

UNIVERSIDAD DE CONCEPCIÓN



CENTRO DE INVESTIGACIÓN EN
INGENIERÍA MATEMÁTICA (CI²MA)



Approximate Lax-Wendroff discontinuous Galerkin methods for
hyperbolic conservation laws

RAIMUND BÜRGER, SUDARSHAN K. KENETTINKARA,
DAVID ZORÍO

PREPRINT 2017-01

SERIE DE PRE-PUBLICACIONES

Approximate Lax-Wendroff discontinuous Galerkin methods for hyperbolic conservation laws

Raimund Bürger^a, Sudarshan Kumar Kenettinkara^b, David Zorío^c

^a*CI²MA and Departamento de Ingeniería Matemática, Universidad de Concepción, Casilla 160-C, Concepción, Chile*

^b*CI²MA, Universidad de Concepción, Casilla 160-C, Concepción, Chile*

^c*Department de Matemàtiques, Universitat de València, Valencia, Spain*

Abstract

The Lax-Wendroff time discretization is an alternative method to the popular total variation diminishing Runge-Kutta time discretization of discontinuous Galerkin schemes for the numerical solution of hyperbolic conservation laws. The resulting fully discrete schemes are known as LWDG and RKDG methods, respectively. Although LWDG methods are in general more compact and efficient than RKDG methods of comparable order of accuracy, the formulation of LWDG methods involves the successive computation of exact flux derivatives. This procedure allows to construct schemes of arbitrary formal order of accuracy in space and time. A new approximation procedure avoids the computation of exact flux derivatives. The resulting approximate LWDG schemes, addressed as ALWDG schemes, are easier to implement than their original LWDG versions. Numerical results for the scalar and system cases in one and two space dimensions indicate that ALWDG methods are more efficient in terms of error reduction per CPU time than LWDG method of the same order of accuracy. Moreover, increasing the order of accuracy leads to substantial reductions of numerical error and gains in efficiency for solutions that vary smoothly.

Keywords: Discontinuous Galerkin scheme, Lax-Wendroff time discretization, systems of conservation laws

2000 MSC: 76S05, 65M08, 65M60, 65M12

1. Introduction

1.1. Scope

We are interested in high-order accurate numerical schemes for systems of conservation laws

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial}{\partial x_1} \mathbf{f}_1(\mathbf{u}) + \cdots + \frac{\partial}{\partial x_d} \mathbf{f}_d(\mathbf{u}) = \mathbf{0}, \quad \mathbf{x} = (x_1, \dots, x_d) \in \Omega \subset \mathbb{R}^d, \quad t > 0 \quad (1.1)$$

in d space dimensions, where t is time, \mathbf{x} is spatial position, $\mathbf{u} = (u_1, \dots, u_m)^\top$ is the sought vector of unknowns, and $\mathbf{f}_j(\mathbf{u}) = (f_{j,1}(\mathbf{u}), \dots, f_{j,m}(\mathbf{u}))^\top$ is the given flux vector in the j -th coordinate direction, $j = 1, \dots, d$. We assume that (1.1) is posed along with an initial condition

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \quad \mathbf{x} \in \Omega. \quad (1.2)$$

One particular case is that of an initial value problem for scalar conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \quad x \in \Omega \subset \mathbb{R}, \quad t > 0; \quad u(x, 0) = u_0(x), \quad x \in \Omega. \quad (1.3)$$

Email addresses: rburger@ing-mat.udec.cl (Raimund Bürger), skumar@ci2ma.udec.cl (Sudarshan Kumar Kenettinkara), david.zorio@uv.es (David Zorío)

It is well known that solutions of (1.1), (1.2) or (1.3) are in general discontinuous, which may occur even if the initial datum is smooth whenever the flux vector is a nonlinear function.

It is the purpose of this work to propose an approximation procedure for the Lax-Wendroff discontinuous Galerkin (LWDG) schemes applied to (1.1), (1.2) that were originally proposed by Qiu et al. [13]. The Lax-Wendroff time discretization is an alternative to the popular total variation diminishing Runge-Kutta time discretization of the discontinuous Galerkin scheme to solve hyperbolic conservation laws, resulting in the so-called RKDG method. As remarked in [13], the LWDG method is more compact and cost-efficient than the RKDG version of corresponding order of accuracy for certain problems, including the two-dimensional Euler system of compressible gas dynamics, when nonlinear limiters are applied. However, LWDG methods require the successive computation of exact flux derivatives or flux Jacobians in the case of systems. This is found to be a major drawback of LWDG methods even though they outperform RKDG methods in efficiency. We overcome this difficulty through an approximation procedure for the flux derivative or flux Jacobians, which is the main novelty of this proposed work. We abbreviate our method as ALWDG for approximate Lax-Wendroff discontinuous Galerkin scheme.

The approximation procedure is primarily motivated by recent results by Zorío et al. [25] that are obtained in the finite-difference setup. For the finite difference method it is sufficient to compute the approximate flux derivatives essentially at grid points, whereas the finite-volume-based DG method requires to compute the approximate derivatives in each cell. This difficulty is resolved by choosing the central differencing on non-uniform grids, which are essentially the available quadrature points in each cell. In the original LWDG scheme apart from the flux derivatives it involves the computation of the derivatives of approximated discontinuous Galerkin (DG) solution in each cell, which is found to be expensive. In our method we do not have to compute these higher derivatives, which in turn saves a considerable amount of CPU time as we see in the later discussions. With the proposed procedure, the original LWDG scheme is likely to become more practicable in usage.

1.2. Related work

Lax-Wendroff-type schemes have been widely used as an alternative to the Runge-Kutta-type time discretizations in the context of hyperbolic conservation laws. The original contribution [11] was a second-order scheme in that was extended for arbitrarily high order for a wide range of spatial schemes by several authors. Among others, this technique has been used in the case of Shu-Osher finite-difference spatial schemes by Qiu and Shu [14] and for discontinuous Galerkin schemes by Qiu et al. [13], followed by further improvements [8, 12]. See, e.g., [3, 4, 5, 6, 7] for details on standard RKDG schemes.

Lax-Wendroff-type schemes outperform Runge-Kutta-type schemes, in the sense that they only need one upwind phase for each time step, regardless of the accuracy order of the scheme, unlike the Runge-Kutta schemes, which need as many upwind phases as the number of stages of the scheme. Furthermore, this number increases nonlinearly with the order of the scheme. However, the implementation of Lax-Wendroff-type schemes may be complicated due to the necessity to compute the high-order partial derivative terms associated with systems of equations, since these expressions usually become very large and may require symbolic manipulation software to compute them, which ultimately yields severe performance drawbacks.

In order to overcome the aforementioned issue, Zorío et al. [25] propose a technique based on the approximation of the high-order terms to a suitable order of accuracy instead of computing the exact symbolic expression of the high order terms. This approximation is applied in [25] to Shu-Osher finite-difference schemes [19, 20]. It yields a more efficient and easy-to-implement scheme, where no symbolic computation is required to handle the high-order terms and no additional high-order terms have to be computed for different equations, namely, for different fluxes.

Furthermore, we note that the so-called TVD RK3 ODE solver achieves third-order accuracy with low storage requirements and a large CFL condition (see [19]). In contrast to this, for orders higher than three the stability properties are not so favorable, forcing either a smaller CFL restriction or inclusion of additional stages in the algorithm which essentially leads to loss in efficiency (for detailed discussion see [2]). These well-known difficulties underline the importance of the proposed ALWDG method.

1.3. Outline of the paper

The remainder of the paper is organized as follows. In Section 2 we briefly review the LWDG schemes. Specifically, after introducing preliminaries (Section 2.1), we develop in Section 2.2 the basic DG discretization for the scalar, one-dimensional problem (1.3). The key difficulty, namely the approximate integration over each spatial cell of a quantity that involves the unknown as well as its time derivatives, is handled in Section 2.3. This issue is resolved by approximating spatial derivatives through central differences defined on the non-uniform spatial grid defined by the nodes of the underlying quadrature formula (Section 2.4). In Section 3, which is at the core of this paper, we describe the new ALWDG method as a new time discretization of the DG scheme based in using an *approximate* Lax-Wendroff technique. The critical ingredient consists in generating approximations to higher time derivatives of the flux that are of certain determined orders of accuracy. This is done in Sections 3.1, 3.2 and 3.3 to generate schemes that are of third, fourth, and fifth order of accuracy, respectively. We then outline how to construct schemes of an arbitrary order of accuracy R . To this end we first recall, in Section 3.4, how central difference coefficients for a scheme of arbitrarily high order are obtained, and then, in Section 3.5, develop a time scheme of general order of accuracy (generalizing the treatment of Section 3.1–3.3). In Section 3.6 we address the extensions of the new method to systems of conservation laws and to several space dimensions, respectively, and summarize the ALWDG schemes in Section 3.7 for ease of reference. Section 4 is devoted to the presentation of numerical examples. It starts with a brief discussion of the CFL condition and the nonlinear limiters required for the implementation of LWDG and ALWDG schemes with non-smooth solutions (Section 4.1). We then consider, in Section 4.2, three examples with a smooth solution that allow us to verify whether the numerical schemes indeed have the advertised orders of accuracy. This is done in Examples 1, 2, 3, 4 and 3 for the one-dimensional inviscid Burgers equation, a one-dimensional equation with exponential flux, the one-dimensional Euler equations of gas dynamics, the two-dimensional inviscid Burgers equation, and a two-dimensional equation with exponential flux, respectively. On the other hand, we consider in Section 4.3 more realistic scenarios of equations (1.1) or (1.3), namely test cases that involve the formation of shocks. These include a scalar, one-dimensional Buckley-Leverett problem (Example 6), the one-dimensional Euler equations of gas dynamics (Examples 7 and 8), and the two-dimensional Euler equations of gas dynamics (Example 9). Finally, in Section 5 some conclusions are drawn.

2. Lax-Wendroff-discontinuous Galerkin (LWDG) schemes

2.1. Preliminaries

Let us consider first initial value problem for a scalar conservation law (1.3), assuming that $\Omega = [0, 1]$. We divide the domain Ω into cells $I_i = [x_{i-1/2}, x_{i+1/2}]$, where $0 = x_{1/2} < x_{3/2} < \dots < x_{N+1/2} = 1$, $x_i = (x_{i-1/2} + x_{i+1/2})/2$, $\Delta x_i := x_{i+1/2} - x_{i-1/2}$, and $h := \max_i \Delta x_i$. Let us assume that the mesh is regular, i.e., there is a constant $c > 0$ such that $\Delta x_i \geq ch$ for all i . Let us consider the following space of broken polynomials:

$$V_h^k := \{v_h \in L^2(\Omega) : v_h|_{I_i} \in P^k(I_i), 1 \leq i \leq N\}, \quad (2.1)$$

where $P^k(I_i)$ is the space of polynomials of degree less or equal k defined on I_i . Note that the elements of V_h^k can be discontinuous on the boundary of the intervals I_i . We define the left and right limits as

$$v_h(x^-) := \lim_{\varepsilon \searrow 0} v_h(x - \varepsilon), \quad v_h(x^+) := \lim_{\varepsilon \searrow 0} v_h(x + \varepsilon).$$

2.2. Discontinuous Galerkin discretization

Assume now that u is a smooth solution of (1.3). Then

$$\frac{\partial^k u}{\partial t^k} = -\frac{\partial}{\partial x} \left(\frac{\partial^{k-1} f(u)}{\partial t^{k-1}} \right).$$

We denote by indices x, t, xx etc., partial derivatives. Then the Taylor expansion of $u(x, t)$ yields

$$u(x, t + \Delta t) = u(x, t) - \Delta t f(u)_x \Big|_{(x,t)} - \frac{\Delta t^2}{2} (f(u)_t)_x \Big|_{(x,t)} - \frac{\Delta t^3}{6} (f(u)_{tt})_x \Big|_{(x,t)} + \mathcal{O}(\Delta t^4).$$

Just neglecting the $\mathcal{O}(\Delta t^4)$ term we get the expression

$$u(x, t + \Delta t) - u(x, t) + \Delta t F_x = 0, \quad \text{where} \quad F = F(u(x, t)) = f(u) + \frac{\Delta t}{2} f(u)_t + \frac{\Delta t^2}{6} f(u)_{tt}. \quad (2.2)$$

We adopt a local orthogonal basis on I_i ,

$$\{v_l^i(x), l = 0, 1, \dots, k\}, \quad v_l^i(x) = p_l \left(\frac{x - x_i}{\Delta x_i} \right), \quad (2.3)$$

where p_l are the Legendre polynomials which may be obtained from the recurrence relation

$$(n+1)p_{n+1}(\xi) = (2n+1)\xi p_n(\xi) - np_{n-1}(\xi), \quad n \in \mathbb{N}, \quad \text{where } p_0(\xi) := 1 \text{ and } p_1(\xi) := \xi.$$

Now the approximate solution $u^h = u^h(x, t) \in V_h^k$ can be written as

$$u^h(x, t) = \sum_{l=0}^k u_i^{(l)}(t) v_l^i(x) \quad \text{for } x \in I_i. \quad (2.4)$$

Multiplying (2.2) by a test function $\phi \in V_h^k$ and integrating the result over the interval I_i we get

$$\int_{I_i} u(x, t + \Delta t) \phi \, dx - \int_{I_i} u(x, t) \phi \, dx + \Delta t \int_{I_i} F(u(x, t))_x \phi \, dx = 0,$$

and the sought approximate solution $u^h \in V_h^k$ satisfies

$$\forall \phi \in V_h^k : \quad \int_{I_i} u^h(x, t + \Delta t) \phi \, dx - \int_{I_i} u^h(x, t) \phi \, dx + \Delta t \int_{I_i} F(u^h(x, t))_x \phi \, dx = 0. \quad (2.5)$$

Assume that the approximated solution u^h at time t^n is given. Our aim is to find the solution u^h at $t^{n+1} = t^n + \Delta t$. To this end we write (2.5) as

$$\int_{I_i} u^h(x, t^{n+1}) \phi \, dx - \int_{I_i} u^h(x, t^n) \phi \, dx + \Delta t \int_{I_i} F(u^h(x, t^n))_x \phi \, dx = 0.$$

Now choosing $\phi|_{I_i} = v_j^i$, using the expression (2.4) and applying integration by parts we get

$$\begin{aligned} & \sum_{l=0}^k u_i^{(l)}(t^{n+1}) \int_{I_i} v_l^i(x) v_j^i(x) \, dx - \sum_{l=0}^k u_i^{(l)}(t^n) \int_{I_i} v_l^i(x) v_j^i(x) \, dx - \Delta t \int_{I_i} F(u^h(x, t^n)) v_{j_x}^i \, dx \\ & + \Delta t \left(F(u^h(x, t^n)) v_j^i(x) \Big|_{x_{i+1/2}^-} - F(u^h(x, t^n)) v_j^i(x) \Big|_{x_{i-1/2}^+} \right) = 0, \end{aligned}$$

Replacing the boundary term by the numerical flux $\hat{F}_{i+1/2}$ we get

$$\begin{aligned} & \sum_{l=0}^k u_i^{(l)}(t^{n+1}) \int_{I_i} v_l^i(x) v_j^i(x) \, dx - \sum_{l=0}^k u_i^{(l)}(t^n) \int_{I_i} v_l^i(x) v_j^i(x) \, dx - \Delta t \int_{I_i} F(u^h(x, t^n)) v_{j_x}^i \, dx \\ & + \Delta t (\hat{F}_{i+1/2} v_j^i(x_{i+1/2}^-) - \hat{F}_{i-1/2} v_j^i(x_{i-1/2}^+)) = 0, \quad j = 0, 1, 2, \dots, k. \end{aligned}$$

Now choosing $j = 0, 1, 2, \dots, k$ in the above expression we obtain the following:

$$\mathbf{A}\mathbf{U}_i(t^{n+1}) = \mathbf{A}\mathbf{U}_i(t^n) + \Delta t \mathbf{B}_i - \Delta t \mathbf{C}_i, \quad i = 1, \dots, N,$$

where the approximate solution on cell I_i at time t^n is represented by the vector of coefficients for (2.4),

$$\mathbf{U}_i(t^n) = (u_i^{(0)}(t^n), u_i^{(1)}(t^n), \dots, u_i^{(k)}(t^n))^T,$$

and the entries of $\mathbf{A} = (a_{lj}) \in \mathbb{R}^{(k+1) \times (k+1)}$, $\mathbf{B}_i = (b_0, b_1, \dots, b_k)^T$ and $\mathbf{C}_i = (c_0, c_1, \dots, c_k)^T$ are given by

$$a_{lj} = \int_{I_i} v_l^i v_j^i dx, \quad b_j = \int_{I_i} F(u^h(x, t^n)) v_{j,x}^i dx, \quad c_j = \hat{F}_{i+1/2} v_j^i(x_{i+1/2}) - \hat{F}_{i-1/2} v_j^i(x_{i-1/2}). \quad (2.6)$$

Since we have an orthogonal basis (2.3), the matrix \mathbf{A} is diagonal and invertible, therefore the coefficients (degrees of freedom) \mathbf{U}_i at time t^{n+1} are given by

$$\mathbf{U}_i(t^{n+1}) = \mathbf{U}_i(t^n) + \Delta t \mathbf{A}^{-1} \mathbf{B}_i - \Delta t \mathbf{A}^{-1} \mathbf{C}_i, \quad i = 1, \dots, N.$$

The DG scheme is now complete provided we know how to deal with the function F and the numerical flux $\hat{F}_{i+1/2}$. A choice for the numerical flux is the Lax-Friedrichs flux (LF) or the so-called local Lax-Friedrichs flux (LLF). Both are defined by

$$\hat{F}_{i+1/2} := \frac{1}{2} \left(F(u^h(x_{i+1/2}^-, t^n)) + F(u^h(x_{i+1/2}^+, t^n)) - \kappa (u^h(x_{i+1/2}^+, t^n) - u^h(x_{i+1/2}^-, t^n)) \right), \quad (2.7)$$

where for LF flux, the constant $\kappa = \max_u |f'(u)|$ is taken as an upper bound on $|f'(u)|$ over the whole range of u in the scalar case, or as a bound on the spectral radius of the Jacobian for the system case. For the LLF flux, κ is locally determined by

$$\kappa = \max_{u \in J} |f'(u)|, \quad J := [\min\{u^h(x_{i+1/2}^-, t^n), u^h(x_{i+1/2}^+, t^n)\}, \max\{u^h(x_{i+1/2}^-, t^n), u^h(x_{i+1/2}^+, t^n)\}] \quad (2.8)$$

in the scalar case or by evaluating or estimating the corresponding spectral radii in the system case. The expression for $\hat{F}_{i-1/2}$ follows in a similar way. Note that the scheme possesses a spatial order of accuracy $\mathcal{O}(h^{k+1})$ when we use piecewise polynomials from P^k , i.e., of maximal degree k .

In two dimensions we use the polynomial space based on Legendre polynomials. In the reference domain $[-1, 1]^2$ the space P^k is spanned by the polynomials of the form $p_{ij}(x, y) = p_i(x)p_j(y)$, for $0 \leq i, j \leq k$, where p_i , $0 \leq i \leq k$ are the corresponding Legendre polynomials defined in the interval $[-1, 1]$. For further details about the implementation in two dimensions we refer to [4].

2.3. Flux approximation

The integral arising in b_j (see (2.6)) is computed approximately by a suitable quadrature rule, for example Gauss-Lobatto quadrature rule. This requires to evaluate the expression $F(u(x, t))$, which has to be computed at (x, t^n) for $x \in I_i$. We do not know the time derivative of u^h at t^n but we know the spatial derivative at (x, t^n) , so a complicated way to write $F(u^h(x, t^n))$ in the interval I_i is the following:

$$\begin{aligned} F(u^h(x, t^n)) &= f(u^h(x, t^n)) + \frac{\Delta t}{2} f'(u^h(x, t^n)) u_t^h(x, t^n) \\ &\quad + \frac{\Delta t^2}{6} \left(f''(u^h(x, t^n)) u_t^h(x, t^n)^2 + f'(u^h(x, t^n)) u_{tt}^h(x, t^n) \right), \end{aligned} \quad (2.9)$$

where we compute the time derivatives $u_t^h(x, t^n)$ and $u_{tt}^h(x, t^n)$ from the following relations:

$$u_t = -f'(u)u_x, \quad u_{tt} = -(f'(u)u)_t = -f''(u)u_x u_t - f'(u)u_{xt}, \quad u_{xt} = -f''(u)(u_x)^2 - f'(u)u_{xx}.$$

Note that in fact we here need to compute $F(u^h(x, t^n))$ only at the quadrature points.

So far we have discussed a third-order in time LWDG discretization, and (2.9) indicates that it is quite complicated. So to avoid this long expression for $F(u^h(x, t^n))$, we may replace the time derivatives $\partial^m f(u^h(x, t^n))/\partial t^m$ by a recurrence approximation of the form

$$\frac{\partial^m f(u^h)}{\partial t^m}(x, t^n) \approx \tilde{f}_{x,n}^{(m)}, \quad (2.10)$$

and in terms of these approximate time derivatives $f_{x,n}^{(m)}$ we can write

$$F(u^h(x, t^n)) \approx f(u^h(x, t^n)) + \sum_{m=1}^{\nu} \frac{\Delta t^m}{(m+1)!} \tilde{f}_{x,n}^{(m)}, \quad (2.11)$$

where ν specifies the order of accuracy in time.

2.4. Central difference approximations on non-uniform grids

To evaluate the term $\tilde{f}_{x,n}^{(m)}$ we need to compute successive derivatives of the flux functions or Jacobians in the case of system of equations. In our proposed scheme we resolve this additional difficulty through an approximation of the expression $\tilde{f}_{x,n}^{(m)}$ using central differencing primarily motivated by [25]. However, in contrast to [25] we utilize central differencing on non-uniform grids which we recall in the following lines. If we are given a function $\varphi : I \rightarrow \mathbb{R}$ and $\{x_q\}$ a set of quadrature points on the interval I , then the non-uniform central differencing approximation of $d\varphi/dx$ at the quadrature points reads as

$$\frac{d\varphi}{dx}\Big|_{x_q} \approx \alpha\varphi(x_{q-1}) + \beta\varphi(x_q) + \gamma\varphi(x_{q+1}),$$

where

$$\alpha = -\frac{h_2}{h_1(h_1+h_2)}, \quad \gamma = \frac{h_1}{h_2(h_1+h_2)}, \quad \beta = -\alpha - \gamma, \quad h_1 = x_q - x_{q-1}, \quad h_2 = x_{q+1} - x_q. \quad (2.12)$$

3. Approximate Lax-Wendroff discontinuous Galerkin (ALWDG) schemes

Let us denote by I_i as the i^{th} cell of the DG discretization. The primary goal is to approximate the following terms arising in (2.10):

$$\frac{\partial^m f(u(x, t))}{\partial t^m}\Big|_{\{x \in I_i, t=t_n\}}, \quad m = 1, \dots, \nu.$$

In the following development of the various high-order schemes we always focus on one cell I_i .

3.1. Third-order scheme

We now construct a third-order in time ALWDG scheme. In order to obtain third-order accuracy in time we need the approximation of f_t to have $\mathcal{O}(\Delta t^2)$ accuracy and that of f_{tt} to have $\mathcal{O}(\Delta t)$ accuracy as $\Delta t \rightarrow 0$.

We begin with the first term $f(u)_t$. For $x \in I_i$ this term can be approximated using central differencing in time. Here and elsewhere, we assume that the ratio $\lambda := \Delta t/\Delta x$ is fixed, and we write $\mathcal{O}(h^p)$ to denote $\mathcal{O}(\Delta t^p)$ or equivalently, $\mathcal{O}(\Delta x^p)$. Clearly,

$$\begin{aligned} f_t := (f(u)_t)(x, t_n) &= \frac{1}{2\Delta t} \left(f(u(x, t_n + \Delta t)) - f(u(x, t_n - \Delta t)) \right) + \mathcal{O}(h^2) \\ &= \frac{1}{2\Delta t} \left(f(u(x, t_n) + \Delta t u_t(x, t_n)) - f(u(x, t_n) - \Delta t u_t(x, t_n)) \right) + \mathcal{O}(h^2) \\ &= \frac{1}{2\Delta t} \left(f(\tilde{u}_n^{(0)}(x) + \Delta t \tilde{u}_n^{(1)}(x)) - f(\tilde{u}_n^{(0)}(x) - \Delta t \tilde{u}_n^{(1)}(x)) \right) + \mathcal{O}(h^2), \end{aligned}$$

where $\tilde{u}_n^{(0)}$ and $\tilde{u}_n^{(1)}$ correspond to the polynomial DG solution and the approximation of u_t in the cell I_i , respectively. We write

$$f_t \approx \tilde{f}_t := \frac{1}{2\Delta t} \left(f(\tilde{u}_n^{(0)}(x) + \Delta t \tilde{u}_n^{(1)}(x)) - f(\tilde{u}_n^{(0)}(x) - \Delta t \tilde{u}_n^{(1)}(x)) \right).$$

In order to evaluate $\tilde{u}_n^{(1)}$, we can use the expression

$$u_t = -f(u)_x \quad (= -f'(u)u_x) \quad (3.1)$$

and compute the exact term

$$\tilde{u}_n^{(1)}(x) = u_t(x, t_n) \quad \text{for } x \in I_i. \quad (3.2)$$

In fact, we could use an approximation as well, which we see later. Now we consider the second term $f(u)_{tt}$. We use the following central difference again:

$$\begin{aligned} f_{tt} &:= f(u)_{tt}(x, t_n) = \frac{1}{\Delta t^2} \left(f(u(x, t_n + \Delta t)) - 2f(u(x, t_n)) + f(u(x, t_n - \Delta t)) \right) + \mathcal{O}(h^2) \\ &= \frac{1}{\Delta t^2} \left[f \left(u(x, t_n) + \Delta t u_t(x, t_n) + \frac{\Delta t^2}{2} u_{tt}^i(x, t_n) \right) - 2f(u(x, t_n)) \right. \\ &\quad \left. + f \left(u(x, t_n) - \Delta t u_t(x, t_n) + \frac{\Delta t^2}{2} u_{tt}(x, t_n) \right) \right] + \mathcal{O}(h^2) \\ &= \frac{1}{\Delta t^2} \left[f \left(\tilde{u}_n^{(0)}(x) + \Delta t \tilde{u}_n^{(1)}(x) + \frac{\Delta t^2}{2} \tilde{u}_n^{(2)}(x) \right) - 2f(\tilde{u}_n^{(0)}(x)) \right. \\ &\quad \left. + f \left(\tilde{u}_n^{(0)}(x) - \Delta t \tilde{u}_n^{(1)}(x) + \frac{\Delta t^2}{2} \tilde{u}_n^{(2)}(x) \right) \right] + \mathcal{O}(h^2), \end{aligned}$$

so that we arrive at the following approximation of f_{tt} :

$$\begin{aligned} f_{tt} \approx \tilde{f}_{tt}(x, t_n) &:= \frac{1}{\Delta t^2} \left[f \left(\tilde{u}_n^{(0)}(x) + \Delta t \tilde{u}_n^{(1)}(x) + \frac{\Delta t^2}{2} \tilde{u}_n^{(2)}(x) \right) - 2f(\tilde{u}_n^{(0)}(x)) \right. \\ &\quad \left. + f \left(\tilde{u}_n^{(0)}(x) - \Delta t \tilde{u}_n^{(1)}(x) + \frac{\Delta t^2}{2} \tilde{u}_n^{(2)}(x) \right) \right], \end{aligned} \quad (3.3)$$

where $\tilde{u}_n^{(2)}(x)$ is an approximation of $u_{tt}(x, t_n)$, which we evaluate using the following central difference:

$$u_{tt}(x, t_n) = -(f(u(x, t_n)))_t)_x \approx -(\tilde{f}_t(x))_x. \quad (3.4)$$

Note that we only need to compute $\tilde{u}_n^{(2)}(x_q)$, where x_q are the quadrature points in the cell I_i . Since we usually use non-equidistant Gauss or Gauss-Lobatto quadrature points, we need to compute the central differencing on non-uniform grids. On non-uniform grids we have

$$(\tilde{f}_t(x_q))_x = \alpha \tilde{f}_t(x_{q-1}) + \beta \tilde{f}_t(x_q) + \gamma \tilde{f}_t(x_{q+1}) + \mathcal{O}(h^2),$$

where α , β and γ are given in (2.12), so from (3.4) we get the approximation

$$u_{tt}(x_q, t_n) \approx \tilde{u}_n^{(2)}(x_q) = -(\alpha \tilde{f}_t(x_{q-1}) + \beta \tilde{f}_t(x_q) + \gamma \tilde{f}_t(x_{q+1})).$$

We need to consider the boundary points of the cell I_i as we deal with the Gauss-Lobatto quadrature points. Therefore to compute $\tilde{u}_n^{(2)}(x)$ at the boundary we can use a two-step forward or two-step backward approximation. Now the method is complete for the third-order scheme.

Remark 3.1. *If the third derivative d^3f/du^3 vanishes, i.e., f is a polynomial of degree less or equal 2, then it is easy to see that the approximation \tilde{f}_t of $f(u)_t$ coincides with the exact derivative $f(u)_t$. In that case the term $F(u) = f(u) + (\Delta t/2)f(u)_t$ is the same for both the approximate and exact versions. Consequently, the second-order versions of both ALWDG and LWDG schemes essentially are the same. As an example, we observe this in the case of inviscid Burgers equation (Examples 1 and 4 in Section 4).*

3.2. Fourth-order scheme

To obtain fourth-order accuracy in time we need to approximate the terms f_t , f_{tt} and f_{ttt} in (2.10) to the respective accuracies $\mathcal{O}(\Delta t^3)$, $\mathcal{O}(\Delta t^2)$ and $f_{ttt} = \mathcal{O}(\Delta t)$. Therefore we begin with a fourth-order approximation of f_t :

$$\begin{aligned} f_t := f(u)_t(x, t_n) &= \frac{1}{12\Delta t} \left(-f(u(x, t_n + 2\Delta t)) + 8f(u(x, t_n + \Delta t)) \right. \\ &\quad \left. - 8f(u(x, t_n - \Delta t)) + f(u(x, t_n - 2\Delta t)) \right) + \mathcal{O}(h^4) \\ &= \frac{1}{12\Delta t} \left(-\varphi_{n,2\Delta t}^1(x) + 8\varphi_{n,\Delta t}^1(x) - 8\varphi_{n,-\Delta t}^1(x) + \varphi_{n,-2\Delta t}^1(x) \right) + \mathcal{O}(h^4), \end{aligned}$$

where we define

$$\varphi_{n,\delta}^1(x) := f(\tilde{u}_n^{(0)}(x) + \delta\tilde{u}_n^{(1)}(x)). \quad (3.5)$$

As in the third-order case here we can compute $\tilde{u}_n^{(1)}$ by (3.1), (3.2). Consequently, for the fourth-order scheme we utilize the following approximation of f_t :

$$f_t \approx \tilde{f}_t(x, t_n) := \frac{1}{12\Delta t} \left(-\varphi_{n,2\Delta t}^1(x) + 8\varphi_{n,\Delta t}^1(x) - 8\varphi_{n,-\Delta t}^1(x) + \varphi_{n,-2\Delta t}^1(x) \right). \quad (3.6)$$

Moreover, since the order of approximation in f_{tt} only needs to be $\mathcal{O}(\Delta t^2)$, we may approximate f_{tt} by (3.3) (as for the third-order scheme). Next, in order to approximate $f(u)_{ttt}(x, t_n)$ we note that

$$\begin{aligned} f_{ttt} := f(u)_{ttt}(x, t_n) &= \frac{1}{2\Delta t^3} \left(f(u(x, t_n + 2\Delta t)) - 2f(u(x, t_n + \Delta t)) \right. \\ &\quad \left. + 2f(u(x, t_n - \Delta t)) - f(u(x, t_n - 2\Delta t)) \right) + \mathcal{O}(h^2) \\ &= \frac{1}{2\Delta t^3} \left(\varphi_{n,2\Delta t}^3(x) - 2\varphi_{n,\Delta t}^3(x) + 2\varphi_{n,-\Delta t}^3(x) - \varphi_{n,-2\Delta t}^3(x) \right) + \mathcal{O}(h^2), \end{aligned}$$

where we define

$$\varphi_{n,\delta}^3(x) := f \left(\tilde{u}_n^{(0)}(x) + \delta\tilde{u}_n^{(1)}(x) + \frac{\delta^2}{2!}\tilde{u}_n^{(2)}(x) + \frac{\delta^3}{3!}\tilde{u}_n^{(3)}(x) \right). \quad (3.7)$$

Thus, the approximation of f_{ttt} for the fourth-order scheme is now given by

$$f_{ttt} \approx \tilde{f}_{ttt}(x, t_n) := \frac{1}{2\Delta t^3} \left(\varphi_{n,2\Delta t}^3(x) - 2\varphi_{n,\Delta t}^3(x) + 2\varphi_{n,-\Delta t}^3(x) - \varphi_{n,-2\Delta t}^3(x) \right). \quad (3.8)$$

Here $\tilde{u}_n^{(3)}(x)$ is an approximation of $u_{ttt}(x, t_n)$, which we evaluate using a central differencing for

$$u_{ttt}(x, t_n) = -\left(f(u(x, t_n))_{tt} \right)_x \approx -\left(\tilde{f}_{tt}(x) \right)_x.$$

Note that we only need to compute $\tilde{u}_n^{(3)}(x_q)$, where x_q are the quadrature points in the cell I_i . Once again, since we usually use non-equidistant Gauss or Gauss-Lobatto quadrature points, we need to compute the central differencing on non-uniform grids. We have

$$\left(\tilde{f}_{tt}(x_q) \right)_x = \alpha\tilde{f}_{tt}(x_{q-1}) + \beta\tilde{f}_{tt}(x_q) + \gamma\tilde{f}_{tt}(x_{q+1}) + \mathcal{O}(h^2),$$

where α , β and γ are given in (2.12). This yields the approximation

$$u_{ttt}(x_q, t_n) \approx \tilde{u}_n^{(3)}(x_q) := -\left(\alpha\tilde{f}_{tt}(x_{q-1}) + \beta\tilde{f}_{tt}(x_q) + \gamma\tilde{f}_{tt}(x_{q+1}) \right).$$

As for the third-order scheme, to compute $\tilde{u}_n^{(2)}(x)$ at the boundary we can use a two-step forward or two-step backward approximation. Now the method is complete for the fourth-order scheme.

3.3. Fifth-order scheme

To obtain fifth-order accuracy in time we need to approximate the terms f_t , f_{tt} , f_{ttt} and f_{tttt} arising in (2.10) to the respective orders of accuracy $\mathcal{O}(\Delta t^4)$, $\mathcal{O}(\Delta t^3)$, $\mathcal{O}(\Delta t^2)$, and $\mathcal{O}(\Delta t)$. Since the approximation \tilde{f}_t of f_t defined by (3.5), (3.6) for the fourth-order scheme does already have the required order of accuracy, we use the same expression also for the fifth-order scheme. Next, we calculate that

$$\begin{aligned} f_{tt} &= \frac{1}{12\Delta t^2} \left(-f(u(x, t_n + 2\Delta t)) + 16f(u(x, t_n + \Delta t)) - 30f(u(x, t_n)) \right. \\ &\quad \left. + 16f(u(x, t_n - \Delta t)) - f(u(x, t_n - 2\Delta t)) \right) + \mathcal{O}(h^4) \\ &= \frac{1}{12\Delta t^2} \left(-\varphi_{n,2\Delta t}^2(x) + 16\varphi_{n,\Delta t}^2(x) - 30\varphi_{n,0}^2(x) + 16\varphi_{n,-\Delta t}^2(x) - \varphi_{n,-2\Delta t}^2(x) \right) + \mathcal{O}(h^4), \end{aligned}$$

where we define

$$\varphi_{n,\delta}^2(x) := f \left(\tilde{u}_n^{(0)}(x) + \delta \tilde{u}_n^{(1)}(x) + \frac{\delta^2}{2!} \tilde{u}_n^{(2)}(x) \right). \quad (3.9)$$

Here $\tilde{u}_n^{(2)}(x)$ is an approximation of $u_{tt}(x, t_n)$ that we evaluate using (3.4). Since a fourth-order approximation of f_{tx} on a non-uniform grid is given by

$$\left(\tilde{f}_t(x_q) \right)_x = \alpha \tilde{f}_t(x_{q-2}) + \beta \tilde{f}_t(x_{q-1}) + \gamma \tilde{f}_t(x_q) + \delta \tilde{f}_t(x_{q+1}) + \mathcal{O}(h^4)$$

for the coefficients

$$\begin{aligned} \delta &= \frac{h_1 h_2}{h_3(h_1 + h_3)(h_3 + h_2)}, \quad \beta = -\frac{h_1 h_3}{h_2(h_3 + h_2)(h_1 - h_2)}, \quad \alpha = \frac{-\beta h_2 + \delta h_3 - 1}{h_1}, \quad \gamma = -\alpha - \beta - \delta, \\ h_1 &= x_q - x_{q-2}, \quad h_2 = x_q - x_{q-1}, \quad h_3 = x_{q+1} - x_q, \end{aligned}$$

we get the approximation

$$\tilde{u}_n^{(2)}(x_q) = -(\alpha \tilde{f}_t(x_{q-2}) + \beta \tilde{f}_t(x_{q-1}) + \gamma \tilde{f}_t(x_q) + \delta \tilde{f}_t(x_{q+1}))$$

to be used in (3.9). Thus, we employ the following approximation of f_{tt} within the fifth-order scheme:

$$f_{tt} \approx \tilde{f}_{tt}(x, t_n) := \frac{1}{12\Delta t^2} \left(-\varphi_{n,2\Delta t}^2(x) + 16\varphi_{n,\Delta t}^2(x) - 30\varphi_{n,0}^2(x) + 16\varphi_{n,-\Delta t}^2(x) - \varphi_{n,-2\Delta t}^2(x) \right).$$

Furthermore, we observe that the approximation \tilde{f}_{ttt} of f_{ttt} defined by (3.7), (3.8) for the fourth-order scheme does already have the required order of accuracy, we use the same expression also for the fifth-order scheme. Finally, the sought approximation of $f_{tttt} := f(u)_{tttt}(x, t_n)$ is obtained from

$$\begin{aligned} f_{tttt} &= \frac{1}{\Delta t^4} \left(f(u(x, t_n + 2\Delta t)) - 4f(u(x, t_n + \Delta t)) + 6f(u(x, t_n)) \right. \\ &\quad \left. - 4f(u(x, t_n - \Delta t)) + f(u(x, t_n - 2\Delta t)) \right) + \mathcal{O}(h^2) \\ &= \frac{1}{\Delta t^4} \left(\varphi_{n,2\Delta t}^4(x) - 4\varphi_{n,\Delta t}^4(x) + 6\varphi_{n,0}^4(x) - 4\varphi_{n,-\Delta t}^4(x) + \varphi_{n,-2\Delta t}^4(x) \right) + \mathcal{O}(h^2), \end{aligned}$$

where we define

$$\varphi_{n,\delta}^4(x) := f \left(\tilde{u}_n^{(0)}(x) + \delta \tilde{u}_n^{(1)}(x) + \frac{\delta^2}{2!} \tilde{u}_n^{(2)}(x) + \frac{\delta^3}{3!} \tilde{u}_n^{(3)}(x) + \frac{\delta^4}{4!} \tilde{u}_n^{(4)}(x) \right).$$

The quantity $\tilde{u}_n^{(4)}(x_q)$ is defined by

$$\tilde{u}_n^{(4)}(x_q) := -(\alpha \tilde{f}_{ttt}(x_{q-1}) + \beta \tilde{f}_{ttt}(x_q) + \gamma \tilde{f}_{ttt}(x_{q+1})),$$

where α , β and γ are given in (2.12). We therefore arrive at the approximation

$$\tilde{f}_{tttt}(x, t_n) = \frac{1}{\Delta t^4} \left(\varphi_{n,2\Delta t}^4(x) - 4\varphi_{n,\Delta t}^4(x) + 6\varphi_{n,0}^4(x) - 4\varphi_{n,-\Delta t}^4(x) + \varphi_{n,-2\Delta t}^4(x) \right).$$

The fifth-order scheme is now complete.

3.4. Obtention of central difference coefficients for an arbitrarily high-order scheme

Assume that we wish to construct an R -th order accurate time scheme. Let $r = \lceil R/2 \rceil$ and then consider p_{2r} the $2r$ -th degree interpolating polynomial satisfying $p_{2r}(i\Delta t) = u_i$, $-r \leq i \leq r$. Then p_{2r} can be univocally determined and the s -th derivative in the central point, $x = 0$, can be written as

$$p_{2r}^{(s)}(0) = \frac{1}{\Delta t^s} \sum_{i=-r}^r \beta_{i,s} u_i$$

for certain coefficients $\beta_{i,s}$, $-r \leq i \leq r$, which can be explicitly computed. Proposition 2 of [25] also describes an alternative (and constructive as well) procedure to find these coefficients.

3.5. Time scheme of general accuracy order

With all the aforementioned ingredients, the time scheme of generalized order R , assuming the spatial accuracy is greater or equal than R , can be computed following the steps indicated below.

1. For $k = 1, \dots, R-1$ we perform the steps 2 and 3 indicated below.
2. Assuming that the corresponding approximations of $u^{(l)}$ are known for $1 \leq l \leq k$, compute an approximation of $f^{(k)}$ using the suitable central difference operator, which can be explicitly obtained as described in Section 3.4. The optimal required accuracy order in this case for such centered difference is $2\lceil \frac{R-k}{2} \rceil$, which can be expressed as follows:

$$\tilde{f}_{x,n}^{(k)} = \frac{1}{\Delta t^k} \sum_{l=0}^s \beta_l^{k, \lceil \frac{R-k}{2} \rceil} \left(f(T_{x,n}^k(l\Delta t)) + (-1)^k f(T_{x,n}^k(-l\Delta t)) \right)$$

with

$$s := \left\lfloor \frac{k-1}{2} \right\rfloor + \left\lfloor \frac{R-k}{2} \right\rfloor \quad \text{and} \quad T_{x,n}^k(\rho) := \sum_{j=0}^k \frac{\rho^j}{j!} \tilde{u}_{x,n}^{(j)} \approx u(x, t_n + \rho),$$

where the coefficients $\beta_{-s}^{p,q}, \dots, \beta_s^{p,q}$ can be computed as shown in Section 3.4 or by solving the $(2s+1) \times (2s+1)$ system of equations suggested in Proposition 2 of [25]:

$$\sum_{l=-s}^s \beta_l^{p,q} l^r = \delta_r^p p!, \quad 0 \leq r \leq 2s, \quad \delta_r^p = \begin{cases} 1 & \text{if } r = p, \\ 0 & \text{if } r \neq p. \end{cases}$$

3. If $k < R-1$, we need to compute $u^{(k+1)}$, which is performed using that $u^{(k+1)} = -(f^{(k)})_x$, where central differencing on non-uniform grids is used through the known Gauss or Gauss-Lobatto quadrature points, provided the accuracy order of such operation is greater or equal than $2\lceil \frac{R-k}{2} \rceil$. To be more precise, if $x_{q-s}, \dots, x_{q+s+\rho}$ are the quadrature points of the current cell I_i , with $\rho \in \{-1, 0, 1\}$, $u^{(k+1)}(x_q)$ can be approximated through the expression

$$u^{(k+1)}(x_q) \approx - \sum_{k=-s}^{s+\rho} \alpha_k \tilde{f}^{(k)}(x_{q+k}),$$

where $\alpha_{-s}, \dots, \alpha_{s+\rho}$ are determined by solving the $(N+1) \times (N+1)$ ($N = 2s + \rho$) system of equations

$$\sum_{k=-s}^{s+\rho} \alpha_k h_{q+k}^j = \delta_j^1, \quad 0 \leq j \leq N, \quad h_p = x_p - x_q.$$

4. Once $f^{(k)}$ is computed for $1 \leq k \leq R-1$, the term $F(u^h(x, t^n))$ is approximated by

$$F(u^h(x, t^n)) \approx f(u^h(x, t^n)) + \sum_{m=1}^{\nu} \frac{\Delta t^m}{(m+1)!} \tilde{f}_{x,n}^{(m)}.$$

One can prove that a general R -th order accurate ALWDG scheme is actually R -th order accurate in terms of the global error. To prove the accuracy involving $f(u)^{(r)}$ it suffices to use the techniques of the proof of Proposition 1 in [25], corresponding to the finite-difference case, taking into account that the spatial variable remains static for the computation of the time derivative approximations. As for the proof involving spatial derivatives, assuming the spatial accuracy is greater or equal than R , using the approximations involving the spatial derivatives plus the results obtained in Proposition 1 of [25] concludes our result.

3.6. System and multi-dimensional cases

In the case of a system of conservation laws (1.1) with $m \geq 2$ there are no additional difficulties for the computation. Assume that we consider one space dimension ($d = 1$) and set $\mathbf{f} = \mathbf{f}_1$. One can then work in a component-wise manner by taking into account that by definition

$$\frac{\partial^n \mathbf{f}(\mathbf{u})}{\partial t^n} = \frac{\partial^n (\mathbf{f}(\mathbf{u})_1, \dots, \mathbf{f}(\mathbf{u})_m)^T}{\partial t^n} = \left(\frac{\partial^n f_1(\mathbf{u})}{\partial t^n}, \dots, \frac{\partial^n f_m(\mathbf{u})}{\partial t^n} \right)^T$$

and that

$$\frac{\partial^{n+1} \mathbf{u}}{\partial t^{n+1}} = - \frac{\partial}{\partial x} \left(\frac{\partial^n \mathbf{f}(\mathbf{u})}{\partial t^n} \right),$$

which yields a component-wise relationship that allows one to reduce the procedure to the scalar case:

$$\frac{\partial^{n+1} u_\mu}{\partial t^{n+1}} = - \frac{\partial}{\partial x} \left(\frac{\partial^n \mathbf{f}(\mathbf{u})_\mu}{\partial t^n} \right), \quad 1 \leq \mu \leq m.$$

The extension to several space dimensions is straightforward by using the following relationship

$$\mathbf{u}^{(r+1)} = (u_t)^{(r)} = (-\nabla \cdot \mathbf{f}(u))^{(r)} = -\nabla \cdot (\mathbf{f}(u)^{(r)}).$$

3.7. Summary

The LWDG method originally consists of two parts, the Lax-Wendroff time discretization of the system of equations (1.1) followed by the DG approximation in space of the corresponding time discretized equation (2.2). This is precisely a forward Euler time discretization of the discontinuous Galerkin scheme with the modified flux function F given by (2.11). The modified flux function F involves the successive time derivatives of the original flux function $f(u)$ of (1.1). This could be computed explicitly as in LWDG [13]. In the new ALWDG method we compute the time derivatives by the approximation explained in Section 2. At each time step and as is usually done within LWDG or RKDG methods, we employ a nonlinear limiter, which is essentially the TVB limiter, see [3, 5] for more details.

4. Numerical examples

4.1. CFL condition and limiters

In this section we present the results of our numerical experiments for the ALWDG, LWDG and RKDG schemes. To be concise we mainly conduct simulations for P^1 and P^2 (see (2.1)) and in some cases we include the results for P^3 and P^4 as well. We choose a uniform mesh in the 1D case and Cartesian grids in the 2D case. As remarked in [13], the CFL number for the LWDG method is smaller than that of the corresponding RKDG method. Nevertheless the CPU time for the LWDG scheme wins over that of the RKDG scheme. We remark that the ALWDG method works satisfactorily under the same CFL condition as the underlying LWDG scheme when nonlinear limiters are applied. For the 2D Euler system of equations the ALWDG scheme is implemented with the same CFL numbers as that of [13], namely 0.2 and 0.12 for P^1 and P^2 respectively. Where ever not specified we have used the same CFL of LWDG found in [13]. In two dimensions, the numerical integrations in the DG formulation are performed with

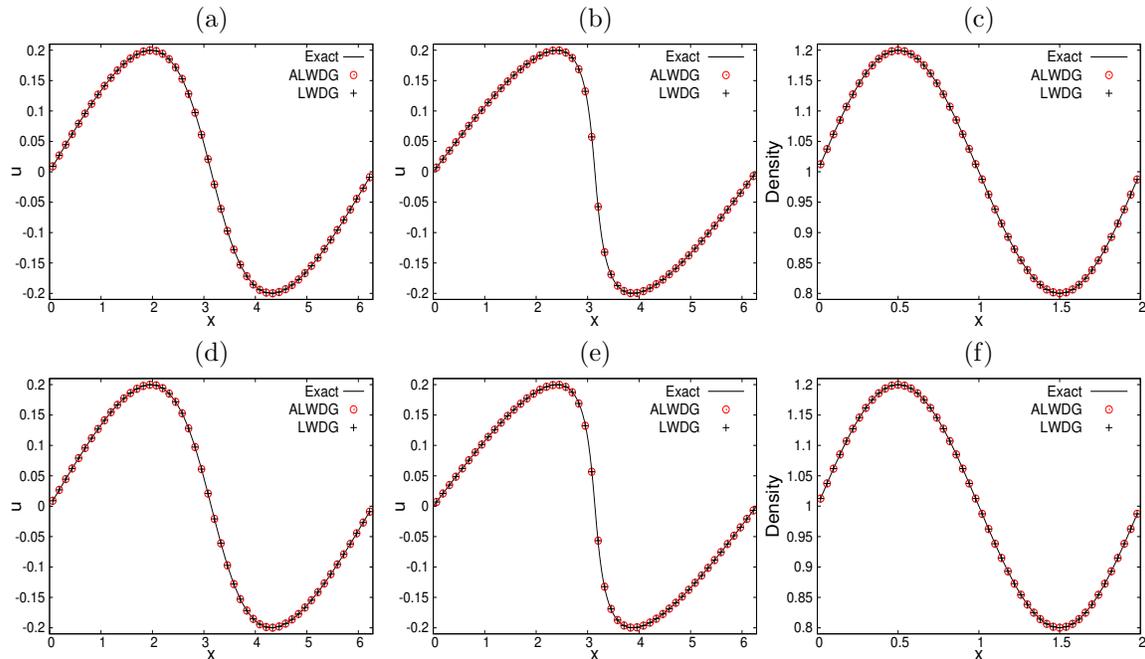


Figure 1: (a, c) Example 1 (1D inviscid Burgers equation with smooth solution), (b, d) Example 2 (1D equation with exponential flux with smooth solution), (e, f) Example 3 (1D Euler equations of gas dynamics, smooth solution, ρ -component): numerical solutions with $N = 50$ at time $t = 2$ obtained by the ALWDG and LWDG methods with (a, b, c) $k = 1$, (d, e, f) $k = 2$.

Gauss quadrature rule on $k + 1$ points for P^k elements and in one dimension we have used Gauss-Lobatto quadrature rule on $k + 2$ points. In the discontinuous test cases both in one two dimensions we apply a TVB limiter, more details are available in [15, 17, 18, 21], and the TVB constant M is specified in the description of examples. We have used the local Lax-Friedrichs (LLF) flux (2.7), (2.8) in all examples, more details are available in [16]. All the schemes described in this work are implemented in C++ using deal.II libraries [1]. In all examples, for 1D case we print the cell averages and for 2D case we plot the whole cell polynomials.

4.2. Accuracy tests

We test the accuracy of the ALWDG method for nonlinear scalar equations and a system of equations in both one and two dimensions. Results are compared with those of the LWDG scheme. In two dimensions we limit our presentation to the scalar case. No limiters are employed for these accuracy tests. To validate the accuracy results further, we provide graphical results as well, see Figures 1 and 2. The computational efficiency of ALWDG is compared with LWDG method and the results are listed in Figure 3. CPU time is plotted against the L^1 error, where we use the same mesh size for both ALWDG and LWDG. As we see in the results the ALWDG outperforms the LWDG in computational efficiency for these problems with smooth solutions.

Example 1: inviscid Burgers equation in one dimension

We conducted an accuracy test with the initial datum $u_0(x) = 0.2 \sin x$, assuming periodic boundary conditions for the interval $[0, 2\pi]$ and the solution is computed at time $t = 2$, see Figures 1 (a) and (c). Table 1 shows the results obtained for ALWDG schemes of second to and fifth order. In light of Remark 3.1, in this case the ALWDG and LWDG schemes produce the same numerical solutions.

# cells	# dofs	L^1 error	rate	L^∞ error	rate	# cells	# dofs	L^1 error	rate	L^∞ error	rate
ALWDG, $k = 1$						ALWDG, $k = 2$					
50	100	8.523927e-04	—	9.043832e-04	—	50	150	1.516174e-05	—	2.753776e-05	—
100	200	2.114508e-04	2.01	2.415103e-04	1.90	100	300	1.849973e-06	3.03	4.300787e-06	2.68
200	400	5.267018e-05	2.01	6.221746e-05	1.96	200	600	2.270759e-07	3.03	6.246012e-07	2.78
400	800	1.314472e-05	2.00	1.580492e-05	1.98	400	1200	2.803340e-08	3.02	8.543263e-08	2.87
800	1600	3.283470e-06	2.00	3.981910e-06	1.99	800	2400	3.473868e-09	3.01	1.129223e-08	2.92
1600	3200	2.050932e-07	2.00	9.993896e-07	1.99	1600	4800	4.313430e-10	3.01	1.461658e-09	2.95
3200	6400	2.051034e-07	2.00	2.503326e-07	2.00	3200	9600	5.373437e-11	3.00	1.868241e-10	2.97
ALWDG, $k = 3$						ALWDG, $k = 4$					
25	100	4.861064e-06	-	1.195543e-05	-	10	50	1.344846e-05	—	2.741170e-05	—
50	200	3.172026e-07	3.94	8.961908e-07	3.74	20	100	8.562981e-07	3.97	1.355695e-06	4.34
100	400	1.956276e-08	4.02	6.704435e-08	3.74	40	200	2.667358e-08	5.00	9.273087e-08	3.87
200	800	1.210728e-09	4.01	4.644211e-09	3.85	80	400	8.741969e-10	4.93	3.233760e-09	4.84
400	1600	7.514776e-11	4.01	3.075426e-10	3.92	160	800	2.680772e-11	5.03	1.334094e-10	4.60
800	3200	4.693472e-12	4.00	1.978160e-11	3.96	320	1600	8.607957e-13	4.96	4.823554e-12	4.79

Table 1: Example 1 (1D inviscid Burgers equation, smooth solution): errors of numerical solutions produced by ALWDG of orders 2, 3, 4 and 5 (corresponding to $k = 1, 2, 3$ and 4, respectively) computed up to the final time $t = 2.0$.

Example 2: scalar equation with exponential flux in one dimension

In Example 2 we choose the same interval and initial and boundary conditions as in Example 1 but use the exponential flux $f(u) = \exp(u^2)$, for which none of the ALWDG schemes coincides with its LWDG version. The corresponding numerical solution at $t = 2$ is displayed in Figures 1 (b) and (d), and the numerical errors are given in Table 2.

Example 3: Euler equations of gas dynamics in one space dimension

The one-dimensional Euler equations of gas dynamics are a case of (1.1) for $d = 1$ and $m = 3$, namely

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho v \\ E \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho v \\ p + \rho v^2 \\ (E + p)v \end{pmatrix} = \mathbf{0}. \quad (4.1)$$

Here ρ , v , E , and p denote the density, velocity, total energy, and pressure of the gas, respectively, and the system (4.1) is supplied with an equation of state $E = E(\rho, p, v)$, for which we choose the usual expression for a polytropic gas

$$E = E(p, \rho, v) = \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2, \quad \gamma > 1, \quad (4.2)$$

where γ denotes the adiabatic constant that is chosen as $\gamma = 1.4$ for air.

For the accuracy test (Example 3) we choose the domain $\Omega = [0, 2]$ along with periodic boundary conditions, and impose the initial datum

$$\mathbf{u}(x, 0) = \mathbf{u}_0(x) = (\rho_0(x), \rho_0(x)v_0(x), E(p_0(x), \rho_0(x), v_0(x)))^T, \quad (4.3)$$

where $\rho_0(x) = 1 + 0.2 \sin(\pi x)$, $v_0(x) = 1$, $p_0(x) = 1$.

The corresponding exact solution under periodic boundary conditions is $\rho(x, t) = \rho_0(x - t)$, $v(x, t) = 1$ and $p(x, t) = 1$. The results are tabulated in Table 3 and 4. Necessary details are indicated in the caption of each table.

# cells	# dofs	L^1 error	rate	L^∞ error	rate	L^1 error	rate	L^∞ error	rate
ALWDG, $k = 1$					LWDG, $k = 1$				
50	100	2.128317e-03	-	6.962886e-03	-	2.128310e-03	-	6.962790e-03	-
100	200	5.066706e-04	2.07	2.179635e-03	1.68	5.066700e-04	2.07	2.179626e-03	1.68
200	400	1.248754e-04	2.02	6.249265e-04	1.80	1.248753e-04	2.02	6.249252e-04	1.80
400	800	3.108018e-05	2.01	1.678809e-04	1.90	3.108018e-05	2.01	1.678808e-04	1.90
800	1600	7.758670e-06	2.00	4.345228e-05	1.95	7.758670e-06	2.00	4.345226e-05	1.95
1600	3200	1.938548e-06	2.00	1.103856e-05	1.98	1.938548e-06	2.00	1.103856e-05	1.98
3200	6400	4.845149e-07	2.00	2.780704e-06	1.99	4.845149e-07	2.00	2.780704e-06	1.99
ALWDG, $k = 2$					LWDG, $k = 2$				
50	150	1.188253e-04	-	5.995293e-04	-	1.188505e-04	-	5.971530e-04	-
100	300	1.985605e-05	2.58	1.556278e-04	1.95	1.985356e-05	2.58	1.554883e-04	1.94
200	600	2.478049e-06	3.00	2.422379e-05	2.68	2.478171e-06	3.00	2.421789e-05	2.68
400	1200	3.168454e-07	2.97	3.687378e-06	2.72	3.168485e-07	2.97	3.688288e-06	2.72
800	2400	3.943921e-08	3.01	5.512340e-07	2.74	3.943915e-08	3.01	5.512973e-07	2.74
1600	4800	4.910838e-09	3.01	7.883687e-08	2.81	4.910851e-09	3.01	7.883993e-08	2.81
3200	9600	6.081055e-10	3.01	1.075938e-08	2.87	6.081063e-10	3.01	1.075921e-08	2.87

Table 2: Example 2 (1D scalar equation with exponential flux and smooth solution): comparison of ALWDG with LWDG for second and third-order discretization ($k = 1$ and $k = 2$, respectively), at simulated time $t = 2.0$.

Example 4: inviscid Burgers equation in two space dimensions

In order to draw the order of accuracy in the two dimensional set up, we consider the initial-value problem for the nonlinear scalar Burgers-type equation

$$u_t + (u^2/2)_x + (u^2/2)_y = h(x, y, t), \quad (x, y) \in [-1, 1]^2, \quad t > 0; \quad u(x, y, 0) = 1 + \sin(\pi(x + y)),$$

where the source term $h(x, y, t)$ is found from evaluating the left-hand side for the exact solution $u(x, t) = 1 + \sin(\pi(x + y - 2t))$. The solutions are computed up to time $t = 0.5$ in each iteration. The errors and numerical order of accuracy for the LWDG and ALWDG scheme are the same for reasons similar to those discussed for Example 1. For this reason we communicate in Table 5 only the errors obtained for the ALWDG scheme with $k = 1$ and $k = 2$. As in the one dimensional case, the ALWDG scheme achieves its expected order of accuracy with comparable errors for the same mesh. See Figures 2 (a) and (b) for sample solutions on a coarse grid.

Example 5: two-dimensional equation with exponential fluxes

As an example of a numerical solution of a two-dimensional problem where the ALWDG method does not produce the same results as the LWDG method, we consider in Example 5 the same scenario as in Example 4 but employ the equation

$$u_t + (\exp(u^2))_x + (\exp(u^2))_y = h(x, y, t), \quad (x, y) \in [-1, 1]^2, \quad t > 0; \quad u(x, y, 0) = 0.2 \sin(\pi(x + y)),$$

where the source term $h(x, y, t)$ is found from evaluating the left-hand side for the exact solution $u(x, t) = 0.2 \sin(\pi(x + y - 2t))$. See Figures 2 (c) and (d) for sample solutions on a coarse grid.

4.3. Test cases with discontinuous solutions

We choose several examples of problems involving shock waves and test the performance of ALWDG method and compare it with that of original LWDG method.

Example 6: Buckley-Leverett problem

We test the performance of ALWDG applied to a nonlinear Buckley-Leverett problem given by (1.3) with

$$f(u) = \frac{4u^2}{4u^2 + (1-u)^2} = 0, \quad u_0(x) = \begin{cases} 1 & \text{if } -1/2 \leq x \leq 0, \\ 0 & \text{otherwise.} \end{cases}$$

h	# dofs	L^1 error	rate	L^∞ error	rate	L^1 error	rate	L^∞ error	rate
ALWDG, $k = 1$					LWDG, $k = 1$				
0.1	40	1.210895e-03	—	2.094281e-03	—	1.210748e-03	—	2.094082e-03	—
0.05	80	2.936750e-04	2.04	5.751100e-04	1.86	2.433167e-04	2.05	5.750088e-04	1.86
0.025	160	7.270918e-05	2.01	1.494143e-04	1.94	7.268426e-05	2.01	1.494090e-04	1.94
0.0125	320	1.810735e-05	2.01	3.806637e-05	1.97	1.810034e-05	2.01	3.806518e-05	1.97
0.00625	640	4.519657e-06	2.00	9.605015e-06	1.99	4.517921e-06	2.00	9.604689e-06	1.99
0.003125	1280	1.129073e-06	2.00	2.412347e-06	1.99	1.128672e-06	2.00	2.412246e-06	1.99
ALWDG, $k = 2$					LWDG, $k = 2$				
0.1	60	4.932893e-05	—	1.092809e-04	—	4.932893e-05	—	1.092809e-04	—
0.05	120	6.191884e-06	2.99	1.407323e-05	2.96	6.191884e-06	2.99	1.407323e-05	2.96
0.025	240	7.760543e-07	3.00	1.771498e-06	2.99	7.760543e-07	3.00	1.771498e-06	2.99
0.0125	480	9.587929e-08	3.02	2.208515e-07	3.00	9.587929e-08	3.02	2.208515e-07	3.00
0.00625	960	1.203892e-08	2.99	2.767148e-08	3.00	1.203892e-08	2.99	2.767146e-08	3.00
0.003125	1920	1.504556e-09	3.00	3.460747e-09	3.00	1.504558e-09	3.00	3.460733e-09	3.00

Table 3: Example 3 (1D Euler equations of gas dynamics, smooth solution): comparison of ALWDG with LWDG for second and third order discretization (corresponding to $k = 1$ and $k = 2$, respectively), computed up to a final time $t = 2.0$.

h	# dofs	L^1 error	rate	L^∞ error	rate	h	# dofs	L^1 error	rate	L^∞ error	rate
ALWDG, $k = 3$					ALWDG, $k = 4$						
0.1	80	5.438355e-07	-	1.606543e-06	—	0.2	50	3.893091e-07	—	1.217045e-06	—
0.05	160	3.409772e-08	4.00	1.000108e-07	4.01	0.1	100	1.275073e-08	4.93	4.021019e-08	4.92
0.025	320	2.133876e-09	4.00	6.248730e-09	4.00	0.05	200	4.055106e-10	4.97	1.273128e-09	4.98
0.0125	640	1.336286e-10	4.00	3.903649e-10	4.00	0.025	400	1.264587e-11	5.00	3.991241e-11	5.00
0.00625	1280	8.352168e-12	4.00	2.442269e-11	4.00	0.0125	800	3.997046e-13	4.98	1.262990e-12	4.98

Table 4: Example 3 (1D Euler equations of gas dynamics, smooth solution): numerical errors produced by ALWDG with LWDG for fourth- and fifth-order discretization (corresponding to $k = 3$ and $k = 4$, respectively), computed up to a final time $t = 2.0$.

The solution is computed up to time $t = 0.4$ with $N = 80$ number of cells. The corresponding solution using the ALWDG and LWDG methods are compared in Figure 4. The exact solution is a mixture of shock, rarefaction and contact discontinuity. The results obtained here indicate the robustness of our proposed scheme.

Examples 7 and 8: Euler equations of gas dynamics in one space dimension (benchmark tests)

We compare the performance of ALWDG method with the LWDG method for two benchmark test cases, namely the shock tube or Riemann problem for the Euler equations of gas dynamics (4.1), (4.2) posed with the initial datum

$$(\rho_0(x), v_0(x), p_0(x)) = \begin{cases} (\rho_L, v_L, p_L) & \text{if } x < x_0, \\ (\rho_R, v_R, p_R) & \text{if } x > x_0, \end{cases} \quad \text{where } p_R < p_L. \quad (4.4)$$

For a given equation of state (4.2), the solution of the shock tube problem (4.1), (4.2), (4.4) is a function of $(x - x_0)/t$ and of the six constants involved in the data (4.4). It is supposed that a ‘‘diaphragm’’ initially located at $x = x_0$ bursts at $t = 0$. Then a pressure discontinuity propagates to the right in the low-pressure gas and an expansion fan (rarefaction wave) propagates to the left into the high-pressure gas. In addition, a contact discontinuity separating the two gas regions propagates to the right in the tube [9]. Thus, we have three simple waves (from the left to the right, a rarefaction wave, a contact discontinuity, and a shock) that separate regions of uniform conditions. The solution of this problem is described in

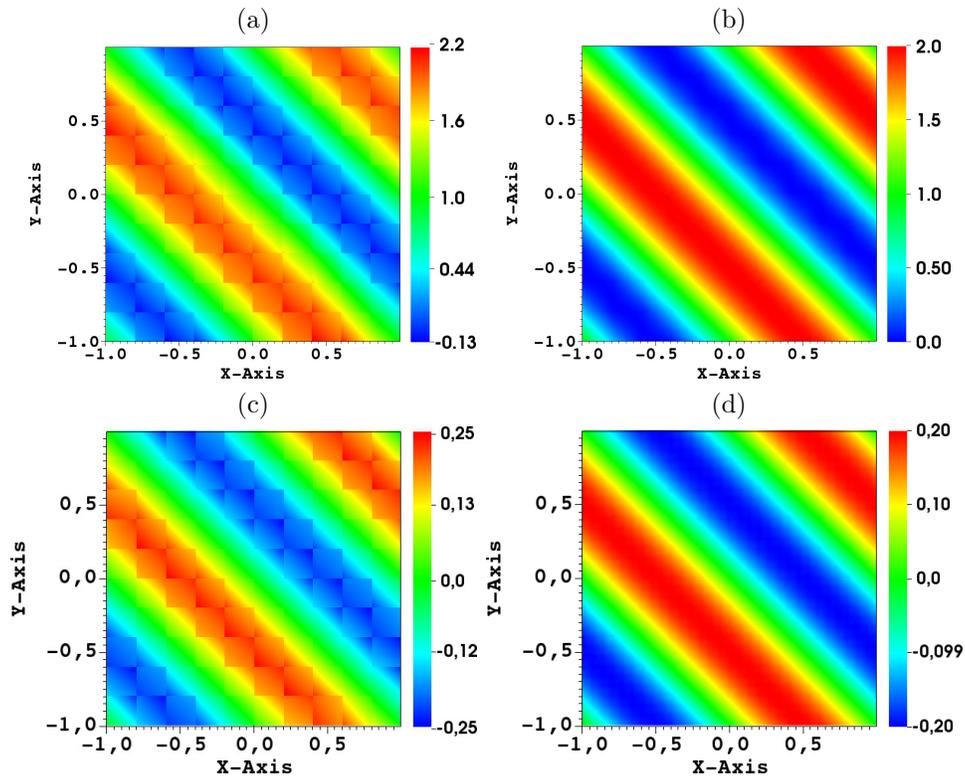


Figure 2: Examples 4 and 5: (a, b) 2D inviscid Burgers equation and (c, d) 2D equation with exponential fluxes: approximate solution computed at a resolution of 10×10 cells and time $T = 0.5$ using the ALWDG method with (a, c) $k = 1$, (b, d) $k = 2$.

detail in [9]; the construction is summarized in the Appendix, where we also provide numerical values of certain constants that arise in the test cases considered here, namely Lax's test problem [10] in Example 7 and Sod's shock tube problem [22] in Example 8.

For Lax's test problem we utilize

$$(\rho_L, v_L, p_L) = (0.445, 0.698, 3.52), \quad (\rho_R, v_R, p_R) = (0.5, 0, 0.571), \quad x_0 = 0. \quad (4.5)$$

The numerical domain for this test case is $[-5, 5]$, and solution is computed at time $t = 1.3$. The initial datum for Sod's shock tube problem is given by (4.4) with

$$(\rho_L, v_L, p_L) = (1, 0, 1), \quad (\rho_R, v_R, p_R) = (0.125, 0, 0.1), \quad x_0 = 0.5. \quad (4.6)$$

The numerical domain for this test problem is $\Omega = [0, 1]$, and we plot the solution at time $t = 0.2$.

The solutions are computed for 100 cells using TVB limiter with the parameter $M=0.0$. The results are depicted in Figure 5 for Examples 7 and 8 combined. It is observed that the results corresponding to ALWDG method agree very well with that of LWDG method. The corresponding errors are displayed in Tables 7 and 8. Here we denote by e_ρ , e_v and e_p the L^1 errors in the variables ρ , v and p at a given time, respectively, and define the total error $e_{\text{tot}} := e_\rho + e_v + e_p$ (utilizing that the three variables assume values roughly in the same order of magnitude).

Example 9: Euler equations of gas dynamics in two space dimensions

In two space dimensions, the Euler equations of gas dynamics turn into a case of (1.1) for $d = 2$ and $m = 4$. The velocity of the gas is now $\mathbf{v} = (v_1, v_2)^T$, and the vector of unknowns becomes $\mathbf{u} = (\rho, \rho v_1, \rho v_2 E)^T$. The two flux vectors are $\mathbf{f}_1(\mathbf{u}) = (\rho v_1, p + \rho v_1^2, \rho v_1 v_2, (E + p)v_1)^T$ and $\mathbf{f}_2(\mathbf{u}) =$

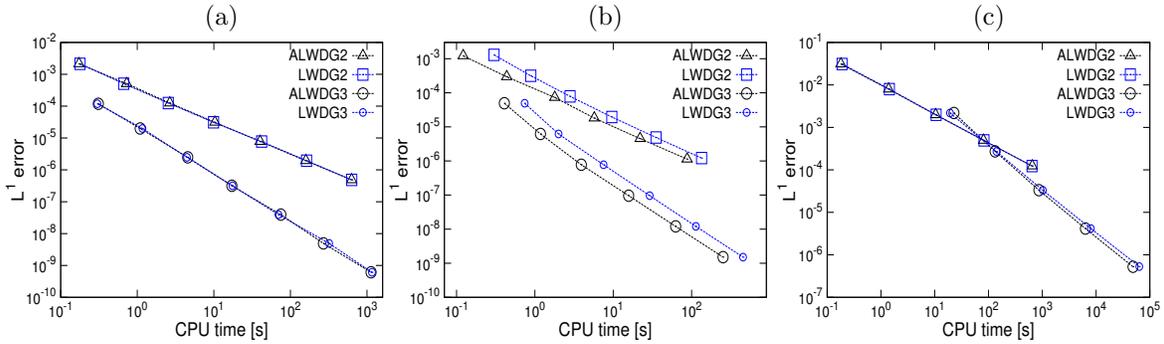


Figure 3: Efficiency plots (L^1 error versus CPU time, in log-log scale) (a) for Example 2 (1D scalar equation with exponential flux and a smooth solution) at $t = 2$, (b) for Example 3 (1D Euler equations of gas dynamics with smooth solution, the error is that of density) at $t = 2$, (c) for Example 5 (2D scalar equation with exponential fluxes and a smooth solution) at $t = 0.5$. The notations ALWDG2 (LWDG2) and ALWDG3 (LWDG3) represent the second- and third-order versions of the corresponding schemes, respectively.

# cells	# dofs	L^1 error	rate	L^∞ error	rate	#	rate	dofs	L^∞ error	rate
ALWDG, $k = 1$						ALWDG, $k = 2$				
10×10	300	1.302100e-01	—	2.206497e-01	—	600	1.541243e-02	—	4.002485e-02	—
20×20	1200	3.187056e-02	2.03	6.054872e-02	1.87	2400	1.701997e-03	3.18	5.239857e-03	2.93
40×40	4800	7.935655e-03	2.01	2.049769e-02	1.56	9600	1.959470e-04	3.12	7.841781e-04	2.74
80×80	19200	1.979620e-03	2.00	6.004550e-03	1.77	38400	2.377506e-05	3.04	1.049010e-04	2.90
160×160	76800	4.952919e-04	2.00	1.666142e-03	1.85	153600	2.948301e-06	3.01	1.400828e-05	2.90

Table 5: Example 4 (inviscid Burgers equation in 2D): errors of numerical solutions produced by the ALWDG methods of second and third order of accuracy ($k = 1$ and $k = 2$, respectively) on the numerical domain $[-1, 1] \times [-1, 1]$. Errors are computed at simulated time $t = 0.5$.

$(\rho v_2, \rho v_1 v_2, p + \rho v_2^2, (E + p)v_2)^T$, where ρ , E and p have the same meaning as before, and the system $\mathbf{u}_t + \mathbf{f}_1(\mathbf{u})_x + \mathbf{f}_2(\mathbf{u})_y = \mathbf{0}$ is supplied with an equation of state $E = E(\rho, p, v)$, for which we choose the same expression (4.2) with $\gamma = 1.4$ as in the one-dimensional case.

In this example we choose the double Mach reflection problem which was originally proposed by Woodward and Colella in [24]. The solution consists of a Mach 10 planar shock wave in air which meets a reflecting wall positioned at $x = 1/6, y = 0$ making an angle of 60° with the x -axis. This situation is simulated in the rectangular domain $\Omega = [0, 4] \times [0, 1]$ with the initial condition

$$\mathbf{u}_0(x, y) = \begin{cases} \mathbf{u}_1 & \text{if } x \leq \frac{1}{6} + \frac{y}{\sqrt{3}}, \\ \mathbf{u}_2 & \text{if } x > \frac{1}{6} + \frac{y}{\sqrt{3}}, \end{cases} \quad \mathbf{u}_1 = \begin{pmatrix} \rho_1 \\ M_1^x \\ M_1^y \\ E_1 \end{pmatrix} = \begin{pmatrix} 8 \\ 57.1576766498 \\ -33 \\ 563.5 \end{pmatrix}, \quad \mathbf{u}_2 = \begin{pmatrix} \rho_2 \\ M_2^x \\ M_2^y \\ E_2 \end{pmatrix} = \begin{pmatrix} 1.4 \\ 0 \\ 0 \\ 2.5 \end{pmatrix}.$$

We impose inflow boundary conditions, with values \mathbf{u}_1 at the left side, $\{0\} \times [0, 1]$, outflow boundary conditions both at $[0, 1/6] \times \{0\}$ and $\{4\} \times [0, 1]$, reflecting boundary conditions at $[1/6, 4] \times \{0\}$ and inflow boundary conditions at the upper side, $[0, 4] \times \{1\}$. We run different simulations until $t = 0.2$ at a resolution of 960×240 (for results on similar resolution we refer to [16]) cells using the proposed ALWDG discretization and the standard RKDG method. To avoid excessive computation of flux derivatives in LWDG we compare our results in the two dimensional Euler system with the standard RKDG method. The non-linear TVB limiter is applied with the constant $M = 100$. The results are depicted in Figure 8 with zoomed plots of the turbulence zone in Figure 9. From those figures we could conclude that the results obtained through ALWDG and RKDG schemes are comparable.

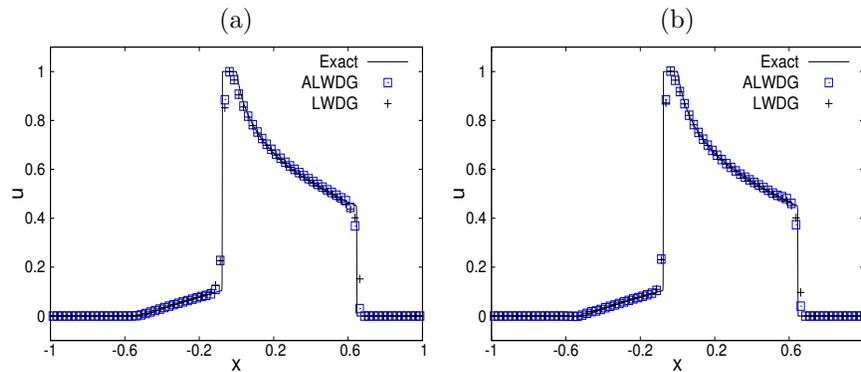


Figure 4: Example 6 (1D Buckley-Leverett equation): numerical solutions with $N = 80$ at time $t = 0.4$ obtained by the ALWDG and LWDG methods with (a) $k = 1$, (b) $k = 2$.

# cells	# dofs	L^1 error	rate	L^∞ error	rate	L^1 error	rate	L^∞ error	rate
ALWDG, $k = 1$					LWDG, $k = 1$				
10×10	300	3.106529e-02	-	6.424697e-02	-	3.101791e-02	-	6.422473e-02	-
20×20	1200	7.969317e-03	1.96	1.664807e-02	1.95	7.966979e-03	1.96	1.664425e-02	1.95
40×40	4800	1.982683e-03	2.01	4.276733e-03	1.96	1.982437e-03	2.01	4.276326e-03	1.96
80×80	19200	4.937426e-04	2.01	1.072786e-03	2.00	4.937241e-04	2.01	1.072748e-03	2.00
160×160	76800	1.231599e-04	2.00	2.675799e-04	2.00	1.231579e-04	2.00	2.675751e-04	2.00
ALWDG, $k = 2$					LWDG, $k = 2$				
10×10	600	2.183560e-03	-	6.750734e-03	-	2.176929e-03	-	6.741173e-03	-
20×20	2400	2.715918e-04	3.01	9.503800e-04	2.83	2.706085e-04	3.01	9.529422e-04	2.82
40×40	9600	3.330633e-05	3.03	1.247777e-04	2.93	3.336922e-05	3.02	1.252792e-04	2.93
80×80	38400	4.145475e-06	3.01	1.589636e-05	2.97	4.250308e-06	2.97	1.599551e-05	2.97
160×160	153600	5.229450e-07	2.99	1.987449e-06	3.00	5.716216e-07	2.89	2.021468e-06	2.98

Table 6: Example 5 (2D equation with exponential fluxes): errors of numerical solutions produced by the ALWDG methods of second and third order of accuracy ($k = 1$ and $k = 2$, respectively) on the numerical domain $[-1, 1] \times [-1, 1]$. Errors are computed at simulated time $t = 0.5$.

5. Conclusions

In this paper a technique involving an approximate Lax-Wendroff-type procedure for discontinuous Galerkin schemes, leading to so-called ALWDG schemes, as an alternative version of the exact procedure defining LWDG schemes [13], has been presented. The basic motivation consists in the ease of implementation of ALWDG schemes that are based on high-order difference approximations on non-uniform grids in each cell. In particular, these approximations do not require a separate symbolic manipulation to determine the higher derivatives of each flux. While one should expect that for a fixed discretization and order, the ALWDG approximation should produce a larger error than the corresponding LWDG method, it turns out in all numerical experiments (in the corresponding error tables and the plots of numerical solutions that hardly differ) that the errors produced by the ALWDG and LWDG versions are practically the same in comparable circumstances. Moreover, in all efficiency plots in Figures 3 and 7 (except for the simple cases of Figures 3 (a) and the $k = 1$ case of Figure 3 (c), where both are almost the same) the traverse line corresponding to the ALWDG version lies below its LWDG counterpart, which means that the ALWDG methods are in some cases comparable and in other cases consistently more efficient (in the usual terms of reduction of error per CPU time) than their LWDG versions (apart from the advantage of ease of implementation), and therefore a serious alternative in themselves to RKDG discretizations.

The results shown in Figures 3 and 7 alert, furthermore, to the fact that the ALWDG methods share properties that are well-known from LWDG schemes as far as the application to smooth or non-smooth

h	# dofs	e_ρ	e_v	e_p	e_{tot}	rate	e_ρ	e_v	e_p	e_{tot}	rate
ALWDG, $k = 1$						LWDG, $k = 1$					
1/3	60	3.266e-01	2.911e-01	3.703e-01	9.880e-01	—	3.272e-01	2.924e-01	3.712e-01	9.908e-01	—
1/6	120	2.110e-01	2.293e-01	2.671e-01	7.074e-01	0.48	2.108e-01	2.286e-01	2.663e-01	7.058e-01	0.49
1/12	240	1.106e-01	1.008e-01	1.315e-01	3.428e-01	1.04	1.106e-01	1.008e-01	1.316e-01	3.430e-01	1.04
1/24	480	6.326e-02	5.996e-02	7.949e-02	2.027e-01	0.76	6.337e-02	5.958e-02	7.915e-02	2.021e-01	0.76
1/48	960	3.328e-02	2.457e-02	4.143e-02	9.929e-02	1.03	3.324e-02	2.452e-02	4.137e-02	9.912e-02	1.03
1/96	1920	2.010e-02	1.625e-02	3.020e-02	6.655e-02	0.58	2.010e-02	1.625e-02	3.019e-02	6.653e-02	0.58
1/192	3840	1.182e-02	8.774e-03	2.204e-02	4.264e-02	0.64	1.182e-02	8.768e-03	2.204e-02	4.263e-02	0.64
ALWDG, $k = 2$						LWDG, $k = 2$					
1/3	90	2.817e-01	4.479e-01	5.021e-01	1.232e00	—	2.746e-01	4.512e-01	5.026e-01	1.228e00	—
1/6	180	1.667e-01	2.386e-01	2.900e-01	6.953e-01	0.82	1.658e-01	2.373e-01	2.880e-01	6.911e-01	0.83
1/12	360	1.139e-01	1.363e-01	1.615e-01	4.118e-01	0.76	1.140e-01	1.364e-01	1.616e-01	4.120e-01	0.75
1/24	720	5.769e-02	5.421e-02	7.532e-02	1.872e-01	1.14	5.759e-02	5.412e-02	7.521e-02	1.869e-01	1.14
1/48	1440	3.245e-02	3.349e-02	4.881e-02	1.148e-01	0.71	3.244e-02	3.354e-02	4.882e-02	1.148e-01	0.70
1/96	2880	1.673e-02	1.466e-02	2.939e-02	6.078e-02	0.92	1.674e-02	1.470e-02	2.938e-02	6.081e-02	0.92
1/192	5760	1.007e-02	9.122e-03	2.265e-02	4.185e-02	0.54	1.007e-02	9.110e-03	2.263e-02	4.182e-02	0.54

Table 7: Example 7 (Lax’s problem; Euler equations in 1D, L^1 error): Comparison of ALWDG with LWDG for second and third order, Lax’s test case. The TVB constant used is $M = 10$.

solution is concerned. Roughly speaking, we observe in the plots of Figure 3, corresponding to problems with a smooth solution, that moving from $k = 1$ to $k = 2$ leads to smaller errors and a marked increase in efficiency. This contrasts with the results of Figure 7 that clearly indicate a loss in efficiency by using a third-order scheme instead of a second-order scheme. It seems pointless, and even counterproductive, to increase the time accuracy in presence of discontinuities that downgrade the spatial accuracy to at most first order. This behaviour is observed with finite difference schemes as well [14]. However, in view of the favorable results obtained for high-order ALWDG schemes with non-linear but smooth solutions, future research should be directed to developing a hybrid time scheme being high order accurate in smooth zones and first order accurate near discontinuities.

Appendix: Solution of the shock tube problem for the Euler equations of gas dynamics

We outline the exact solution of the shock tube or Riemann problem for the one-dimensional Euler equations of gas dynamics (4.1), (4.2), (4.4) following [9, Sect. 16.6.3] (with a few corrections). That treatment is sufficient to handle the two test cases considered herein; a broader (and more recent) exposition of the solution of the shock tube problem is provided, for instance, in [23].

Consistently with [9] we denote by indices L, 5, 3, 2, and R, in the order of increasing $(x - x_0)/t$, the initial left state; the variable solution within the rarefaction wave; the solution between the rarefaction wave and the contact discontinuity; between the contact discontinuity and the shock; and between the shock and the initial right state, respectively. If σ denotes the velocity of propagation of a jump and $[[\cdot]]$ the difference of values of a quantity adjacent to the jump, then the jump condition (Rankine-Hugoniot condition) for (4.1) is

$$[[\rho v]] = \sigma [[\rho]], \quad [[\rho v^2 + p]] = \sigma [[\rho v]], \quad [[(E + p)v]] = \sigma [[\rho E]]. \quad (\text{A.1})$$

The result is as follows. We define the sound speed $c = (\gamma p / \rho)^{1/2}$ (c_1 etc., c_L and c_R are defined analogously), the parameter $\alpha = (\gamma + 1) / (\gamma - 1)$, and the pressure ratio $P = p_2 / p_R$, for which a nonlinear equation needs to be solved. Then exploiting (A.1) we get

$$\rho_2 = \frac{1 + \alpha P}{\alpha + P} \rho_R, \quad v_2 = \frac{P - 1}{(1 + \alpha P)^{1/2}} \frac{1}{(\gamma(\gamma - 1)/2)^{1/2}} c_R + v_R, \quad \sigma = \frac{(P - 1)c_R^2}{\gamma(v_2 - v_R)} + v_R. \quad (\text{A.2})$$

h	# dofs	e_ρ	e_v	e_p	e_{tot}	rate	e_ρ	e_v	e_p	e_{tot}	rate
ALWDG, $k = 1$						LWDG, $k = 1$					
1/50	100	8.000e-03	1.274e-02	6.059e-03	2.680e-02	—	8.009e-03	1.272e-02	6.054e-03	2.678e-02	—
1/100	200	4.191e-03	7.255e-03	3.094e-03	1.454e-02	0.88	4.190e-03	7.260e-03	3.096e-03	1.455e-02	0.88
1/200	400	2.138e-03	3.651e-03	1.538e-03	7.327e-03	0.99	2.139e-03	3.652e-03	1.538e-03	7.329e-03	0.99
1/400	800	1.116e-03	1.869e-03	7.698e-04	3.755e-03	0.96	1.116e-03	1.869e-03	7.700e-04	3.756e-03	0.96
1/800	1600	5.946e-04	1.016e-03	3.960e-04	2.007e-03	0.90	5.945e-04	1.015e-03	3.958e-04	2.005e-03	0.91
1/1600	3200	3.049e-04	4.237e-04	1.875e-04	9.162e-04	1.13	3.049e-04	4.236e-04	1.875e-04	9.160e-04	1.13
1/3200	6400	1.679e-04	2.593e-04	9.983e-05	5.270e-04	0.80	1.679e-04	2.594e-04	9.986e-05	5.271e-04	0.80
ALWDG, $k = 2$						LWDG, $k = 2$					
1/50	150	9.417e-03	1.839e-02	7.551e-03	3.535e-02	—	9.431e-03	1.847e-02	7.567e-03	3.547e-02	—
1/100	300	4.748e-03	9.269e-03	3.816e-03	1.783e-02	0.99	4.754e-03	9.292e-03	3.825e-03	1.787e-02	0.99
1/200	600	2.408e-03	4.741e-03	1.921e-03	9.070e-03	0.98	2.411e-03	4.753e-03	1.925e-03	9.089e-03	0.98
1/400	1200	1.226e-03	2.460e-03	9.715e-04	4.657e-03	0.96	1.227e-03	2.466e-03	9.737e-04	4.667e-03	0.96
1/800	2400	6.188e-04	1.125e-03	4.782e-04	2.222e-03	1.07	6.193e-04	1.127e-03	4.791e-04	2.225e-03	1.07
1/1600	4800	3.205e-04	6.271e-04	2.445e-04	1.192e-03	0.90	3.211e-04	6.290e-04	2.452e-04	1.195e-03	0.90
1/3200	9600	1.591e-04	2.807e-04	1.190e-04	5.588e-04	1.09	1.593e-04	2.813e-04	1.192e-04	5.597e-04	1.09

Table 8: Example 8 (Euler equations in 1D, L^1 error): Comparison of ALWDG with LWDG for second and third order, Sod's test case. The TVB constant used is $M = 10$.

On the other hand, across the contact discontinuity only ρ suffers a jump, so $p_3 = p_2$ and $v_3 = v_2 =: V$. The value of ρ_3 is obtained by considering that the rarefaction wave propagating to the left is formed by characteristics moving at velocity $v - c$. Since the leftmost characteristic of that fan the entropy p/ρ^γ is constant and along the rightmost characteristic the Riemann invariant $(\gamma - 1)u/2 + c$ is constant, we get

$$\rho_3 = (p_3/p_L)^{1/\gamma} \rho_L = (p_2/p_L)^{1/\gamma} \rho_L \quad (\text{A.3})$$

and the following equation that relates $p_2 = p_3$ and p_L :

$$V - v_L = \frac{2c_L}{\gamma - 1} \left[1 - \left(\frac{p_2}{p_L} \right)^{(\gamma-1)/(2\gamma)} \right]. \quad (\text{A.4})$$

Since V is also defined by the second equation of (A.2), eliminating V between that equation and (A.4) we obtain the following equation that can be solved iteratively for P in terms of p_L/p_R :

$$\left(\frac{2}{\gamma(\gamma - 1)} \right)^{1/2} \frac{P - 1}{(1 + \alpha P)^{1/2}} = \frac{2}{\gamma - 1} \frac{c_L}{c_R} \left[1 - \left(\frac{p_R}{p_L} \right)^{(\gamma-1)/(2\gamma)} P^{(\gamma-1)/(2\gamma)} \right] + \frac{v_L - v_R}{c_R}. \quad (\text{A.5})$$

Finally, a discussion of the characteristics that form the rarefaction fan reveals that region 5 corresponds to those (x, t) for which

$$\xi^- := v_L - c_L < \frac{x - x_0}{t} < \frac{\gamma + 1}{2} V - c_L - \frac{\gamma - 1}{2} v_L =: \xi^+, \quad (\text{A.6})$$

and the unknown functions are given by

$$\begin{aligned} v_5(x, t) &= \frac{2}{\gamma + 1} \left(\frac{x - x_0}{t} + c_L + \frac{\gamma - 1}{2} v_L \right), & c_5(x, t) &= v_5(x, t) - \frac{x - x_0}{t}, \\ p_5(x, t) &= (c_5(x, t)/c_L)^{2\gamma/(\gamma-1)} p_L, & \rho_5(x, t) &= (p_5(x, t)/p_L)^{1/\gamma} \rho_L. \end{aligned} \quad (\text{A.7})$$

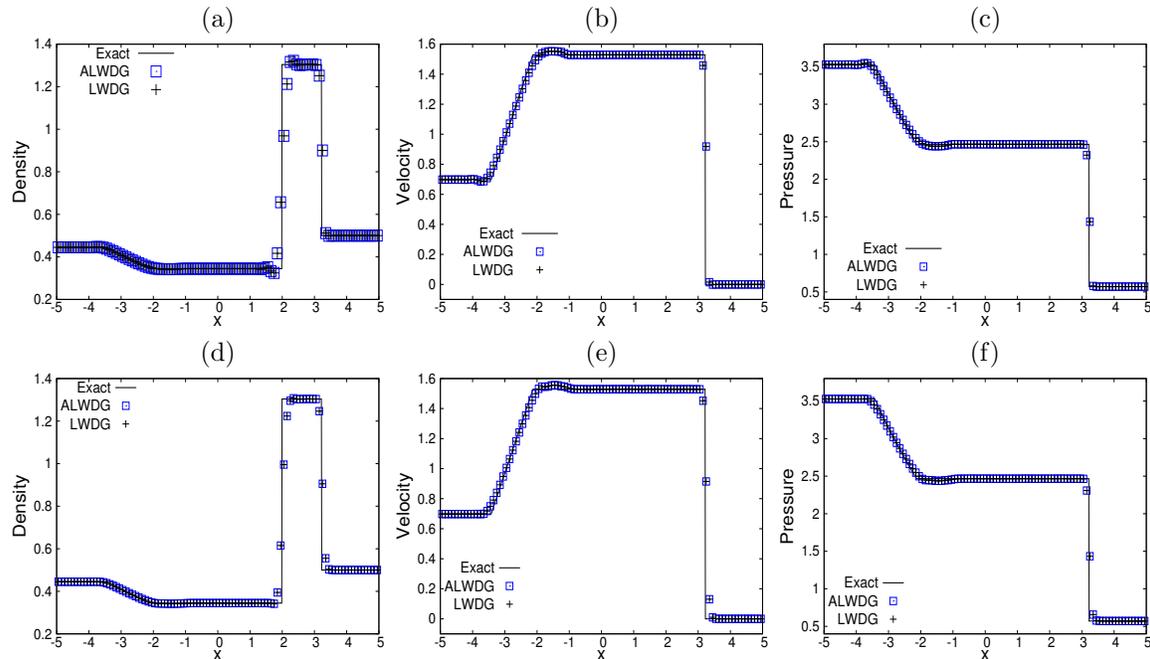


Figure 5: Example 7 (1D Euler equations of gas dynamics, Lax's test case): numerical solutions with $N = 100$ at $t = 1.3$ obtained by the ALWDG and LWDG methods with (a, b, c) $k = 1$, (d, e, f) $k = 2$. The value of the constant M associated with the TVB limiter is $M = 10$.

Consequently, the solution is given by

$$(\rho, v, p)(x, t) = \begin{cases} (\rho_R, v_R, p_R) & \text{for } x - x_0 > \sigma t, \\ (\rho_3, v_2, p_2) & \text{for } v_2 t < x - x_0 \leq \sigma t, \\ (\rho_2, v_2, p_2) & \text{for } \xi^+ t < x - x_0 \leq v_2 t, \\ (\rho_5(x, t), v_5(x, t), p_5(x, t)) & \text{(defined in (A.7)) for } \xi^- t < x - x_0 \leq \xi^+ t, \\ (\rho_L, v_L, p_L) & \text{for } x - x_0 \leq \xi^- t. \end{cases} \quad (\text{A.8})$$

(Equations (A.5), (A.6) and the equation for p_5 in (A.7) are unfortunately stated incorrectly in [9].)

Table 9 provides accurate values of the constants arising in the solution of Sod's problem (4.6) and of Lax's problem (4.5) calculated from (A.2)–(A.6). In both cases, $\alpha = 6$.

Acknowledgements

RB is supported by Fondecyt project 1130154; BASAL project CMM, Universidad de Chile and Centro de Investigación en Ingeniería Matemática (CI²MA), Universidad de Concepción; and CRHIAM, project CONICYT/FONDAP/15130015. SKK acknowledge the support of Fondecyt project 3150313 and BASAL project CMM, Universidad de Chile and CI²MA. Also SKK would like to thank Prof. Praveen C (TIFR CAM, Bangalore, India) for some helpful discussion. DZ is supported by project CPI-16-388, Universitat de València.

References

- [1] W. Bangerth, R. Hartmann, G. Kanschat, ACM Trans. Math. Softw. 33 (2007) No. 4, 24/1–24/27.
- [2] C. Bresten, S. Gottlieb, Z. Grant, D. Higgs, D.I. Ketcheson, A. Németh, Explicit strong stability preserving multistep Runge-Kutta methods, Math. Comp. 86 (2017) 747–769.

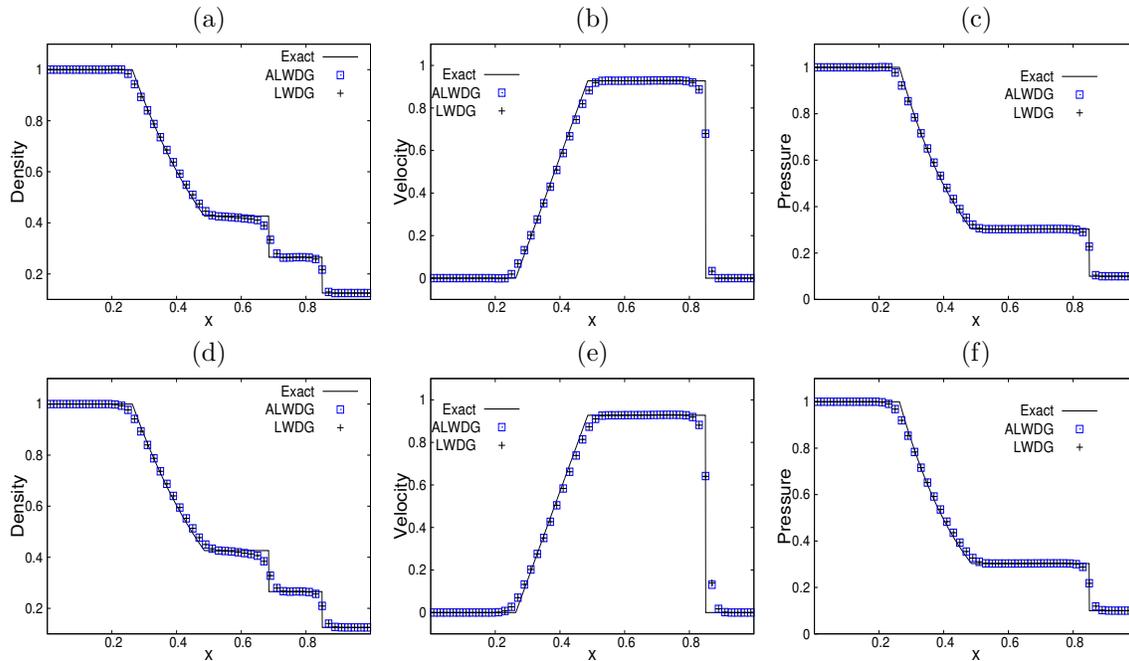


Figure 6: Example 8 (1D Euler equations of gas dynamics, Sod's test case): numerical solutions with $N = 100$ at $t = 0.2$ obtained by the ALWDG and LWDG methods with (a, b, c) $k = 1$, (d, e, f) $k = 2$. The value of the constant M associated with the TVB limiter is $M = 10$.

- [3] B. Cockburn, C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws. V. Multidimensional systems, *J. Comput. Phys.* 141 (1998) 199–224.
- [4] B. Cockburn, S. Hou, C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, *Math. Comp.* 54 (1990) 545–581.
- [5] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, *J. Comput. Phys.* 84 (1989) 90–113.
- [6] B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, *Math. Comp.* 52 (1989) 411–435.
- [7] B. Cockburn, C.-W. Shu, The Runge-Kutta local projection P1-discontinuous Galerkin finite element method for scalar conservation laws, *Math. Model. Numer. Anal.* 25 (1991) 337–361.
- [8] W. Guo, J.-M. Qiu, J. Qiu, A new Lax-Wendroff discontinuous Galerkin method with superconvergence, *J. Sci. Comput.* 65 (2015) 299–326.
- [9] C. Hirsch, *Numerical Computation of Internal and External Flows, Volume 2: Computational Methods for Inviscid and Viscous Flows*, Wiley, 1990.
- [10] P.D. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computation, *Commun. Pure Appl. Math.* 7 (1954) 159–193.
- [11] P. Lax, B. Wendroff, Systems of conservation laws, *Commun. Pure Appl. Math.* 13 (1960) 217–237
- [12] J. Qiu, A numerical comparison of the Lax-Wendroff discontinuous Galerkin method based on different numerical fluxes, *J. Sci. Comput.* 30 (2007) 345–367.

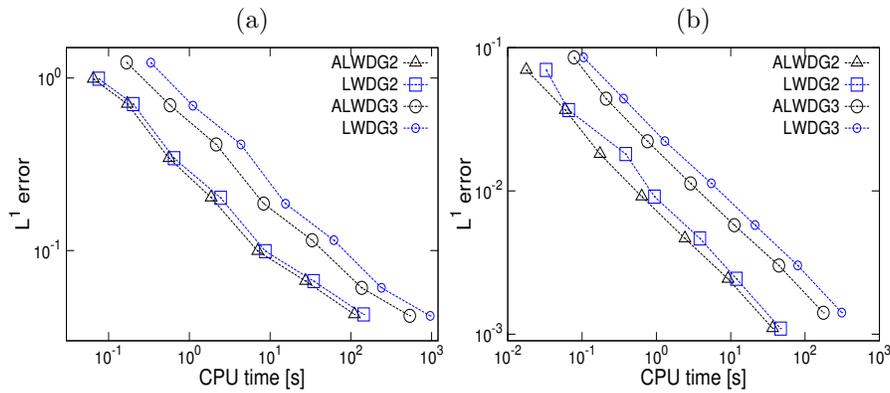


Figure 7: Examples 7 and 8 : comparison of efficiency: 1D Euler equations of gas dynamics, (a) Lax's test case, (b) Sod's test case. In each case we plot the CPU time against the sum of three errors of density, velocity and pressure in log-log scale. The notations ALWDG2 (LWDG2) and ALWDG3 (LWDG3) represent the second- and third-order versions of the corresponding schemes, respectively.

	Lax's problem (4.5)	Sod's problem (4.6)
c_L	3.3277856456465	1.1832159566199
c_R	1.2644366334459	1.0583005244258
σ	2.4772593104830	1.7521557320302
ρ_2	1.3028578166990	0.2655737117053
v_2	1.5265572159292	0.9274526200490
p_2	2.4618390380729	0.3031301780506
ρ_3	0.3447018194278	0.4263194281785
ξ^+	-1.6355169865314	-0.0702728125612
ξ^-	-2.6297856456465	-1.1832159566199

Table 9: Accurate values of constants arising in the solution (A.8) of Lax's and Sod's problems, (4.5) and (4.6).

- [13] J. Qiu, M. Dumbser, C.-W. Shu, The discontinuous Galerkin method with Lax-Wendroff type time discretizations, *Comput. Meth. Appl. Mech. Engrg.* 194 (2005) 4528–4543.
- [14] J. Qiu, C.-W. Shu, Finite difference WENO schemes with Lax-Wendroff-type time discretizations, *SIAM J. Sci. Comput.* 24 (2003) 2185–2198.
- [15] J. Qiu, C.-W. Shu, Runge-Kutta discontinuous Galerkin method using WENO limiters, *SIAM J. Sci. Comput.* 26 (2005) 907–929.
- [16] J. Qiu, B.C. Khoo, C.-W. Shu, A numerical study for the performance of the Runge-Kutta discontinuous Galerkin method based on different numerical fluxes, *J. Comput. Phys.* 212 (2006) 540–565.
- [17] J. Qiu, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case, *J. Comput. Phys.* 193 (2003) 115–135.
- [18] J. Qiu, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: two-dimensional case, *Comput. Fluids*, 34 (2005) 642–663.
- [19] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* 77 (1988) 439–471.
- [20] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes II., *J. Comput. Phys.* 83 (1989) 32–78.

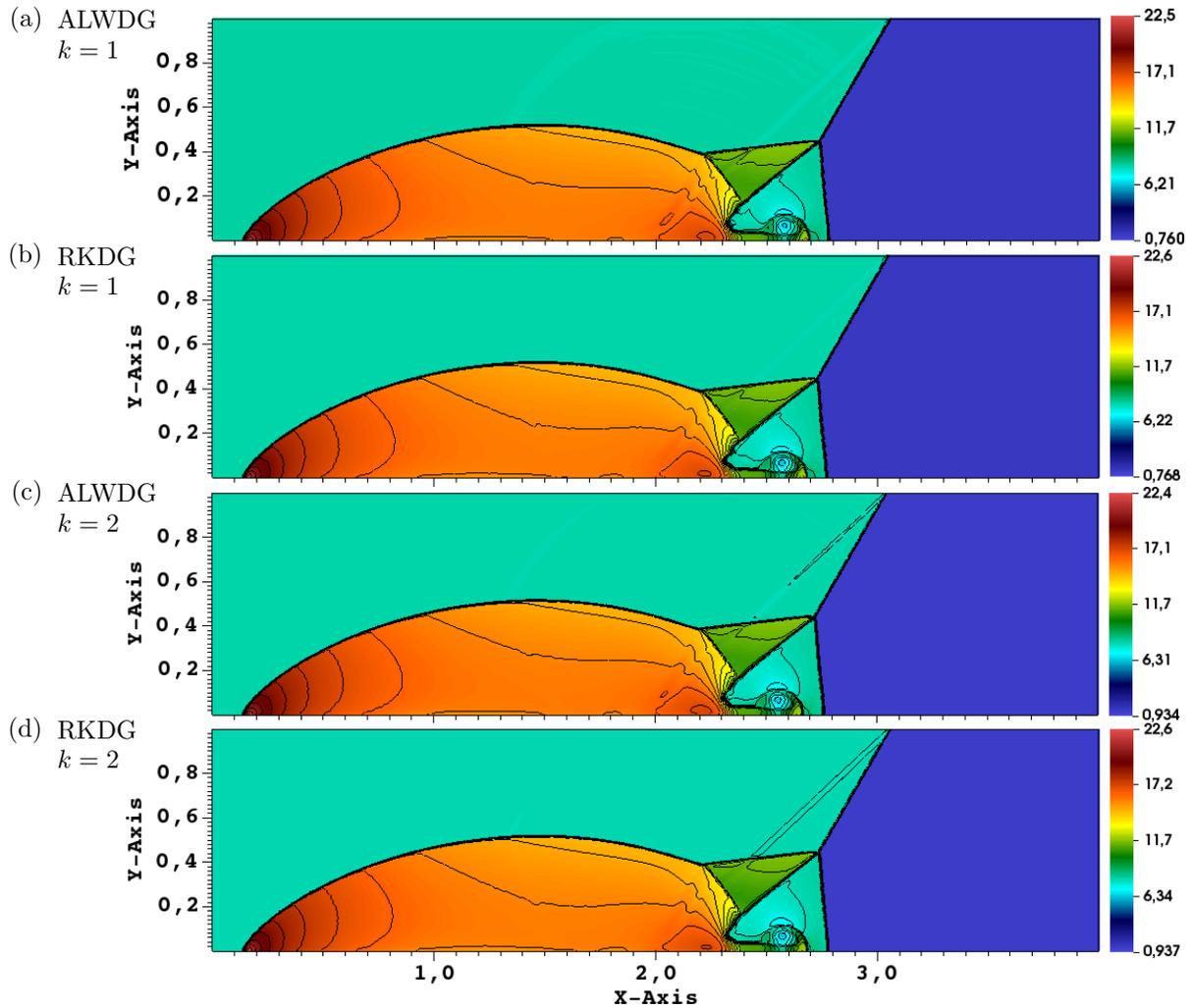


Figure 8: Example 9 (2D Euler equations of gas dynamics, double Mach reflection problem): contours of 30 steps are printed, $\Delta x = \Delta y = 1/240$. Contours of density obtained by the method indicated in each case (ALWDG or RKDG, $k = 1$ or $k = 2$).

- [21] C.-W. Shu, TVB uniformly high-order schemes for conservation laws, *Math. Comp.* 49 (1987) 105–121.
- [22] G.A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, *J. Comput. Phys.* 27 (1978) 1–31.
- [23] E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, 3rd edn., Springer, Berlin, 2009.
- [24] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comput. Phys.* 54 (1984) 115–173.
- [25] D. Zorío, A. Baeza, P. Mulet, An approximate Lax-Wendroff-type procedure for high order accurate schemes for hyperbolic conservation laws, *J. Sci. Comput.*, in press.

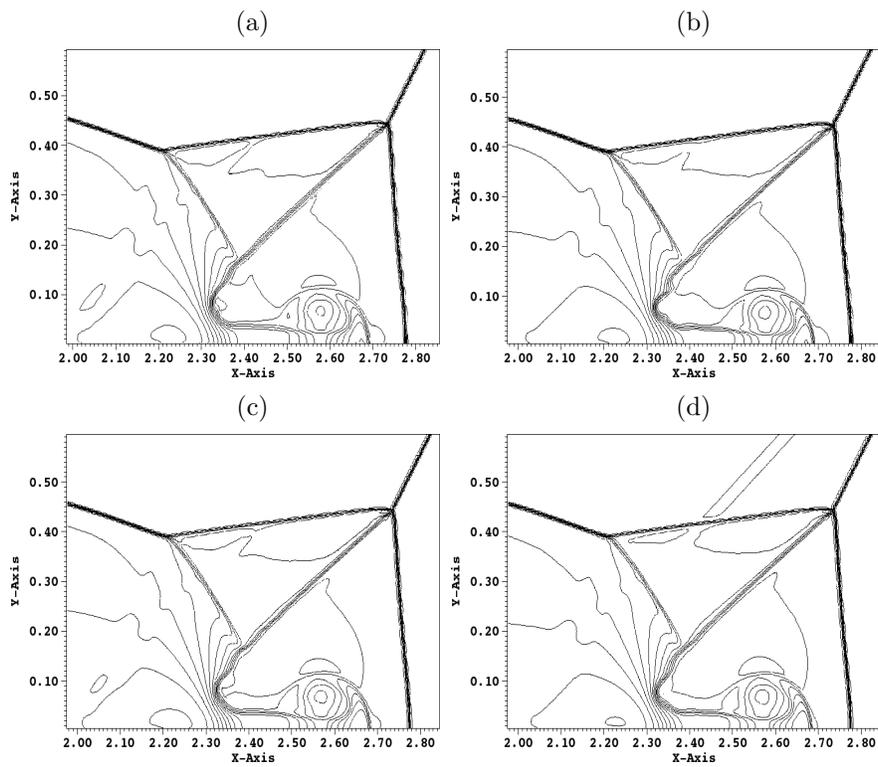


Figure 9: Example 9 (2D Euler equations of gas dynamics, double Mach reflection problem): contours of 30 steps are printed, $\Delta x = \Delta y = 1/240$. Enlarged views of Figures 8 (a), (b), (c), and (d), respectively.

Centro de Investigación en Ingeniería Matemática (CI²MA)

PRE-PUBLICACIONES 2016 - 2017

- 2016-32 DAVID MORA, GONZALO RIVERA, RODOLFO RODRÍGUEZ: *A posteriori error estimates for a virtual elements method for the Steklov eigenvalue problem*
- 2016-33 RAIMUND BÜRGER, GERARDO CHOWELL, ELVIS GAVILÁN, PEP MULET, LUIS M. VILLADA: *Numerical solution of a spatio-temporal gender-structured model for hantavirus infection in rodents*
- 2016-34 FELIPE LEPE, SALIM MEDDAHI, DAVID MORA, RODOLFO RODRÍGUEZ: *Acoustic vibration problem for dissipative fluids*
- 2016-35 ALFREDO BERMÚDEZ, MARTA PIÑEIRO, RODOLFO RODRÍGUEZ, PILAR SALGADO: *Analysis of an ungauged $T, \phi - \phi$ formulation of the eddy current problem with currents and voltage excitations*
- 2016-36 JESSIKA CAMAÑO, RICARDO OYARZÚA, RICARDO RUIZ-BAIER, GIORDANO TIERRA: *Error analysis of an augmented mixed method for the Navier-Stokes problem with mixed boundary conditions*
- 2016-37 ELIGIO COLMENARES, GABRIEL N. GATICA, RICARDO OYARZÚA: *A posteriori error analysis of an augmented mixed-primal formulation for the stationary Boussinesq model*
- 2016-38 ROMMEL BUSTINZA, ARIEL LOMBARDI, MANUEL SOLANO: *An anisotropic a priori error estimate for a convection-dominated diffusion problem using the HDG method*
- 2016-39 RAIMUND BÜRGER, STEFAN DIEHL, MARÍA CARMEN MARTÍ, PEP MULET, INGMAR NOPENS, ELENA TORFS, PETER VANROLLEGHEM: *Numerical solution of a multi-class model for batch settling in water resource recovery facilities*
- 2016-40 RAIMUND BÜRGER, JULIO CAREAGA, STEFAN DIEHL: *Flux identification for scalar conservation laws modelling sedimentation in vessels with varying cross-sectional area*
- 2016-41 CHRISTOPHE CHALONS, PAOLA GOATIN, LUIS M. VILLADA: *High order numerical schemes for one-dimension non-local conservation laws*
- 2016-42 JESSIKA CAMAÑO, CHRISTOPHER LACKNER, PETER MONK: *Electromagnetic Stekloff eigenvalues in inverse scattering*
- 2017-01 RAIMUND BÜRGER, SUDARSHAN K. KENETTINKARA, DAVID ZORÍO: *Approximate Lax-Wendroff discontinuous Galerkin methods for hyperbolic conservation laws*

Para obtener copias de las Pre-Publicaciones, escribir o llamar a: DIRECTOR, CENTRO DE INVESTIGACIÓN EN INGENIERÍA MATEMÁTICA, UNIVERSIDAD DE CONCEPCIÓN, CASILLA 160-C, CONCEPCIÓN, CHILE, TEL.: 41-2661324, o bien, visitar la página web del centro: <http://www.ci2ma.udec.cl>



**CENTRO DE INVESTIGACIÓN EN
INGENIERÍA MATEMÁTICA (CI²MA)
Universidad de Concepción**



Casilla 160-C, Concepción, Chile
Tel.: 56-41-2661324/2661554/2661316
<http://www.ci2ma.udec.cl>