int_{-1}^{1} \phi_m(x) \cdot 1 dx = \sqrt{2} \int_{-1}^{1} \phi_m(x)\phi_0(x)dx = \begin{cases} \sqrt{2} & \text{if } m = 0, \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, the average can be computed as follows, using equation (11),

$$\begin{aligned} \overline{u}_j^n &= \frac{1}{\Delta x} \int_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \sum_{\ell=0}^{k} u_j^{(\ell)}(t^n)\phi_\ell \left(\frac{2}{\Delta x}(x - x_j) \right) dx \\ &= \frac{1}{\Delta x} \sum_{\ell=0}^{k} u_j^{(\ell)}(t^n) \cdot \frac{\Delta x}{2} \int_{-1}^{1} \phi_\ell(\xi) d\xi \\ &= \frac{1}{\sqrt{2}} u_j^{(0)}(t^n). \end{aligned}$$

In Van Leer's limiter the slope σ_j^n is chosen to be

$$\sigma_j^n = m\left(\left(\frac{\overline{u}_{j+1}^n - \overline{u}_{j-1}^n}{2\Delta x} \right), 2\left(\frac{\overline{u}_j^n - \overline{u}_{j-1}^n}{\Delta x} \right), 2\left(\frac{\overline{u}_{j+1}^n - \overline{u}_j^n}{\Delta x} \right) \right).$$

Since the solution is approximated to be a linear function on element j, only DG coefficients $u_j^{(0)}(t^n)$ and $u_j^{(1)}(t^n)$ arise in the expansion:

$$u(x, t^n) = \sum_{\ell=0}^{1} u_j^{(\ell)}(t^n)\phi_\ell(\xi), x \in I_j.$$

The value of the DG coefficients can be found using equation (53).
Note that the method lowers the order in the neighbourhood of shocks: the approximation is only linear in these elements.

4.1.2 Projection limiters

Projection limiters restrict the solution after each RK stage to eliminate oscillations. This limiting technique has been developed by Cockburn and Shu [8], and is well described in Biswas [5].
Approximation $u_h(x, t^n)$ at time t^n on element $I_j, j = 0, \ldots, N$ is written as

$$u_h(x, t^n) = u_j^{(0)}(t^n)\phi_0(\xi) + \sum_{\ell=1}^{k} u_j^{(\ell)}(t^n)\phi_\ell(\xi) = \frac{1}{\sqrt{2}} u_j^{(0)} + s_j(\xi, t^n),$$

where $s_j(\xi, t^n) = \sum_{\ell=1}^{k} u_j^{(\ell)}(t^n)\phi_\ell(\xi)$.

The limited values for the boundaries $s_j(-1,t^n)$ and $s_j(1,t^n)$ are given by

$$s_j(1,t^n)^* = m(s_j(1,t^n), \frac{1}{\sqrt{2}}(u_j^{(0)}(t^n) - u_{j-1}^{(0)}(t^n)), \frac{1}{\sqrt{2}}(u_{j+1}^{(0)}(t^n) - u_j^{(0)}(t^n)))$$
$$-s_j(-1,t^n)^* = m(-s_j(-1,t^n), \frac{1}{\sqrt{2}}(u_j^{(0)}(t^n) - u_{j-1}^{(0)}(t^n)), \frac{1}{\sqrt{2}}(u_{j+1}^{(0)}(t^n) - u_j^{(0)}(t^n))).$$

If $k = 1$ or $k = 2$, then $u_j^{(1)}(t^n)$ and $u_j^{(2)}(t^n)$ can be determined uniquely.
If $k \geq 3$ and $s_j(1,t^n)^* \neq s_j(1,t^n)$, or $s_j(-1,t^n)^* \neq s_j(-1,t^n)$, then the coefficients $u_j^{(\ell)}(t^n)$ are set to zero for $\ell \geq 3$.
The order of the approximation is lowered in the neighbourhood of shocks and the solution accuracy is reduced: only a first-order accuracy is obtained. Besides that, the limiting flattens solutions near smooth extrema where no limiting is required.

4.1.3 Moment limiter

The moment limiter, described in Krivodonova [14], uses the same idea as the minmod limiter of Section 4.1.1, but is applied to the coefficients $u_j^{(\ell)}(t^n)$ at time $t^n, j = 0,\ldots,N$, $\ell = 0,\ldots,k$. The limiter works for higher order methods, but causes problems with multi-dimensions and complex geometries.

4.1.4 WENO

Another limiter is found by using the weighted essentially nonoscillatory (WENO) finite volume methodology as described in Qiu [16]. WENO replaces the solution in the troubled cells with reconstructed polynomials which maintain the original cell averages, have the same orders of accuracy as before, but are less oscillatory. These new polynomials are reconstructed from the information of neighbouring cells.
The main difficulty is maintaining the original high order accuracy.

4.1.5 Multiwavelet limiter

The multiwavelet limiter that is currently used is applied on the highest refinement level n. First a threshold $\epsilon > 0$ is used where $d_{\ell j}^n$ is set equal to zero if $|d_{\ell j}^n| < \epsilon$. Next, a projection limiter is used as described in Section 4.1.2.
The multiwavelet limiter works only for low order methods.

5 Further research

In Section 4.1 some examples of limiters for DG were given. All of them have disadvantages, such as the low order of the limited version and the disability to work with multidimensions or complex geometries.
Multiwavelets are expected to be a promising tool to overcome these problems. In the MSc thesis that follows this literature study, the questions as stated in the introduction are addressed.

References

[1] Milton Abramowitz and Irene A. Stegun, editors. *Handbook of Mathematical Functions With Formulas, Graphs, and Mathematical Tables*, volume 55 of *Applied Mathematics*. National Bureau of Standards, Washington, D.C., 1964.

[2] Bradley K. Alpert. A Class of Bases in L^2 for the Sparse Representation of Integral Operators. *SIAM J. on Mathematical Analysis*, 24:246–262, 1993.

[3] Rick Archibald, George Fann, and William Shelton. Adaptive Discontinuous Galerkin Methods with Multiwavelets Bases. *Applied Numerical Mathematics*, 61(7):879–890, 2011.

[4] B.Alpert, G. Beylkin, D. Gines, and L.Vozovoi. Adaptive Solution of Partial Differential Equations in Multiwavelet Bases. *Journal of Computational Physics*, 182:149–190, 2002.

[5] Rupak Biswas, Karen D. Devine, and Joseph E. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14:255–283, 1994.

[6] C.Sidney Burrus, Ramesh A. Gopinath, and Haitao Guo. *Introduction to Wavelets and Wavelet Transforms*. Prentice-Hall, New Jersey, 1998.

[7] Bernardo Cockburn. *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, volume 1697 of *Lecture Notes in Mathematics*, chapter An introduction to the Discontinuous Galerkin method for convection-dominated problems. Springer Berlin/Heidelberg, 1998.

[8] Bernardo Cockburn and Chi-Wang Shu. TVB Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for Conservation Laws II: General Framework. *Mathematics of Computation*, 52(186):411–435, 1989.

[9] Bernardo Cockburn and Chi-Wang Shu. Runge-Kutta Discontinuous Galerkin Methods for Convection-Dominated Problems. *Journal of Scientific Computing*, 16(3):173–261, 2001.

[10] F.Iacono, G.May, S.Müller, and R.Schäfer. A High-Order Discontinuous Galerkin Discretization with Multiwavelet-Based Grid Adaptation for Compressible Flows. Technical Report AICES-2011/08-02, Aachen Institute for Advanced Study in Computational Engineering Science, Aachen, aug 2011. http://www.aices.rwth-aachen.de/preprints.

[11] Sigal Gottlieb and Chi-Wang Shu. Total Variation Diminishing Runge-Kutta Schemes. *Mathematics of Computation*, 67(221):73–85, 1998.

[12] Nune Hovhannisyan, Siegfried Müller, and Roland Schäfer. Adaptive Multiresolution Discontinuous Galerkin Schemes for Conservation Laws. Report 311, Institut für Geometrie und Praktische Mathematik, Aachen, sep 2010. http://www.igpm.rwth-aachen.de/en/reports2010.

[13] Fritz Keinert. *Wavelets and Multiwavelets*. Studies in Advanced Mathematics. Chapman and Hall/CRC, Florida, 2004.

[14] Lilia Krivodonova. Limiters for high-order discontinuous Galerkin methods. *Journal of Computational Physics*, 226:879–896, 2007.

[15] Randall J. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, New York, sixth edition, 2002.

[16] Jianxian Qiu and Chi-Wang Shu. Runge-Kutta Discontinuous Galerkin Method using WENO limiters. *Society for Industrial and Applied Mathematics*, 26(3):907–929, 2005.

[17] David K. Ruch and Patrick J. van Fleet. *Wavelet theory: An Elementary Approach With Applications*. John Wiley & Sons, Inc., New Jersey, 2009.

A Proof of multiwavelet reconstruction

In this section, a proof of the multiwavelet reconstruction is given (see Section 3.3.2), using matrix-vector products. This means that the following matrices are defined:

$$H^{(j)} = \left\{ h_{i\ell}^{(j)} \right\}, G^{(j)} = \left\{ g_{i\ell}^{(j)} \right\}, \tag{54}$$

$j = 0, 1;\ i, \ell = 0, \ldots, p$, and

$$S^m = \left(\mathbf{s}_0^m \ldots \mathbf{s}_{2^m-1}^m \right), D^m = \left(\mathbf{d}_0^m \ldots \mathbf{d}_{2^m-1}^m \right), m \in \mathbb{N}, \tag{55}$$

where

$$\begin{aligned} \mathbf{s}_j^m &= \left(s_{0j}^m \cdots s_{pj}^m \right)^\top, \\ \mathbf{d}_j^m &= \left(d_{0j}^m \cdots d_{pj}^m \right)^\top, \end{aligned}$$

$j = 0, \ldots, 2^m - 1$.
Using this notation, the following relations hold (see Alpert [4]):

$$H^{(0)} H^{(0)^\top} + H^{(1)} H^{(1)^\top} = I, \tag{56}$$

$$G^{(0)} G^{(0)^\top} + G^{(1)} G^{(1)^\top} = I, \tag{57}$$

$$H^{(0)} G^{(0)^\top} + H^{(1)} G^{(1)^\top} = 0. \tag{58}$$

$$G^{(0)} H^{(0)^\top} + G^{(1)} H^{(1)^\top} = 0. \tag{59}$$

Equation (56) can be proven as follows, using equation (45) and the orthonormality of the functions $\phi_0, \ldots, \phi_p$:

$$\delta_{i\ell} = \langle \phi_i, \phi_\ell \rangle = \sqrt{2} \sum_{r=0}^{p} \left(h_{ir}^{(0)} \langle \phi_r(2x+1), \phi_\ell(x) \rangle + h_{ir}^{(1)} \langle \phi_r(2x-1), \phi_\ell(x) \rangle \right). \tag{60}$$

The two inner products on the righthand side are computed separately, using equation (45) a second time, and the same reasoning as in equations (46) and (47):

$$\begin{aligned} \langle \phi_r(2x+1), \phi_\ell(x) \rangle &= \sqrt{2} \int_{-1}^{0} \phi_r(2x+1) \left(\sum_{q=0}^{p} \left(h_{\ell q}^{(0)} \phi_q(2x+1) + h_{\ell q}^{(1)} \phi_q(2x-1) \right) \right) dx \\ &= \sqrt{2} \sum_{q=0}^{p} h_{\ell q}^{(0)} \int_{-1}^{0} \phi_r(2x+1) \phi_q(2x+1) dx = \sqrt{2} \sum_{q=0}^{p} h_{\ell q}^{(0)} \frac{1}{2} \langle \phi_r, \phi_q \rangle \\ &= \frac{1}{\sqrt{2}} h_{\ell r}^{(0)}. \end{aligned}$$

The last relation used the orthonormality relation of the scaling functions.
The second inner product in equation (60) gives

$$\langle \phi_r(2x-1), \phi_\ell(x) \rangle = \frac{1}{\sqrt{2}} h_{\ell r}^{(1)},$$

so that equation (60) becomes

$$\delta_{i\ell} = \langle \phi_i, \phi_\ell \rangle = \sqrt{2} \sum_{r=0}^{p} \left(h_{ir}^{(0)} \frac{1}{\sqrt{2}} h_{\ell r}^{(0)} + h_{ir}^{(1)} \frac{1}{\sqrt{2}} h_{\ell r}^{(1)} \right) = \sum_{r=0}^{p} \left(h_{ir}^{(0)} h_{\ell r}^{(0)} + h_{ir}^{(1)} h_{\ell r}^{(1)} \right),$$

from which relation (56) easily follows.
To prove relation (57), note that equation (48) gives

$$\delta_{i\ell} = \langle \psi_i, \psi_\ell \rangle = \sqrt{2} \sum_{r=0}^{p} \left(g_{ir}^{(0)} \langle \phi_r(2x+1), \psi_\ell(x) \rangle + g_{ir}^{(1)} \langle \phi_r(2x-1), \psi_\ell(x) \rangle \right).$$

The same approach as above can be used.
Relations (58) and (59) use the orthogonality relation as in equation (33) to find

$$\begin{aligned} 0 &= \langle \phi_i, \psi_\ell \rangle = \sqrt{2} \textstyle\sum_{r=0}^{p} \left(h_{ir}^{(0)} \langle \phi_r(2x+1), \psi_\ell(x) \rangle + h_{ir}^{(1)} \langle \phi_r(2x-1), \psi_\ell(x) \rangle \right), \\ 0 &= \langle \psi_i, \phi_\ell \rangle = \sqrt{2} \textstyle\sum_{r=0}^{p} \left(g_{ir}^{(0)} \langle \phi_r(2x+1), \phi_\ell(x) \rangle + g_{ir}^{(1)} \langle \phi_r(2x-1), \phi_\ell(x) \rangle \right). \end{aligned}$$

Introducing

$$U = \begin{pmatrix} H^{(0)} & H^{(1)} \\ G^{(0)} & G^{(1)} \end{pmatrix},$$

equations (56) to (59) give $UU^\top = I$. This means that U is an orthogonal matrix and satisfies $U^\top = U^{-1}$. This condition gives rise to an additional set of relations:

$$\begin{aligned} {H^{(0)}}^\top H^{(0)} + {G^{(0)}}^\top G^{(0)} &= I, \\ {H^{(1)}}^\top H^{(1)} + {G^{(1)}}^\top G^{(1)} &= I, \\ {H^{(0)}}^\top H^{(1)} + {G^{(0)}}^\top G^{(1)} &= 0. \\ {H^{(1)}}^\top H^{(0)} + {G^{(1)}}^\top G^{(0)} &= 0. \end{aligned}$$

Using the matrix notation as introduced in (54) and (55), equations (49) and (50) can be equivalently written as:

$$\mathbf{s}_j^{n-1} = H^{(0)} \mathbf{s}_{2j}^n + H^{(1)} \mathbf{s}_{2j+1}^n, \tag{61}$$

$$\mathbf{d}_j^{n-1} = G^{(0)} \mathbf{s}_{2j}^n + G^{(1)} \mathbf{s}_{2j+1}^n. \tag{62}$$

The multiwavelet reconstruction follows from multiplying equation (61) on the left by ${H^{(0)}}^\top$, equation (62) on the left by ${G^{(0)}}^\top$, and summing such that

$$\begin{aligned} &{H^{(0)}}^\top \mathbf{s}_j^{n-1} + {G^{(0)}}^\top \mathbf{d}_j^{n-1} = \\ &\quad = {H^{(0)}}^\top \left(H^{(0)} \mathbf{s}_{2j}^n + H^{(1)} \mathbf{s}_{2j+1}^n \right) + {G^{(0)}}^\top \left(G^{(0)} \mathbf{s}_{2j}^n + G^{(1)} \mathbf{s}_{2j+1}^n \right) \\ &\quad = \left\{ {H^{(0)}}^\top H^{(0)} + {G^{(0)}}^\top G^{(0)} \right\} \mathbf{s}_{2j}^n + \left\{ {H^{(0)}}^\top H^{(1)} + {G^{(0)}}^\top G^{(1)} \right\} \mathbf{s}_{2j+1}^n \\ &\quad = \mathbf{s}_{2j}^n. \end{aligned}$$

The same approach leads to

$$ {H^{(1)}}^\top \mathbf{s}_j^{n-1} + {G^{(1)}}^\top \mathbf{d}_j^{n-1} = \mathbf{s}_{2j+1}^n, $$

from which equations (51) and (52) immediately follow.