

# Implicit multistage two-derivative discontinuous Galerkin schemes for viscous conservation laws

Alexander Jaust, Jochen Schütz and David C. Seal

October 27, 2015

In this paper we apply implicit two-derivative multistage time integrators to viscous conservation laws in one and two dimensions. The one dimensional solver discretizes space with the classical discontinuous Galerkin (DG) method, and the two dimensional solver uses a hybridized discontinuous Galerkin (HDG) spatial discretization for efficiency. We propose methods that permit us to construct implicit solvers using each of these spatial discretizations, wherein a chief difficulty is how to handle the higher derivatives in time. The end result is that the multiderivative time integrator allows us to obtain high-order accuracy in time while keeping the number of implicit stages at a minimum. We show numerical results validating and comparing methods.

## 1 Introduction

In this work, we focus on viscous conservation laws and present an implicit high-order time integration schemes for the discontinuous Galerkin (DG) method [38, 11, 10, 9, 8, 12]. One of the advantages of using DG is the easy increase of accuracy by locally increasing the polynomial degree which makes it imperative to also use high-order time integrators. Explicit Runge-Kutta time integration schemes are often used because of their low computational costs, low dissipation, favorable stability regions and ease of implementation. However, time step size is limited by the CFL condition which links both time step size  $\Delta t$  and mesh size to the speed of propagation of the physical system. In practical applications, the required time step may be unacceptably small due to very small local mesh sizes or the stiffness of the discretized problem. Frequently, a severe CFL restriction can be overcome by using implicit time integrators, allowing larger time steps at the cost of each time step being more expensive. The latter is due to the fact that each time step requires one or more systems of (usually nonlinear) equations to be solved. Common examples of implicit time integrators include multistep methods such as the backward differentiation formulae (BDF) or multistage methods such as diagonally implicit Runge-Kutta (DIRK) methods [1].

One drawback of the discontinuous Galerkin method within an implicit solver is the large number of degrees of freedom. The hybridized discontinuous Galerkin (HDG) methods, see, e.g., [7, 32, 39], has initially been designed to reduce the memory footprint of steady-state computations, but can also be applied to time-dependent problems [31, 33, 32, 26].

In this paper, we employ two-derivative methods as our time discretization. As the name suggests, these algorithms make use of more than one time derivative and can be constructed to have a strong stability preserving (SSP) property [6, 34]. The principle can be best explained by starting with Taylor methods, which are a subclass of the so-called multistage multiderivative methods [20, 40]. Assuming an unknown function  $w(x, t)$  known at some given time  $t^n$ . Then, an approximation at some time  $t > t^n$  could be

obtained from a Taylor expansion in time

$$w(x, t) = w(x, t^n) + (t - t^n)w_t(x, t^n) + \frac{(t - t^n)^2}{2}w_{tt}(x, t^n) + \dots \quad (1)$$

Obviously, within this approach, it is necessary to approximate additional time derivatives. In the context of viscous conservation laws (that we define in Eqn. (2)), one can replace temporal derivatives by spatial derivatives using the Cauchy-Kowalevskaya procedure.

In the context of numerical methods, the idea of using Taylor series to discretize time is also often referred to as Lax-Wendroff method, because they construct their second-order scheme using this procedure [30]. This approach is used for so-called Taylor-Galerkin methods from the 1980's [14, 13], and the original ENO scheme of Harten et al. [22] uses the same procedure. In [15, 16], Dumbser and Munz construct discontinuous Galerkin schemes with arbitrary order of accuracy in space and time based on ADER schemes. They also present an approach to evaluate the Cauchy-Kowalevskaya procedure efficiently in [16] based on the work of Dyson [17] relying on the application of the Leibniz rule. Qiu et al. [36] present an approach to couple Lax-Wendroff and discontinuous Galerkin methods (LWDG). They obtain a high-order, explicit one step method for hyperbolic problems that is up to third order accurate in time. They show that for their setting, the LWDG scheme is more efficient than a Runge-Kutta discontinuous Galerkin (RKDG). In [35] the approach is extended to 1D convection-diffusion equations based on the local discontinuous Galerkin (LDG) method. Furthermore, the behavior of the method coupled to different numerical fluxes is studied. Additionally, Taylor discretizations are investigated for finite difference weighted ENO methods in [37, 29, 41, 5].

Seal et al. [40] are the first to extend the approach to explicit multiderivative Runge-Kutta methods with DG and WENO spatial discretizations for hyperbolic conservation laws in a single dimension. They develop a framework for two-derivative Runge-Kutta methods that can be easily extended to incorporate additional stages or derivatives. Tsai et al. [44] apply explicit and implicit two-derivative Runge-Kutta methods to PDEs with high-order finite-difference methods for spatial discretization.

Instead of replacing time derivatives with spatial derivatives, it is also possible to go the opposite way. This is done in the so-called inverse Lax-Wendroff (ILW) methods [24, 23, 42, 43, 3]

In this paper, we develop a strategy to apply implicit multistep two-derivative methods to convection-diffusion type equations in 1D using DG. That approach is then extended to first-order PDEs in 2D where we employ HDG for increased efficiency. The approach shows some similarities to the one in [44], but we use DG instead of finite differences for our spatial discretization. As the time derivatives that arise from two-derivative time integrators are replaced by spatial derivatives, an accurate way to represent them is needed. This could be done by differentiating the polynomial representation of the solution in the DG setting, having been justified in [40] by the fact that the derivatives are multiplied by additional  $\Delta t$  that scales as  $\Delta t = \mathcal{O}(\Delta x)$ . In general, the time step restriction for implicit time integration is less severe. Therefore, we employ the LDG approach to accurately represent the additional derivatives, which has the additional benefit of potentially recovering superconvergence properties [18]. We refer the interested reader to [45] where the application of LDG to PDEs of higher order is discussed extensively. In this work, we also show how to extend our approach efficiently to solve for two dimensional convection problems with the hybridized DG method.

The paper is structured as follows. In Section 2, we introduce a 1D nonlinear viscous conservation law that serves as a model equation. Then, in Section 3, we briefly describe the two-derivative multistage time integrators that are used in this work. This also explains the appearance of higher order spatial derivatives that are not directly present in the underlying PDE. Afterwards, we discretize the model equation in time and space using the LDG approach (c.f. Section 4) and verify the method using linear and nonlinear PDEs in Section 5. The two-derivative time integration is then extended to first order PDEs in two dimensions in Section 6. The resulting equation is discretized using an HDG method that significantly reduces the size of the globally coupled system. Finally, we verify the approach using the linear advection and nonlinear Euler equations in Section 7.

## 2 Underlying equation

In this work, we begin with the scalar nonlinear viscous conservation law

$$\begin{aligned} w_t + f(w)_x &= \varepsilon w_{xx} \quad \forall (x, t) \in \Omega \times \mathbb{R}^+ \\ w(x, 0) &= w_0(x) \quad \forall x \in \Omega \end{aligned} \quad (2)$$

with  $\varepsilon \geq 0$  given on a domain  $\Omega \subset \mathbb{R}$  equipped with periodic boundary conditions. The method to be developed relies - similar to a Lax-Wendroff procedure [30] - on the use of the second temporal derivative  $w_{tt}$ , expressed in terms of spatial derivatives. For the underlying problem, we state the following lemma:

**Lemma 1.** *Let  $w \in C^4(\Omega \times \mathbb{R}^+)$ . Then, the second temporal derivative can be expressed as*

$$w_{tt} = (f'(w)f(w)_x - \varepsilon f'(w)w_{xx})_x + \varepsilon (-f(w)_x + \varepsilon w_{xx})_{xx} =: \mathcal{R}^2(w). \quad (3)$$

*Proof.* Obviously, there holds

$$w_t = -f(w)_x + \varepsilon w_{xx} =: \mathcal{R}(w)$$

and consequently,

$$\begin{aligned} w_{tt} &= (-f(w)_x + \varepsilon w_{xx})_t = (-f(w)_t)_x + \varepsilon (w_t)_{xx} = (-f'(w)w_t)_x + \varepsilon (w_t)_{xx} \\ &= (f'(w)f(w)_x - \varepsilon f'(w)w_{xx})_x + \varepsilon (-f(w)_x + \varepsilon w_{xx})_{xx} =: \mathcal{R}^2(w). \end{aligned}$$

□

□

**Remark 1** (Limiting cases). *The term for  $w_{tt}$  simplifies significantly in some limiting cases:*

1. *If  $f$  is linear, i.e.,  $f(w) = cw$ , then*

$$w_{tt} = c^2 w_{xx} - 2\varepsilon c w_{xxx} + \varepsilon^2 w_{xxxx}.$$

2. *This also means that for  $f \equiv 0$ ,  $w_{tt} = \varepsilon^2 w_{xxxx}$ .*

3. *If  $\varepsilon \equiv 0$ , there  $w_{tt} = (f'(w)f(w)_x)_x$ .*

*Note that the viscous and convective terms influence each other mutually, i.e., one obtains cross-terms that need to be dealt with.*

## 3 Time integration

In this section, we shortly review multiderivative time integrators as far as it is of importance for this work. Assume that the ordinary differential equation

$$y'(t) = g(y(t))$$

is given for a smooth function  $g$ . Classical approaches (e.g. multistage Runge-Kutta, or linear multistep methods) to the numerical approximation of these equations [19, 21] only use  $g$  itself. A multiderivative method, on the other hand, takes knowledge of higher derivatives of  $y$  into consideration. As an example, the second derivative  $y''$  is given by

$$y''(t) = \dot{g}(y) := g'(y)g(y),$$

which can be readily computed for an ODE using symbolic differentiation software. In this publication, we assume that  $0 \leq t \leq T$ , and that this temporal interval is uniformly subdivided into  $0 = t^0 < t^1 < \dots < t^N = T$  with spacing  $\Delta t$ . We note that none of the algorithms presented depends on a uniform time step size, and this choice is just for the ease of presentation. As is customary,  $y^n$  denotes an approximation to  $y$  at time  $t = t^n$ .

In this work, we consider *implicit* two-point collocation methods with multiple derivatives of the form

$$\sum_{j=0}^m \Delta t^j (D^j y)(y^{n+1}) P^{(m-j)}(0) = \sum_{j=0}^m \Delta t^j (D^j y)(y^n) P^{(m-j)}(1) \quad (4)$$

where  $P(t) = \frac{t^k(t-1)^l}{(k+l)!}$ . Each of these methods are of order  $m = l + k$  (c.f. II.13 in [19]). Because of the growing complexity of higher order derivatives (see also Eqn. (3)), we rely on schemes involving only two derivatives of the unknown.

**Remark 2** (Employed methods). 

- *In this work, we discuss the third-order method with  $k = 1, l = 2$ , given by*

$$y^{n+1} = y^n + \frac{\Delta t}{3} (g(y^n) + 2g(y^{n+1})) - \frac{\Delta t^2}{6} \dot{g}(y^{n+1}), \quad (5)$$

and the fourth-order scheme with  $k = l = 2$  given by

$$y^{n+1} = y^n + \frac{\Delta t}{2} (g(y^n) + g(y^{n+1})) + \frac{\Delta t^2}{12} (\dot{g}(y^n) - \dot{g}(y^{n+1})). \quad (6)$$

These are the same methods that are used to discretize the non-linear terms in [4].

• *More generally, the methods can be written in form*

$$y^{n+1} = y^n + \Delta t (\alpha_1 g(y^n) + \alpha_2 g(y^{n+1})) + \Delta t^2 (\beta_1 \dot{g}(y^n) + \beta_2 \dot{g}(y^{n+1})), \quad (7)$$

where the coefficients  $\alpha_i, \beta_i, i = 1, 2$  are chosen to maximize the order of accuracy (e.g. Eqn. (6)), or to modify the region of absolute stability (e.g. Eqn. (5) is an L-stable method).

**Lemma 2** (Stability). *The integrators (5) and (6) are third- and fourth-order accurate, respectively, and A-stable. The third-order method (5) is L-stable.*

Applying (7) to equation (2) on a semi-discrete level yields the expression

$$w^{n+1} - \Delta t \alpha_2 w_t^{n+1} - \beta_2 \Delta t^2 w_{tt}^{n+1} = w^n + \Delta t \alpha_1 w_t^n + \beta_1 \Delta t^2 w_{tt}^n,$$

where  $w_{tt}$  has to be replaced by the expression in (3). The term  $w_{tt}$  contains spatial derivatives up to fourth order, so we have to discuss how to discretize them in a DG framework efficiently. In [45, 18], the authors show how to use the *local discontinuous Galerkin* (LDG) method to discretize the higher spatial derivatives in an *explicit* DG solver. Their work will be the basis for the algorithm to be presented in the sequel.

## 4 1D: Spatial and temporal discretization

It is the aim of this publication to couple temporal discretization in (7) to the discontinuous Galerkin method. A semi-discretization of (2) is given by

$$w^{n+1} = w^n + \Delta t (\alpha_1 \mathcal{R}(w^n) + \alpha_2 \mathcal{R}(w^{n+1})) + \Delta t^2 (\beta_1 \mathcal{R}^2(w^n) + \beta_2 \mathcal{R}^2(w^{n+1})), \quad (8)$$

where  $\mathcal{R}(w)$  and  $\mathcal{R}^2(w)$  denote the expressions for  $w_t$  and  $w_{tt}$ , respectively (c.f. Eqn. (3)), and  $w^n$  denotes an approximation to  $w$  at time  $t^n$ .

Before introducing the full spatial and temporal discretization, we start with some preliminaries. To introduce a finite element method, we begin by defining a triangulation of  $\Omega$  into cells  $\Omega_k$  such that they define a partition

$$\Omega = \bigcup_{k=1}^{N_e} \Omega_k$$

with a total of  $N_e$  elements. For a given polynomial order  $p$ , we define the ansatz space  $V_h$  to consist of cell-wise polynomials of order  $p$  with no continuity restriction along the cell boundaries

$$V_h := \{q \in L^2(\Omega) \mid q \in \Pi^p(\Omega_k) \quad \forall k = 1, \dots, N_e\}.$$

Again, it is possible to choose an adaptive  $p$  that differs from cell to cell. We neglect this, for ease of exposition.

The method to be presented relies on the quantities

$$\sigma := u_x, \quad \tau := \sigma_x = u_{xx}, \quad \psi := \tau_x = u_{xxx}.$$

In the most straightforward way, these variables are discretized as

$$\begin{aligned} (\psi_h, \varphi_h)_{\Omega_k} + (\tau_h, (\varphi_h)_x)_{\Omega_k} - \langle \widehat{\tau}, \varphi_h n \rangle_{\partial\Omega_k} &= 0 \quad \forall \varphi_h \in V_h, \\ (\tau_h, \varphi_h)_{\Omega_k} + (\sigma_h, (\varphi_h)_x)_{\Omega_k} - \langle \widehat{\sigma}, \varphi_h n \rangle_{\partial\Omega_k} &= 0 \quad \forall \varphi_h \in V_h, \\ (\sigma_h, \varphi_h)_{\Omega_k} + (w_h, (\varphi_h)_x)_{\Omega_k} - \langle \widehat{w}, \varphi_h n \rangle_{\partial\Omega_k} &= 0 \quad \forall \varphi_h \in V_h. \end{aligned}$$

As is customary, we have defined the abbreviations

$$(f, g)_{\Omega_k} := \sum_{\Omega_k} \int_{\Omega_k} fg dx, \quad \langle f, g \rangle_{\partial\Omega_k} := \sum_{\Omega_k} \int_{\partial\Omega_k} fg d\sigma(x).$$

In one dimension, the last term can be simplified into function evaluations at two points. However, we prefer keeping the integral on the boundary to indicate the algorithm extends to multiple dimensions. The numerical fluxes  $\widehat{\tau}$ ,  $\widehat{\sigma}$  and  $\widehat{w}$  have to be identified appropriately. One way to achieve a stable scheme is to choose ‘upwinding’ in an alternating fashion [45]. The corresponding fluxes read

$$\widehat{w} = w_h^+, \quad \widehat{\sigma} = \sigma_h^-, \quad \widehat{\tau} = \tau_h^+, \quad \widehat{\psi} = \psi_h^-, \quad (9)$$

where we stick to the convention that  $w_h^-$  refers to the left value of  $w_h$  at the interface, and  $w_h^+$  refers to the right value.

We summarize these quantities in an auxiliary variable  $\mathbf{x}_h \in V_h^4 =: \mathbf{X}_h$ , given by

$$\mathbf{x}_h := (w_h, \sigma_h, \tau_h, \varphi_h). \quad (10)$$

This simplifies the defining equations for  $\sigma_h$ ,  $\tau_h$  and  $\varphi_h$  as

$$\mathcal{N}_{\text{aux}}(\mathbf{x}_h, \varphi_h) = 0 \quad \forall \varphi_h \in V_h^3.$$

**Remark 3** (Lifting operators). *It is pointed out in [45] that it is possible to express the variable  $\sigma$  in terms of  $w$  via lifting operators, and subsequently  $\tau$  and  $\psi$  as terms of  $w$  as well. This comes at the expense of computing lifting operators (c.f. [2]).*

With these preliminaries, we now consider the semi-discretization (8) once again. It is well-known how to spatially discretize  $\mathcal{R}(w)$  using the DG method

$$\begin{aligned} (\mathcal{R}(w_h), \varphi_h)_{\Omega_k} &\approx nr(\mathbf{x}_h, \varphi_h) \\ &:= (f(w_h) - \varepsilon\sigma_h, (\varphi_h)_x)_{\Omega_k} - \left\langle \widehat{f}(w_h^+, w_h^-) - \varepsilon\widehat{\sigma}_h, \varphi_h n \right\rangle_{\partial\Omega_k} \end{aligned}$$

with the discretization of  $\sigma$  and  $\widehat{\sigma}$  as before. The numerical flux  $\widehat{f}$  denotes a standard consistent and conservative Riemann solver. Details on the chosen flux are given in the numerical results section.

The discretization of  $\mathcal{R}^2(w)$  is less straightforward. In particular, both the occurring higher derivatives and the nonlinearity of  $f$  pose severe problems. Based on the definition of  $\psi_h$ ,  $\tau_h$  and  $\sigma_h$  earlier, we propose the following discretization:

$$\begin{aligned} (\mathcal{R}^2(w_h), \varphi_h)_{\Omega_k} &\approx nrt(\mathbf{x}_h, \varphi_h) := \\ &- (f'(w_h)^2\sigma_h - \varepsilon f'(w_h)\tau_h, (\varphi_h)_x)_{\Omega_k} + \langle f'(\widehat{w}_h)^2\widehat{\sigma}_h - \varepsilon f'(\widehat{w}_h)\widehat{\tau}_h, \varphi_h n \rangle_{\partial\Omega_k} \\ &+ (\varepsilon\mathcal{D}_h^2 f, (\varphi_h)_x)_{\Omega_k} - \left\langle \varepsilon\widehat{\mathcal{D}}_h^2 f, (\varphi_h)_x \right\rangle_{\partial\Omega_k} - (\varepsilon^2\psi_h, \varphi_h)_{\Omega_k} + \left\langle \varepsilon^2\widehat{\psi}_h, \varphi_h n \right\rangle_{\partial\Omega_k}. \end{aligned}$$

Again, the fluxes  $\widehat{w}$ ,  $\widehat{\sigma}$ ,  $\widehat{\tau}$  and  $\widehat{\psi}$  are the LDG fluxes with alternating evaluation, see (9).  $\mathcal{D}_h^2 f$  denotes an approximation to  $f(w)_{xx}$ , see Remark 4.

**Remark 4** (Discretization of  $\mathcal{D}_h^2 f$ ). *The suitable discretization of  $\mathcal{D}_h^2 f \approx f(w)_{xx}$  depends on the choice of the convective flux  $f$ . We show two prototypical examples:*

1. *Linear equation, i.e.,  $f(w) = cw$ . In this case,  $f(w)_{xx} = cw_{xx}$ , and a suitable choice is*

$$\mathcal{D}_h^2 f := c\tau, \quad \widehat{\mathcal{D}}_h^2 f := c\widehat{\tau}.$$

2. *Burgers equation, i.e.,  $f(w) = \frac{1}{2}w^2$ . In this case,  $f(w)_{xx} = w_x^2 + ww_{xx}$ . As all occurring derivatives are known explicitly in the algorithm, a suitable approximation is*

$$\mathcal{D}_h^2 f := \sigma^2 + w\tau, \quad \widehat{\mathcal{D}}_h^2 f := \widehat{\sigma}^2 + \widehat{w}\widehat{\tau}.$$

3. *A similar procedure as with Burgers equation is possible with any flux function - also for Euler equation. However, the result might become increasingly complex.*

Ultimately, this leads to the formulation of the full algorithm, summarized in the following definition:

**Definition 1** (Numerical method). *Let  $\varphi_h = (\varphi_h^{(1)}, \varphi_h^{(2)}) \in \mathbf{X}_h$  with  $\varphi_h^{(1)} \in V_h^3$ . Furthermore, let the semi-linear form  $\mathcal{N}$  be given by*

$$\mathcal{N}(\mathbf{x}_h, \varphi_h) := \begin{pmatrix} \mathcal{N}_{aux}(\mathbf{x}_h, \varphi_h^{(1)}) \\ \mathcal{N}_{eq}(\mathbf{x}_h, \varphi_h^{(2)}) \end{pmatrix}$$

where  $\mathcal{N}_{eq}(\mathbf{x}_h, \varphi_h^{(2)})$  is given by

$$\begin{aligned} \mathcal{N}_{eq}(\mathbf{x}_h, \varphi_h^{(2)}) &:= \alpha_1 \mathcal{N}_{\mathcal{R}}(\mathbf{x}_h^n, \varphi_h^{(2)}) + \alpha_2 \mathcal{N}_{\mathcal{R}}(\mathbf{x}_h^{n+1}, \varphi_h^{(2)}) \\ &+ \Delta t \left( \beta_1 \mathcal{N}_{\mathcal{R}^2}(\mathbf{x}_h^n, \varphi_h^{(2)}) + \beta_2 \mathcal{N}_{\mathcal{R}^2}(\mathbf{x}_h^{n+1}, \varphi_h^{(2)}) \right). \end{aligned}$$

The approximate solution  $\mathbf{x}_h^{n+1} = (w_h^{n+1}, \sigma_h^{n+1}, \tau_h^{n+1}, \psi_h^{n+1}) \in \mathbf{X}_h$  is given as the solution to the problem

$$\begin{pmatrix} \mathbf{0} \\ \frac{1}{\Delta t} \left( w_h^{n+1} - w_h^n, \varphi_h^{(2)} \right)_{\Omega_k} \end{pmatrix} = \mathcal{N}(\mathbf{x}_h, \varphi_h) \quad \forall \varphi_h \in \mathbf{X}_h.$$

The following lemma is a straightforward consequence of both the order of accuracy of the ODE integrator and the consistency of the underlying DG schemes:

**Lemma 3** (Consistency in time). *The algorithm is consistent with the order of the temporal integration scheme chosen in (8), i.e., there holds:*

$$\left( \frac{1}{\Delta t} (w(\cdot, t^{n+1}) - w(\cdot, t^n), \varphi_h)_{\Omega_k} \right) - \mathcal{N}(\mathbf{x}, \varphi_h) = \mathcal{O}(\Delta t^q)$$

where  $q = 3$  for integrator (5) and  $q = 4$  for integrator (6).

**Lemma 4** (Conservation). *The algorithm is both locally and globally conservative if  $\widehat{\mathcal{D}}_h^2 f$  is conservative.*

*Proof.* Testing with a piecewise constant test function yields that the integral of  $w_h^{n+1}$  only depends on the fluxes over the boundaries. This yields local conservation. Noting that the fluxes are conservative and testing against a constant function yields that the algorithm is globally conservative.  $\square$   $\square$

## 5 1D: Numerical results

In this section, we present numerical results for the newly developed scheme. In each case, we demonstrate the optimal order of convergence through.

In all our computations, we use periodic boundary conditions on the unit interval  $\Omega := [0, 1]$ , and compute until a final time of  $T = 0.5$ . For the cases involving linear convection, we choose the upwind numerical flux

$$\widehat{f}(w^+, w^-) := cw^-, \quad c > 0,$$

whereas for Burgers equation, we use a local Lax-Friedrichs flux. The domain  $\Omega$  is subdivided into equally spaced intervals with spacing  $h$ . As an error measure, we compute the  $L^2$ -error at time  $T$ , that is, we define the error as

$$e_h := \|w(\cdot, T) - w_h(\cdot, T)\|_{L^2(\Omega)},$$

where  $w$  is the exact and  $w_h$  the approximate solution to the underlying problem.

### 5.1 Heat equation

The first equation to be considered is the pure heat equation

$$w_t = \varepsilon w_{xx} \quad \forall (x, t) \in \Omega \times (0, T)$$

with initial conditions  $w_0(x) = \sin(2\pi x)$ , and  $\varepsilon = 0.1$ .

Numerical results for different values of the polynomial order  $p$  of the ansatz space are shown in Fig. 1 for the third-order integrator (5) (left) and the fourth-order integrator (6) (right). The expected order of accuracy of  $\max\{p + 1, 3\}$  and  $\max\{p + 1, 4\}$ , respectively, is achieved. The time step is set to  $\Delta t = \Delta x$ . Experiments with other ratios of  $\frac{\Delta t}{\Delta x}$  introduce no stability problems, which is independent on the choice of  $\varepsilon$ . Thus, we conjecture that the algorithm is uniformly stable for this simple 1D test case without transport.

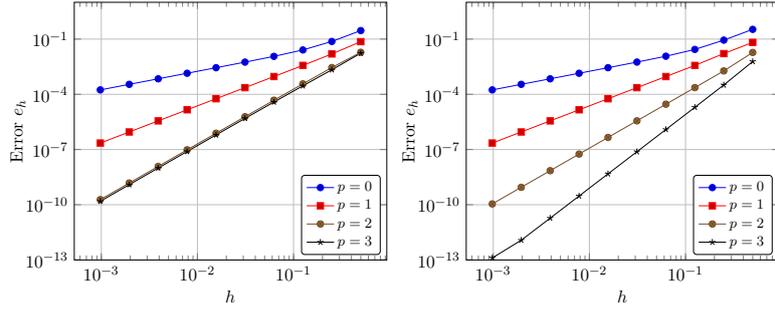


Figure 1: Numerical results for the heat equation. In both computations, we choose the ratio  $\frac{\Delta t}{\Delta x}$  to be one. Temporal integration is performed via the third-order accurate integrator (5) (left) and the fourth-order integrator (6) (right).

## 5.2 Convection equation

Next, we test the algorithm on the pure convection equation

$$w_t + cw_x = 0 \quad \forall (x, t) \in \Omega \times (0, T),$$

again with initial conditions  $w_0(x) = \sin(2\pi x)$ , and constant  $c = 1$ . Numerical results are displayed in Fig. 2, again for the third-order (left) and the fourth-order (right) temporal integrator. The CFL number for this example is one for the third-order integrator, and only 0.1 for the fourth-order integrator. The reason for this choice is that we find stability constraints with the fourth-order integrator. Our experience with other time integrators has led to this in the past, and we suspect that it is most likely due to the loss of  $L$ -stability in the fourth-order solver. Numerical experiments indicate that the third-order integrator is uniformly stable. We note that obviously, such a severe CFL restriction is not a desired feature of an implicit scheme, and a detailed investigation into how to fix the fourth-order scheme is the subject of future work.

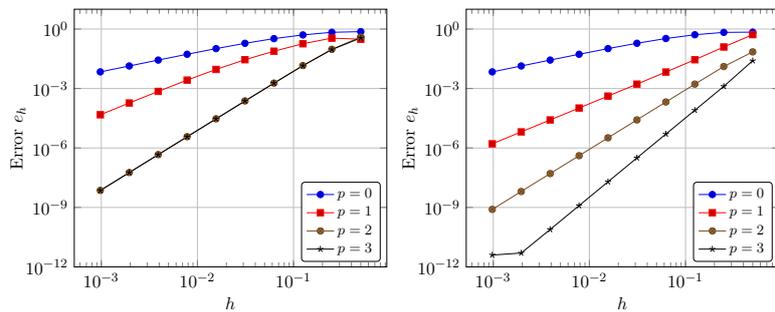


Figure 2: Numerical results for the convection equation. Temporal integration is done via the third-order integrator (5) (left) and the fourth-order integrator (6) (right). The ratio  $\frac{\Delta t}{\Delta x}$  is set to be 1.0 (left) and 0.1 (right). The third-order integrator seems to be uniformly stable, yet the fourth-order integrator is not, which is why we find it necessary to reduce the CFL number.

### 5.3 Convection-diffusion equation

The final linear single-dimensional test case is the convection-diffusion equation

$$w_t + cw_x = \varepsilon w_{xx} \quad \forall (x, t) \in \Omega \times (0, T),$$

with values  $c = 1$  and  $\varepsilon = 0.1$ . This exercises the ability of the scheme to correctly account for the additional coupling terms that arise in the discretization of  $w_{tt}$ . We present two examples: a) an example with a smooth initial profile, and b) a problem with discontinuous initial conditions.

#### 5.3.1 Convection-diffusion: Smooth initial conditions.

We present numerical results for this problem in Fig. 3. The initial conditions for these results are given by  $w_0(x) = \sin(2\pi x)$ , and the ratio  $\frac{\Delta t}{\Delta x}$  is chosen to be one for all cases. For the third-order integrator, we do not observe any stability issues, while obviously, from our experiences with the convection equation, the fourth-order scheme tends to be only conditionally stable. However, this stability condition does not seem to depend on the magnitude of the viscosity coefficient  $\varepsilon$ , but rather on the ratio  $\frac{c}{\varepsilon}$ .

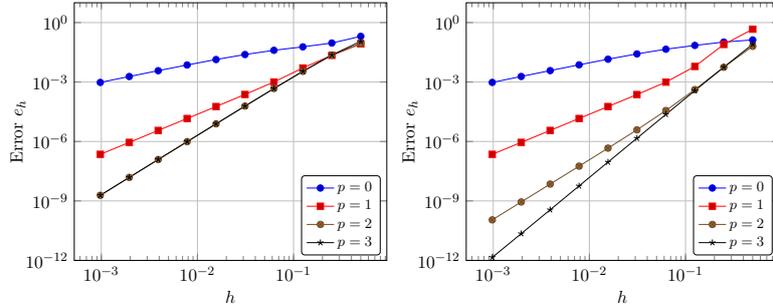


Figure 3: Numerical results for the convection-diffusion equation. In both computations, we choose the ratio  $\frac{\Delta t}{\Delta x}$  to be one. Temporal integration is done via the third-order integrator (5) (left) and the fourth-order integrator (6) (right).

#### 5.3.2 Convection-diffusion: Discontinuous initial conditions.

In this section, we consider discontinuous initial data

$$w_0(x) = \mathcal{H}(\sin(2\pi(x - 0.3)))e^{\sin(2\pi x)},$$

where  $\mathcal{H}$  denotes the Heaviside function.

Because the initial conditions are not smooth, there are at least two ways of defining the initial conditions for the auxiliary variables  $\mathbf{x}_h^0$  in Eqn. (10) that require the spatial derivatives of the initial conditions. One way is to fill it with the given initial conditions, i.e., set  $\sigma_h^0 = \Pi_{V_h} w_0' \dots$  where  $\Pi_{V_h}$  denotes the  $L^2$ -projection onto  $V_h$ . We use this choice for the computations in the previous sections, because the initial datum is smooth. However, for this problem that has non-smooth initial conditions,  $w_0'$  does not exist. For this case, one alternative is to compute  $\sigma_h^0, \tau_h^0$  and  $\psi_h^0$  as a solution to  $\mathcal{N}_{\text{aux}}(\mathbf{x}_h^0, \varphi_h) = 0$  for all  $\varphi_h \in V_h^3$  for a given  $w_h = \Pi_{V_h} w_0$ . This is the ansatz we pursue in this section.

In Fig. 4, we show an approximate solution at time  $T = 0.5$ , which uses a spatial width of  $h = \frac{1}{16}$ , quadratic (third-order) polynomials, CFL number of  $\frac{\Delta t}{\Delta x} = 0.5$  and the third-order ODE integrator. We observe a strong agreement between the exact and approximate solution.

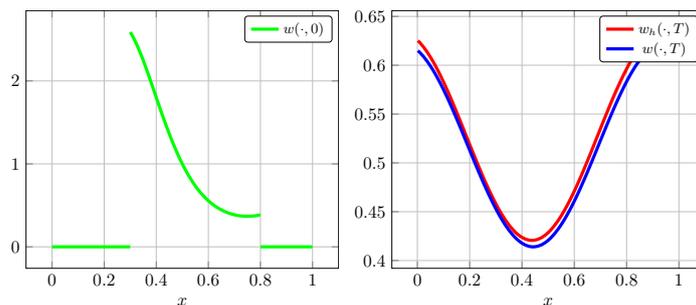


Figure 4: Approximate versus exact solution at time  $T = 0.5$  (right) to the convection-diffusion equation with discontinuous initial datum (left). Results are computed with  $p = 2$ ,  $h = \frac{1}{16}$  and the third-order integrator (5).

## 5.4 Viscous Burgers equation

Our final single dimensional numerical result is the nonlinear Burgers equation

$$w_t + f(w)_x = \varepsilon w_{xx}$$

with  $f(w) = 0.5w^2$  and  $\varepsilon = 0.1$ . Equipped with initial conditions  $w_0(x) = \sin(2\pi x)$ , this test case has a smooth solution  $w$  for all times  $T$ . As before, we show convergence results in Fig. 5. The exact solution is computed using the Cole-Hopf transformation [25]. No stability issues are observed in the computations, and the plots show perfect order of convergence. The results are similar to those of the convection-diffusion equation, which is mainly because diffusion is dominant in this test case. For implicit methods, this is probably the most relevant case, as for purely hyperbolic problems, explicit methods are often times the preferred method of choice given the finite speed of propagation of information. Furthermore, we do not implement any limiters but a final algorithm should have a suitable way of stabilizing discontinuities in the case of a convection dominated problem. This is one topic of future research, where one option is to introduce artificial viscosity into the time stepping, because we already have access to these higher derivatives.

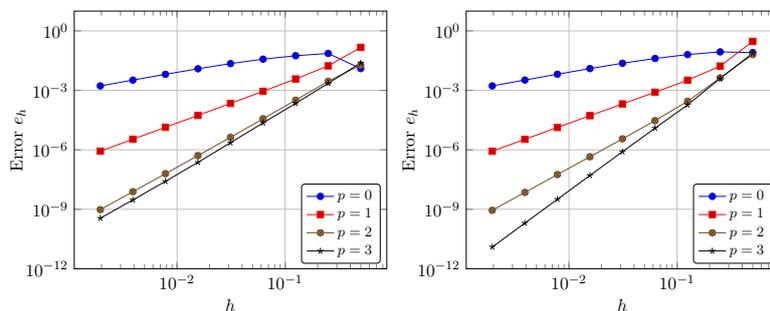


Figure 5: Numerical results for Burgers equation. In both computations, we choose the ratio  $\frac{\Delta t}{\Delta x}$  to be one. Temporal integration is done via the third-order integrator (5) (left) and the fourth-order integrator (6) (right).

## 6 2D: Extensions to multiple dimensions

In this section, we describe the extension of implicit two-derivative multistage methods to hyperbolic systems of first-order PDEs

$$\begin{aligned}\mathbf{w}_t + \nabla \cdot \mathbf{f}(\mathbf{w}) &= 0 & \forall (\mathbf{x}, t) \in \Omega \times \mathbb{R}^+ \\ \mathbf{w}(\mathbf{x}, 0) &= \mathbf{w}_0(\mathbf{x}) & \forall \mathbf{x} \in \Omega\end{aligned}\tag{11}$$

on a domain  $\Omega \subset \mathbb{R}^2$  with appropriate boundary conditions. In general, the unknown  $\mathbf{w}$  is a function of space and time  $\mathbf{w} := \mathbf{w}(\mathbf{x}, t)$ , but we may drop  $\mathbf{x}$  and  $t$  for a more compact notation. We allow  $\mathbf{f}(\mathbf{w})$  to be a general, possibly nonlinear, flux. Note, that the flux  $\mathbf{f}(\mathbf{w})$  (as well as  $\mathbf{w}$ ) is a vector in  $\mathbb{R}^m$  for a system with a total of  $m$  equations. In order to limit the complexity we stay with first-order PDEs for the 2D case. Due to the additional spatial dimension, the total number of derivatives also doubles which would make the assembly of the matrices required for second or higher order PDEs tedious, especially in the case of a system of equations. Nevertheless, the two-derivative methods are still applicable to higher order PDEs in 2D using the techniques from Section 4.

Here, we apply the same third and fourth-order two derivative methods defined in Eqns. (5)-(6), but we discretize the resulting system using the *hybridized discontinuous Galerkin* [7, 32, 39] method. For Poisson's equation, this discretization is equivalent to the LDG method [7]. A preliminary investigation that couples two-derivative Runge-Kutta methods with HDG for a linear advection equation can be found in in [27]. The HDG method usually leads to a much smaller system of globally coupled equations than the LDG approach. This is especially beneficial in the current case where additional unknowns are introduced by the spatial derivatives from the two-derivative time discretization. We aim to stress at this point that the additional unknowns are, for the HDG method, only *local* unknowns, and the globally coupled degrees of freedom are not affected.

The representations of  $\mathcal{R}(\mathbf{w})$  and  $\mathcal{R}^2(\mathbf{w})$  only differ slightly from the one dimensional DG case. Here, we find that the first and second derivatives are given by

$$\begin{aligned}\mathbf{w}_t &= -\nabla \cdot \mathbf{f}(\mathbf{w}) =: \mathcal{R}(\mathbf{w}) \\ \mathbf{w}_{tt} &= \nabla \cdot (\mathbf{f}'(\mathbf{w}) \nabla \cdot \mathbf{f}(\mathbf{w})) =: \mathcal{R}^2(\mathbf{w})\end{aligned}\tag{12}$$

which follows directly from Eqn. (3) by setting  $\varepsilon = 0$ .

We follow the discretization procedure of Section 4. The semi-discrete system again reads

$$\mathbf{w}^{n+1} = \mathbf{w}^n + \Delta t (\alpha_1 \mathcal{R}(\mathbf{w}^n) + \alpha_2 \mathcal{R}(\mathbf{w}^{n+1})) + \Delta t^2 (\beta_1 \mathcal{R}^2(\mathbf{w}^n) + \beta_2 \mathcal{R}^2(\mathbf{w}^{n+1}))$$

with  $\mathcal{R}(\mathbf{w})$  and  $\mathcal{R}^2(\mathbf{w})$  defined in Eqn. (12). For the spatial discretization we triangulate the domain such that

$$\Omega = \bigcup_{k=1}^{N_e} \Omega_k.$$

The hybridized DG method requires a description of the edges. We refer to edges of two intersecting elements and elements intersecting the domain boundary  $\partial\Omega$  with  $e_k$ . The set of all edges is  $\Gamma$  and its number of elements is  $N_f := |\Gamma|$ . This is needed to introduce a new hybrid unknown  $\lambda = \mathbf{w}|_{\Gamma}$  that represents the solution evaluated on the trace of each element. This allows us to reduce the size of the globally coupled system by using static condensation [7]. For the approximation of  $\lambda_h \approx \lambda$  we need to introduce the ansatz space  $M_h$  that consists of edge-wise polynomials of degree  $p$  defined by

$$M_h := \{q \in L^2(\Gamma) \mid q|_{e_k} \in \Pi^p(e_k) \forall k = 1, \dots, N_f, e_k \in \Gamma\}^m.$$

For the approximation of second order spatial derivatives introduced by a two-derivative time discretization, we again define an auxiliary variable through

$$\sigma := \nabla \mathbf{w}.$$

The ansatz spaces for  $\sigma_h$  and  $\mathbf{w}_h$  are the common spaces

$$\begin{aligned} H_h &:= \{q \in L^2(\Omega) \mid q|_{\Omega_k} \in \Pi^p(\Omega_k) \forall k = 1, \dots, N_e\}^{2m}, \\ V_h &:= \{q \in L^2(\Omega) \mid q|_{\Omega_k} \in \Pi^p(\Omega_k) \forall k = 1, \dots, N_e\}^m, \end{aligned}$$

that contain all polynomials of degree at most  $p$ . In order to condense notation, we define the vector of unknowns as

$$\mathbf{x}_h := (\mathbf{w}_h, \sigma_h, \lambda_h)$$

that come from the ansatz space  $\mathbf{X}_h := H_h \times V_h \times M_h$ , and corresponding test functions  $\varphi_h = (\varphi^{(1)}, \varphi^{(2)}, \varphi^{(3)}) \in \mathbf{X}_h$ . Then,  $\sigma_h$  is approximated through

$$\left( \sigma_h, \varphi_h^{(1)} \right)_{\Omega_k} + \left( \mathbf{w}_h, \nabla \cdot (\varphi_h^{(1)}) \right)_{\Omega_k} - \left\langle \lambda_h, \varphi_h^{(1)} \cdot \mathbf{n} \right\rangle_{\partial\Omega_k} = 0 \quad \forall \varphi_h^{(1)} \in H_h,$$

that is very similar to the approximation before, but we use the hybrid variable  $\lambda_h$  as the numerical flux  $\widehat{\mathbf{w}} := \lambda_h$ . The equation is abbreviated by

$$\mathcal{N}_{\text{aux}}(\mathbf{x}_h, \varphi_h^{(1)}) = 0 \quad \forall \varphi_h^{(1)} \in H_h.$$

Finally, the discretization of  $\mathcal{R}$  and  $\mathcal{R}^2$  for this first-order PDE is given by

$$\left( \mathcal{R}(\mathbf{w}_h), \varphi_h^{(2)} \right)_{\Omega_k} \approx \mathcal{N}_{\mathcal{R}}(\mathbf{x}_h, \varphi_h^{(2)}) := \left( \mathbf{f}(\mathbf{w}_h), \nabla \varphi_h^{(2)} \right)_{\Omega_k} - \left\langle \widehat{\mathbf{f}}, \varphi_h^{(2),-} \cdot \mathbf{n} \right\rangle_{\partial\Omega_k}$$

and

$$\begin{aligned} \left( \mathcal{R}^2(\mathbf{w}_h), \varphi_h^{(2)} \right)_{\Omega_k} &\approx \mathcal{N}_{\mathcal{R}^2}(\mathbf{x}_h, \varphi_h^{(2)}) \\ &:= - \left( \mathcal{D}_h \mathbf{f}(\mathbf{w}_h, \sigma_h), \nabla \varphi_h^{(2)} \right)_{\Omega_k} + \left\langle \widehat{\mathcal{D}_h \mathbf{f}}, \varphi_h^{(2),-} \cdot \mathbf{n} \right\rangle_{\partial\Omega_k}, \end{aligned}$$

where

$$\mathbf{f}'(\mathbf{w}) \nabla \cdot \mathbf{f}(\mathbf{w}) = \mathbf{f}'(\mathbf{w}) \mathbf{f}'_i(\mathbf{w}) \partial_{x_i} \mathbf{w} \approx \mathbf{f}'(\mathbf{w}_h) \mathbf{f}'_i(\mathbf{w}_h) \sigma_{h,i} =: \mathcal{D}_h \mathbf{f}(\mathbf{w}_h, \sigma_h).$$

Thus,  $\sigma_h$  is involved when the flux is evaluated. On each element interface, we insert numerical fluxes

$$\begin{aligned} \widehat{\mathbf{f}} &= \mathbf{f}(\lambda_h) + \eta(\mathbf{w}_h^- - \lambda_h) \mathbf{n} \\ \widehat{\mathcal{D}_h \mathbf{f}} &= \widehat{\mathcal{D}_h \mathbf{f}}(\lambda_h, \sigma_h^-) - \theta(\mathbf{w}_h^- - \lambda_h) \mathbf{n}, \end{aligned}$$

that are modified Lax-Friedrichs (Rusanov) fluxes with  $\eta$  and  $\theta$  being stabilization parameters. Whenever  $\theta$  is multiplied with a negative coefficient from the time integrator, we invert the sign. Note, that at this point the equations are only coupled through the hybrid variable  $\lambda_h$ . An additional equation arises from the additional unknown  $\lambda_h$  through

$$\left\langle -\widehat{\mathbf{f}} + \widehat{\mathcal{D}_h \mathbf{f}}, \varphi_h^{(3)} \right\rangle_{\partial\Omega_k} = 0 \quad \forall \varphi_h^{(3)} \in M_h.$$

With these preliminaries in place, we are now ready to define the full hybridized DG method.

**Definition 2** (HDG method). Let  $\varphi_h = \mathbf{X}_h$ . Furthermore, let the semi-linear form  $\mathcal{N}$  be given by

$$\mathcal{N}(\mathbf{x}_h, \varphi_h) := \begin{pmatrix} \mathcal{N}_{aux}(\mathbf{x}_h, \varphi_h^{(1)}) \\ \mathcal{N}_{eq}(\mathbf{x}_h, \varphi_h^{(2)}) \\ \mathcal{N}_{hyb}(\mathbf{x}_h, \varphi_h^{(3)}) \end{pmatrix},$$

where  $\mathcal{N}_{eq}(\mathbf{x}_h, \varphi_h^{(2)})$  is given by

$$\begin{aligned} \mathcal{N}_{eq}(\mathbf{x}_h, \varphi_h^{(2)}) &:= \alpha_1 \mathcal{N}_{\mathcal{R}}(\mathbf{x}_h^n, \varphi_h^{(2)}) + \alpha_2 \mathcal{N}_{\mathcal{R}}(\mathbf{x}_h^{n+1}, \varphi_h^{(2)}) \\ &\quad + \Delta t \left( \beta_1 \mathcal{N}_{\mathcal{R}^2}(\mathbf{x}_h^n, \varphi_h^{(2)}) + \beta_2 \mathcal{N}_{\mathcal{R}^2}(\mathbf{x}_h^{n+1}, \varphi_h^{(2)}) \right), \end{aligned}$$

and the hybrid term is given by

$$\mathcal{N}_{hyb}(\mathbf{x}_h, \varphi_h^{(3)}) := \left\langle \llbracket -\alpha_2 \widehat{\mathbf{f}}^{n+1} + \beta_2 \widehat{\mathcal{D}}_h \mathbf{f}^{n+1} \rrbracket, \varphi_h^{(3)} \right\rangle_{\partial\Gamma}.$$

The brackets denote the jump operator

$$\llbracket \mathbf{v} \rrbracket = \mathbf{v}^- \mathbf{n} - \mathbf{v}^+ \mathbf{n}$$

with  $v(\mathbf{x})^\pm$  being

$$\mathbf{v}(\mathbf{x})^\pm = \lim_{\epsilon \rightarrow 0} \mathbf{v}(\mathbf{x} \pm \epsilon \mathbf{n}), \quad \mathbf{x} \in \partial\Omega_k \quad (13)$$

where  $\mathbf{n}$  is the outward pointing normal. The approximate solution  $\mathbf{x}_h^{n+1} = (\sigma_h^{n+1}, \mathbf{w}_h^{n+1}, \lambda_h^{n+1}) \in \mathbf{X}_h$  is given as the solution to the problem

$$\begin{pmatrix} \mathbf{0} \\ \frac{1}{\Delta t} \left( \mathbf{w}_h^{n+1} - \mathbf{w}_h^n, \varphi_h^{(2)} \right)_{\Omega_k} \\ \mathbf{0} \end{pmatrix} = \mathcal{N}(\mathbf{x}_h, \varphi_h) \quad \forall \varphi_h \in \mathbf{X}_h.$$

**Remark 5** (Number of unknowns). All evaluations on elements only depend on local values of  $\mathbf{w}_h$  and  $\sigma_h$ , and therefore the total number of unknowns can be significantly reduced when compared to a classical LDG method. The coupling between elements is achieved solely by the hybrid variable  $\lambda_h$ , and therefore, the system to be solved for globally can be condensed [7]. This means that the resulting system is usually much smaller than it would be for the standard LDG approach, which typically requires solving simple local problems on each element in an element-wise fashion.

## 7 2D: Numerical results

In this section we show two-dimensional numerical results. Here, we solve the (nonlinear) system of equations using Newton's method. The resulting linear system is solved using GMRES with block Jacobi preconditioning until the relative residual drops below  $10^{-12}$ . Newton's method is carried out until the  $L_2$ -norm of the residual drops below  $10^{-10}$ .

### 7.1 Linear advection equation

We first examine a system of linear advection equations. It can be written as in Eqn. (11). We compute a solution on  $\Omega = [0, 2]^2$  at final time  $T = 0.1$ . The flux is chosen to be  $\mathbf{f}(\mathbf{w}) = (\mathbf{f}_1, \mathbf{f}_2)$  with

$$\mathbf{f}_1(\mathbf{w}) = \mathbf{A}_1 \mathbf{w}, \quad \mathbf{f}_2(\mathbf{w}) = \mathbf{A}_2 \mathbf{w}, \quad (14)$$

The vector of unknowns is  $\mathbf{w} = (w_1, w_2)^T$ . The matrices for this linear system are given by

$$\mathbf{A}_1 = \begin{pmatrix} \frac{1}{3} & \frac{8}{3} \\ \frac{16}{3} & -\frac{7}{3} \end{pmatrix}, \mathbf{A}_2 = \begin{pmatrix} -\frac{7}{3} & -\frac{5}{3} \\ -\frac{10}{3} & -\frac{2}{3} \end{pmatrix}. \quad (15)$$

These matrices have the same eigenvector basis, which means we can express these as  $\mathbf{A}_1 = \mathbf{S}\mathbf{D}_{A_1}\mathbf{S}^{-1}$  and  $\mathbf{A}_2 = \mathbf{S}\mathbf{D}_{A_2}\mathbf{S}^{-1}$  with

$$\mathbf{D}_{A_1} = \begin{pmatrix} -5 & 0 \\ 0 & 3 \end{pmatrix}, \mathbf{D}_{A_2} = \begin{pmatrix} 1 & 0 \\ 0 & -4 \end{pmatrix}, \mathbf{S} = \begin{pmatrix} -\frac{1}{2} & 1 \\ 1 & 1 \end{pmatrix}, \mathbf{S}^{-1} = \begin{pmatrix} -\frac{2}{3} & \frac{2}{3} \\ \frac{2}{3} & \frac{1}{3} \end{pmatrix}. \quad (16)$$

After choosing the initial conditions to be

$$\mathbf{w}_0(\mathbf{x}) = \begin{pmatrix} \sin(\pi(x+y)) \\ \sin(\pi(x+y)) \end{pmatrix}, \quad (17)$$

and taking into account periodic boundary conditions, we write the exact solution as

$$\mathbf{w}(\mathbf{x}, t) = \begin{pmatrix} \sin(\pi(x+y+t)) \\ \sin(\pi(x+y+t)) \end{pmatrix}. \quad (18)$$

We compute solutions on meshes that are presented in Fig. 6. Results are presented in Fig. 7 for the

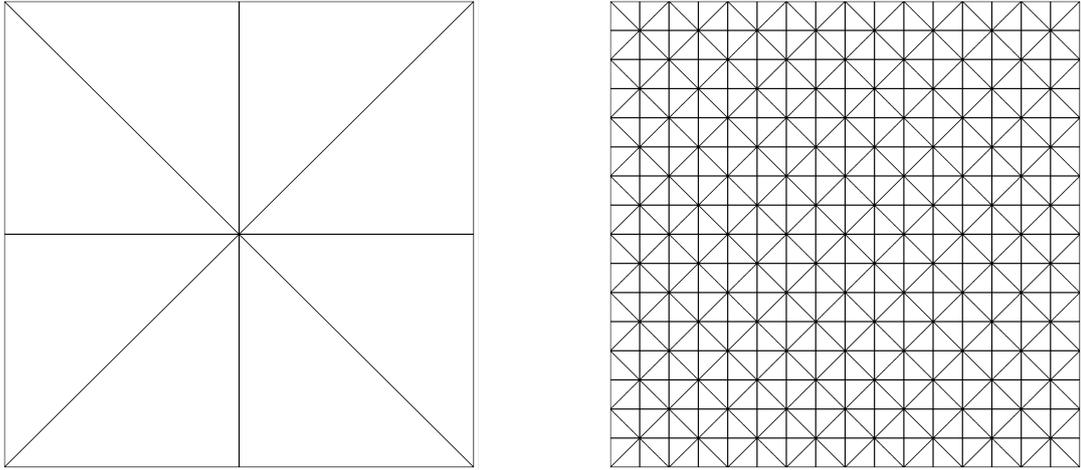


Figure 6: Left : Initial mesh. Right: Mesh after a total of three refinements.

ratio  $\frac{\Delta t}{\Delta x} = 0.025$ . The errors for  $w_1$  and  $w_2$  are perfectly identical. The third order integrator reaches the expected order of convergence in all cases. For  $p = 3$ , the method is still third-order accurate, but it has a lower error than in the case with  $p = 2$ . The fourth-order integrator, however, does not achieve fourth-order in time. In the case  $p < 3$ , the method gets close to the expected order of  $p + 1$  while for  $p = 3$  the order deteriorates during the refinements. After the sixth refinement it seems not to converge any further. Most likely, this behavior is observed due to stability issues of the fourth-order integrator.

### 7.1.1 Euler equations

As second test case in two space dimensions, we solve the Euler equations with periodic boundary conditions. The flux  $\mathbf{f}(\mathbf{w}) = (\mathbf{f}_1, \mathbf{f}_2)$  is nonlinear and is given by

$$\begin{aligned} \mathbf{f}_1(\mathbf{w}) &= (\rho u, P + \rho u^2, \rho uv, u(E + P))^T, \\ \mathbf{f}_2(\mathbf{w}) &= (\rho v, \rho uv, P + \rho v^2, v(E + P))^T, \end{aligned} \quad (19)$$

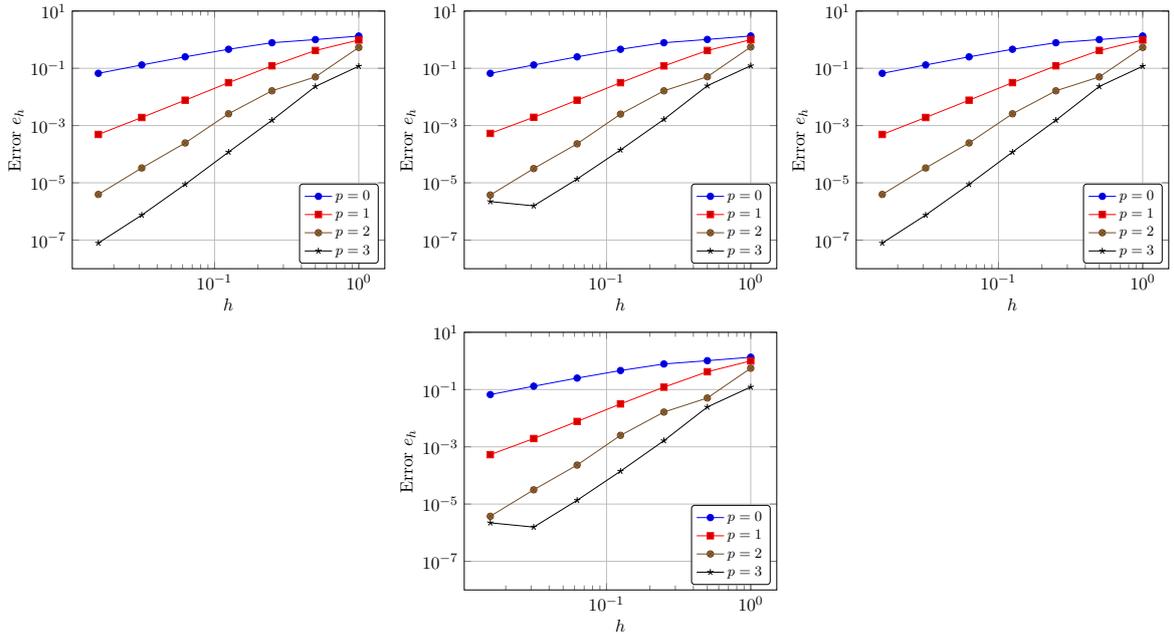


Figure 7: Numerical results for the linear coupled advection equation. Temporal integration is performed with the third-order integrator (5) (left) and the fourth-order (6) integrator (right). We show the results for components  $w_1$  (top) and  $w_2$  (bottom). In all computations we choose the ratio  $\frac{\Delta t}{\Delta x}$  to be 0.025 to ensure stability of the numerical method.

and the vector of unknowns is  $\mathbf{w} = (\rho, \rho u, \rho v, E)$ , which define the density  $\rho$ , momentum  $\rho u$  and  $\rho v$  in the  $x$ - and  $y$ -direction, and energy  $E$ . The pressure  $P$  is given by the equation of state

$$P = (\gamma - 1) \left( E - \frac{1}{2} \rho (u^2 + v^2) \right),$$

and the ratio of specific heats is  $\gamma = 1.4$  for the test cases considered in this work. To analyze the accuracy of the method, we make use of a test case similar to the one presented in [28] that has a smooth analytical solution. The domain  $\Omega = [0, 2]^2$  is equipped with periodic boundary conditions, and the initial conditions are

$$\rho(x, y) = 1.0 + 0.2 \sin(\pi(x + y)), \quad u = 0.7, \quad v = 0.3, \quad P = 1. \quad (20)$$

A convergence study is presented in Fig. 8, where we compute the solution to a final time of  $T = 0.5$ .

Both integrators produce very similar errors, but the third-order integrator has slightly lower errors. The conclusion is that the higher-order integrator does not exhibit any serious advantage over the lower-order integrator for this test case. For this problem, we find that increasing the polynomial order always increases the rate of convergence, which is in contrast to the previous cases. For example, in the linear test case, going from  $p = 2$  to  $p = 3$  decreased the error level, whereas the slope of the error graph stayed almost constant (c.f. Fig. 7). For this problem, this actually *increases* the slope. Nevertheless, both integrators have a slight loss of convergence rate during refinements.

## 8 Conclusions and outlook

In this work, we present a novel application of high-order, implicit multiderivative time integrators to the discontinuous Galerkin framework. Two dimensional results are realized by employing the hybridized

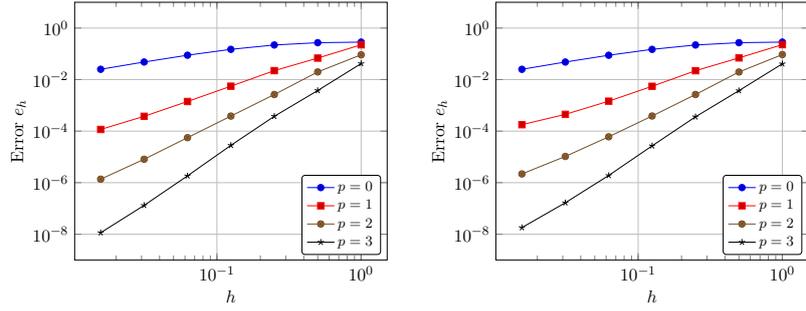


Figure 8: Numerical results for the Euler equations. Temporal integration is done via the third-order integrator (5) (left) and the fourth-order (6) integrator (right). We show the error in the density  $\rho$ . In all computations we choose the ratio  $\frac{\Delta t}{\Delta x}$  to be 0.05 to ensure stability of the methods.

discontinuous Galerkin method in order to reduce the total number of unknowns that would otherwise be required to discretize the system. Results for a third- and fourth-order time integrator are presented, where we observe the expected order of convergence in time for all of our 1D test cases. However, we numerically observe that the time integrators are not uniformly stable for convection dominated problems, which is not a desirable property for an implicit scheme. This behavior can also be observed in a more strict sense in the 2D case, where we additionally observe a loss of accuracy for the fourth-order integrator. One possible explanation for this drawback is that higher derivatives carry negative coefficients, and effectively introduce anti-diffusion into the method. Future work must deal with improving stability properties of these methods.

## References

- [1] R. Alexander. Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s. *SIAM J. Numer. Anal.*, 14:1006–1021, 1977.
- [2] Douglas N. Arnold, Franco Brezzi, Bernardo Cockburn, and L. Donatella Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal of Numerical Analysis*, 39:1749–1779, 2002.
- [3] J. Banks and W. Henshaw. Upwind schemes for the wave equation in second-order form. *Journal of Computational Physics*, 231:5854–5889, 2012.
- [4] M. F. Causley, H. Cho, A. J. Christlieb, and D.C. Seal. Method of lines transpose: High order L-stable  $O(N)$  schemes for parabolic equations using successive convolution. *arXiv preprint arXiv:1508.03105*, 2015.
- [5] Andrew J. Christlieb, Xiao Feng, David C. Seal, and Qi Tang. A high-order positivity-preserving single-stage single-step method for the ideal magnetohydrodynamic equations. *arXiv preprint arXiv:1509.09208*, 2015.
- [6] Andrew J. Christlieb, Sigal Gottlieb, Zachary J. Grant, and David C Seal. Explicit strong stability preserving multistage two-derivative time-stepping schemes. *arXiv preprint arXiv:1504.07599*, 2015.
- [7] B. Cockburn and J. Gopalakrishnan. A characterization of hybridized mixed methods for second order elliptic problems. *SIAM Journal on Numerical Analysis*, 42:283–301, 2004.

- [8] B. Cockburn, S. Hou, and C.-W. Shu. The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. *Mathematics of Computation*, 54:545–581, 1990.
- [9] B. Cockburn and S. Y. Lin. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems. *Journal of Computational Physics*, 84:90–113, 1989.
- [10] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. *Mathematics of Computation*, 52:411–435, 1988.
- [11] B. Cockburn and C.-W. Shu. The Runge-Kutta local projection  $p^1$ -discontinuous Galerkin finite element method for scalar conservation laws. *RAIRO Mathematical modelling and numerical analysis*, 25:337–361, 1991.
- [12] B. Cockburn and C.-W. Shu. The Runge-Kutta discontinuous Galerkin Method for conservation laws V: Multidimensional Systems. *Mathematics of Computation*, 141:199–224, 1998.
- [13] J. Donéa, S. Giuliani, and J. P. Halleux. Taylor-Galerkin methods for the wave equation. In *Numerical methods for nonlinear problems, Vol. 2 (Barcelona, 1984)*, pages 813–825. Pineridge, Swansea, 1984.
- [14] Jean Donéa. A Taylor-Galerkin method for convective transport problems. In *Numerical methods in laminar and turbulent flow (Seattle, Wash., 1983)*, pages 941–950. Pineridge, Swansea, 1983.
- [15] M. Dumbser. *Arbitrary High Order Schemes for the Solution of Hyperbolic Conservation Laws in Complex Domains*. Shaker Verlag, Aachen, 2005.
- [16] M. Dumbser and C. D. Munz. Building blocks for arbitrary high order discontinuous Galerkin schemes. *Journal of Scientific Computing*, 27(1-3):215–230, 2006.
- [17] R. Dyson. Technique for very high order nonlinear simulation and validation. *Journal of Computational Acoustics*, 211, 2002.
- [18] Wei Guo, Jing-Mei Qiu, and Jianxian Qiu. A New Lax–Wendroff Discontinuous Galerkin Method with Superconvergence. *J. Sci. Comput.*, 65(1):299–326, 2015.
- [19] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I*. Springer Series in Computational Mathematics, 1987.
- [20] E. Hairer and G. Wanner. Multistep-multistage-multiderivative methods of ordinary differential equations. *Computing (Arch. Elektron. Rechnen)*, 11(3):287–303, 1973.
- [21] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer Series in Computational Mathematics, 1991.
- [22] A. Harten, B. Enquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71:231–303, 1987.
- [23] W. Henshaw. A high-order accurate parallel solver for Maxwell’s equations on overlapping grids. *Siam Journal on Scientific Computing*, 28(5):1730–1765, 2006.
- [24] W. Henshaw, H.-O. Kreiss, and L. Reyna. A fourth-order-accurate difference approximation for the incompressible Navier-Stokes equations. *Computers Fluids*, 23(4):575–593, 1994.

- [25] E. Hopf. The partial differential equation  $u_t + uu_x = \mu u_{xx}$ . *Communications on Pure and Applied Mathematics*, 3:201–230, 1950.
- [26] A. Jaust and J. Schütz. A temporally adaptive hybridized discontinuous Galerkin method for time-dependent compressible flows. *Computers and Fluids*, 98:177–185, 2014.
- [27] A. Jaust, J. Schütz, and D. Seal. Multiderivative time-integrators for the hybridized discontinuous Galerkin method. In *Proceedings to YIC GACM 2015*, 2015.
- [28] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted {ENO} schemes. *Journal of Computational Physics*, 126(1):202–228, 1996.
- [29] Yan Jiang, Chi-Wang Shu, and Mengping Zhang. An alternative formulation of finite difference weighted ENO schemes with Lax-Wendroff time discretization for conservation laws. *SIAM J. Sci. Comput.*, 35(2):A1137–A1160, 2013.
- [30] Peter Lax and Burton Wendroff. Systems of conservation laws. *Communications on Pure and Applied Mathematics*, 13(2):217–237, 1960.
- [31] N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. *Journal of Computational Physics*, 228:3232–3254, 2009.
- [32] N. C. Nguyen, J. Peraire, and B. Cockburn. High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. *Journal of Computational Physics*, 230:3695–3718, 2011.
- [33] N. C. Nguyen, J. Peraire, and Bernardo Cockburn. An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations. *Journal of Computational Physics*, 228:8841–8855, 2009.
- [34] Truong Nguyen-Ba, Huong Nguyen-Thu, Thierry Giordano, and Remi Vaillancourt. One-step strong-stability-preserving Hermite-Birkhoff-Taylor methods. *Sci. J. of Riga Technical University*, 45:95–104, 2010.
- [35] J. Qiu. Development and comparison of numerical fluxes for LWDG methods. *Numerical Mathematics: Theory, Methods and Applications*, 1(4):435–459, 2008.
- [36] J. Qiu, M. Dumbser, and C.-W. Shu. The discontinuous Galerkin method with Lax–Wendroff type time discretizations. *Computer Methods in Applied Mechanics and Engineering*, 194(42-44):4528–4543, 2005.
- [37] Jianxian Qiu and Chi-Wang Shu. Finite difference WENO schemes with Lax-Wendroff-type time discretizations. *SIAM J. Sci. Comput.*, 24(6):2185–2198, 2003.
- [38] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. Technical report, Los Alamos Scientific Laboratory, 1973.
- [39] J. Schütz and G. May. A hybrid mixed method for the compressible Navier-Stokes equations. *Journal of Computational Physics*, 240:58–75, 2013.
- [40] David C. Seal, Yaman Güçlü, and Andrew J. Christlieb. High-order multiderivative time integrators for hyperbolic conservation laws. *J. Sci. Comput.*, 60(1):101–140, 2014.

- [41] David C. Seal, Qi Tang, Zhengfu Xu, and Andrew J. Christlieb. An explicit high-order single-stage single-step positivity-preserving finite difference weno method for the compressible euler equations. *arXiv preprint arXiv:1411.0328*, 2015.
- [42] S. Tan and C.-W. Shu. Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws. *Journal of Computational Physics*, 229(21):8144 – 8166, 2010.
- [43] S. Tan, C.-W. Shu, and J. Ning. Efficient implementation of high order inverse Lax-Wendroff boundary treatment for conservation laws. *Journal of Computational Physics*, 231:2510–2527, 2012.
- [44] A. Tsai, R. Chan, and S. Wang. Two-derivative Runge-Kutta methods for PDEs using a novel discretization approach. *Numerical Algorithms*, 65:687–703, 2014.
- [45] Y. Xu and C.-W. Shu. Local discontinuous Galerkin methods for high-order time-dependent partial differential equations. *Communications in Computational Physics*, 7(1):1–46, 2010.