



# Comparison of Semi-Lagrangian Discontinuous Galerkin Schemes for Linear and Nonlinear Transport Simulations

Xiaofeng Cai<sup>1</sup> · Wei Guo<sup>2</sup> · Jing-Mei Qiu<sup>1</sup>

Received: 31 March 2020 / Revised: 23 June 2020 / Accepted: 8 July 2020 / Published online: 14 September 2020  
© Shanghai University 2020

## Abstract

Transport problems arise across diverse fields of science and engineering. Semi-Lagrangian (SL) discontinuous Galerkin (DG) methods are a class of high-order deterministic transport solvers that enjoy advantages of both the SL approach and the DG spatial discretization. In this paper, we review existing SLDG methods to date and compare numerically their performance. In particular, we make a comparison between the splitting and non-splitting SLDG methods for multi-dimensional transport simulations. Through extensive numerical results, we offer a practical guide for choosing optimal SLDG solvers for linear and nonlinear transport simulations.

**Keywords** Semi-Lagrangian (SL) · Discontinuous Galerkin (DG) · Transport simulations · Splitting · Non-splitting · Comparison

**Mathematics Subject Classification** 65M60 · 65M25

## 1 Introduction

Semi-Lagrangian (SL) discontinuous Galerkin (DG) methods are a class of high-order transport solvers that enjoy the computational advantages of both the SL approach and the DG spatial discretization. In this paper, we conduct a systematic comparison for several existing SLDG methods in the literature by considering aspects including the accuracy,

---

W. Guo: Research is supported by NSF grant NSF-DMS-1830838.

J.-M. Qiu: Research is supported by NSF grant NSF-DMS-1522777 and NSF-DMS-1818924, Air Force Office of Scientific Computing FA9550-18-1-0257.

---

✉ Wei Guo  
weimath.guo@ttu.edu

Xiaofeng Cai  
xfcrai@udel.edu

Jing-Mei Qiu  
jingqiu@udel.edu

<sup>1</sup> Department of Mathematical Sciences, University of Delaware, Newark, DE 19716, USA

<sup>2</sup> Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX 70409, USA

CPU efficiency, conservation properties, implementation difficulties, with the aim to provide a brief survey of the recent development along this line of research for simulating linear and nonlinear transport problems.

The DG methods belong to the family of finite element methods, which employ piecewise polynomials as approximation and test function spaces. Such methods have undergone rapid development for simulating partial differential equations (PDEs) over the last few decades. For the time-dependent transport simulations, the DG methods are often coupled with the method-of-lines Eulerian framework, using appropriate time integrators for time evolution, e.g., the well-known TVD Runge-Kutta (RK) methods. It is well known that the DG method, when coupled with an explicit time integrator, suffers a very stringent CFL time step restriction for stability, which may be much smaller than that needed to resolve interesting timescales in physics. Implicit methods can be used to avoid the CFL time step restriction, yet the additional computational cost is required for solving the resulting linear or nonlinear system. The SL approach allows for large time step evolution by incorporating the characteristic tracing mechanism without implicit treatments. Meanwhile, the high-order accuracy can be conveniently attained because of its meshed-based nature. Such a distinguished property makes the SL approach very competitive in transport simulations, and it has been more and more popular across diverse fields of science and engineering, such as fluid dynamics [53], numerical weather prediction [26, 36], plasma physics [12, 14, 48], among many others. To alleviate the CFL time step restriction associated with the explicit Eulerian DG methods, the SLDG methods were developed in various settings [26, 44–46]. This kind of methods enjoys extraordinary computational advantages for transport simulations, such as high-order accuracy, provable unconditional stability and convergence, local mass conservation, small numerical dissipation, superior ability to resolve complex solution structures. Note that the SL framework is flexible and hence able to accommodate various existing spatial discretization methods, see the review paper [39].

In this paper, we are mainly concerned with the SLDG methods. For one-dimensional (1D) transport problems, there exist two types of SLDG methods: the characteristic Galerkin weak formulation and the flux difference formulation. When generalizing to the multi-dimensional case, similar to other SL methods, the available SLDG methods can be classified into two categories depending on whether the operator splitting strategy is used. The splitting based SL methods are popular in practice since one can directly make use of a preferred 1D SL formulation. A celebrated example is the Strang splitting SL method for the Vlasov-Poisson (VP) system proposed by Cheng and Knorr [12]. In particular, the nonlinearly coupled high-dimensional transport equation is decomposed into a set of lower dimensional sub-equations that are linear and hence much easier to evolve numerically in the SL setting. Following a similar idea, many splitting SL schemes based on various spatial discretization are designed, such as finite volume-based [22] and finite difference-based [10, 40, 42, 43] methods. More recently, several splitting SLDG methods are proposed in [44, 46]. On the other hand, the splitting methods suffer from the inherent splitting error. For instance, if the Strang splitting is used, then the second-order splitting error in time is incurred. As mentioned in [13], such a relatively low-order error may become significant for long-term transport simulations and hence greatly compromise the performance of the SL methods. In addition, for some non-linear problems, it can be very difficult to track characteristics accurately for the split sub-equations, posing challenges for the development of high-order splitting SL methods. This issue will be discussed in detail in Sect. 2.2. To completely avoid the splitting error, several non-splitting SL methods are developed. Here, we mention the conservative SL multi-tracer transport methods (CSLAM) [29, 33] and the SL spectral element transport methods [5, 19]. The first non-splitting SLDG scheme is proposed in [45]. Such a method is based on a flux

difference form but subject to a CFL time step restriction, degrading its computational efficiency to some extent. Recently, a line of research has been carried out for the development of non-splitting SLDG methods [7–9, 34]. The proposed methods are unconditionally stable, leading to computational efficiency. Meanwhile, the implementation is much more involved than the splitting counterpart, especially for problems in higher dimensions.

In this paper, we review the recent development of the state-of-the-art SLDG methods and systematically compare these methods in various settings. Our goal here is to provide a practical reference guide based on extensive numerical results on the study of the accuracy and error versus CPU time. Based on our investigation, we observe that the splitting SLDG method may be preferred due to its simplicity of implementation, especially when small CFLs are needed for accuracy. On the other hand, when large CFLs are used and the splitting error becomes pronounced, the non-splitting SLDG methods are preferred for long-time simulations. For example, for the nonlinear incompressible flow simulation, the non-splitting method is more efficient and effective, even though its implementation demands a large amount of human effort. Our simulation results demonstrate that the non-splitting method performs better. Note-worthy, both splitting and non-splitting SLDG formulations can enjoy other desired properties beyond mass conservation, including positivity preservation, high resolution for strong discontinuities, among others, if an appropriate limiter is further coupled. Other issues such as the parallel implementation and the superconvergence of SLDG schemes are discussed in [18] and [56], respectively.

The rest of the paper is organized as follows. In Sect. 2, we review the existing SLDG methods for solving linear and nonlinear problems. In Sect. 3, we compare the performance of SLDG methods via extensive numerical experiments with an emphasis on efficiency comparison between the splitting and non-splitting formulations. The concluding remark is provided in Sect. 4.

## 2 A Collection of SLDG Methods

In this section, we review several SLDG formulations to date. We start with 1D formulations for transport. When extending to multi-dimensions, we discuss both splitting and non-splitting strategies. We will highlight the key differences in the scheme formulations together with implementations, and compare their performance in the numerical section.

### 2.1 1D SLDG Methods

Consider the following 1D linear transport equation:

$$u_t + (a(x, t)u)_x = 0, \quad x \in \Omega \quad (2.1)$$

with proper initial and boundary conditions. For simplicity, we consider periodic boundary conditions in this paper. Here,  $a(x, t)$  is a velocity field, that could be space and time dependent. Note that (2.1) admits the characteristic equation

$$\frac{dx(t)}{dt} = a(x(t), t). \quad (2.2)$$

The domain is partitioned by non-overlapping intervals, i.e.,  $\Omega = \bigcup_j I_j$ , where  $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ . We denote  $\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$  as the length of an element and denote

$h = \max_j \Delta x_j$  as the mesh size. We further denote  $t^n$  as the discrete  $n$ th time level with  $\Delta t^n = t^{n+1} - t^n$  being the  $n$ th time stepping size. We define the finite dimensional DG approximation space,  $V_h^k = \{v_h : v_h|_{I_j} \in P^k(I_j)\}$ , where  $P^k(I_j)$  denotes the set of polynomials of degree at most  $k$  over  $I_j$ . Next we review several SLDG formulations proposed in the past two decades [7, 26, 45, 46].

### 2.1.1 SLDG Method Based on Characteristic Galerkin Weak Formulation

Based on the characteristic Galerkin weak formulation [26], we let the test function  $\psi(x, t)$  solve the adjoint problem satisfying the final value problem  $\psi(t = t^{n+1}) = \Psi(x)$  with  $\forall \Psi \in P^k(I_j)$ ,

$$\psi_t + a(x, t)\psi_x = 0, \quad t \in [t^n, t^{n+1}]. \quad (2.3)$$

Equation (2.3) is in the convective form; hence, the solution stays constant along a characteristic trajectory determined by (2.2). Such a strategy has been used in the ELLAM settings (see, e.g., [11, 20, 30, 50]) with continuous finite elements. It can be shown that [26]

$$\frac{d}{dt} \int_{\tilde{I}_j(t)} u(x, t)\psi(x, t)dx = 0, \quad (2.4)$$

where  $\tilde{I}_j(t)$  is a dynamic interval bounded by characteristics emanating from cell boundaries of  $I_j$  at  $t = t^{n+1}$ . Equation (2.4) leads to the following SLDG weak formulation. Given  $u^n \in V_h^k$ , we seek  $u^{n+1} \in V_h^k$ , such that  $\forall \Psi \in P^k(I_j)$ ,  $j = 1, \dots, m$ ,

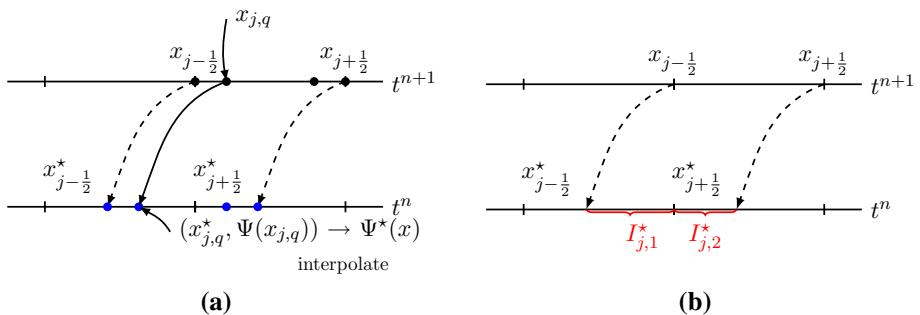
$$\int_{I_j} u^{n+1} \Psi dx = \int_{I_j^\star} u(x, t^n) \psi(x, t^n) dx, \quad (2.5)$$

where  $I_j^\star = [x_{j-\frac{1}{2}}^\star, x_{j+\frac{1}{2}}^\star]$  with  $x_{j\pm\frac{1}{2}}^\star$  being the feet of trajectory emanating from  $(x_{j\pm\frac{1}{2}}^\star, t^{n+1})$  at the time level  $t^n$ . To update the numerical solution  $u^{n+1}$ , we need to evaluate the integral on the right-hand side (RHS) of (2.5). Here, we briefly review the procedure and refer the reader to [7] for details. First, we choose  $k+1$  interpolation points  $x_{j,q}$ ,  $q = 0, \dots, k$  over  $I_j$  and locate their feet  $x_{j,q}^\star$  by backward characteristics tracing from  $t^{n+1}$  to  $t^n$ . Since the test function  $\psi$  stays constant along the characteristics, i.e.,  $\psi(x_{j,q}^\star, t^n) = \Psi(x_{j,q})$ , we can uniquely determine a polynomial  $\Psi^\star(x)$  that interpolates  $\psi(x, t^n)$  with  $(x_{j,q}^\star, \Psi(x_{j,q}))$  for  $q = 0, \dots, k$ , see Fig. 1a. Further observing that solution  $u^n$  is a piecewise polynomial, the RHS of (2.5) has to be approximated subinterval-by-subinterval for accuracy and stability, i.e.,

$$\int_{I_j^\star} u(x, t^n) \psi(x, t^n) dx \approx \sum_l \int_{I_{j,l}^\star} u^n(x) \Psi^\star(x) dx, \quad (2.6)$$

where  $\bigcup_l I_{j,l}^\star = I_j^\star$  denotes the collection of the intersections between  $I_j^\star$  and the grid elements ( $l$  is the index for subinterval), see Fig. 1b. Note that the integrands in (2.6) are polynomials of degree  $2k$ ; thus, the integration can be evaluated exactly.

**Remark 2.1** There is another slightly different procedure in evaluating the RHS integrals in (2.5), that is, to locate quadrature points on subintervals, track their values along



**Fig. 1** Schematic illustration for the 1D SLDG scheme in [7]

characteristic curves using (2.3), and perform quadrature integrations subinterval-by-subinterval. For example, see the procedures proposed in [26, 34]. In the case of having the constant advection coefficient, the procedure in [26, 34] and the one reviewed above [7] are equivalent; they are also equivalent to the shift-projection strategy in the SLDG scheme as proposed in [46]. In the case of the variable coefficient, the procedure reviewed above as proposed in [7] can be better generalized to the two-dimensional (2D) case in Sect. 2.3.

**Remark 2.2** More recently, an SLDG method based on the forward characteristics tracing mechanism has been developed [6], which can be considered as an extension of the SL continuous finite element method, see, e.g., [31]. In such a formulation, the DG solution is first evolved forward following the characteristics from  $t^n$  to  $t^{n+1}$ , yielding an intermediate solution which in general does not belong to space  $V_h^k$ . Then based on a standard DG weak formulation, the numerical solution at  $t^{n+1}$  is obtained by projecting the intermediate solution back to space  $V_h^k$ . The forward SLDG formulation is inherently nonconservative for mass, as opposed to the backward formulation (2.5), while a mass fixer can be coupled to ensure global mass conservation. On the other hand, an extra cost is incurred, and the original high-order accuracy of the SLDG method may be destroyed, see [6].

### 2.1.2 SLDG Methods Based on a Flux Difference Form

There exist two SLDG methods based on a flux difference form in the literature [44, 45]. Such methods are motivated by the standard DG method for solving hyperbolic conservation laws. Multiplying (2.1) by a time-independent test function  $\Psi$ , integrating the resulting equation over  $I_j$  and performing integration by parts yields

$$\frac{d}{dt} \int_{I_j} u(x, t) \Psi dx - \int_{I_j} a(x, t) u(x, t) \Psi_x dx + a(x, t) u(x, t) \Psi \Big|_{x_{j+\frac{1}{2}}} - a(x, t) u(x, t) \Psi \Big|_{x_{j-\frac{1}{2}}} = 0. \quad (2.7)$$

Instead of coupling a time integrator in the method-of-lines fashion, the SLDG methods make use of the characteristic tracing mechanism. In particular, we further integrate (2.7) over time interval  $[t^n, t^{n+1}]$  and define the SLDG method accordingly. Given data  $u^n \in V_h^k$ , we seek  $u^{n+1} \in V_h^k$  such that  $\forall \Psi \in V_h^k$ ,

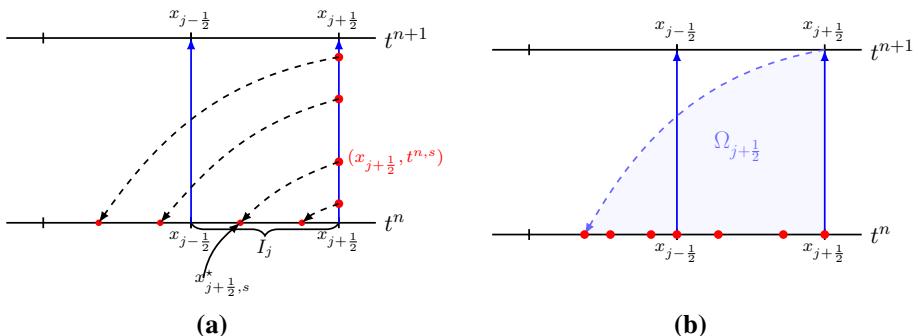
$$\begin{aligned} \int_{I_j} u^{n+1} \Psi \, dx &= \int_{I_j} u^n \Psi \, dx + \int_{t^n}^{t^{n+1}} \left( \int_{I_j} a(x, t) S_t(u^n) \Psi_x \, dx \right) dt \\ &\quad + \int_{t^n}^{t^{n+1}} \left( a(x, t) S_t(u^n) \Psi \Big|_{x_{j+\frac{1}{2}}^-} \right) dt - \int_{t^n}^{t^{n+1}} \left( a(x, t) S_t(u^n) \Psi \Big|_{x_{j-\frac{1}{2}}^+} \right) dt, \end{aligned} \quad (2.8)$$

$n = 0, 1, \dots$ , where  $S_t(u^n)$  denotes the function evolved from the given initial data  $u^n$  by following the characteristics, see [44, 45] for more details. The integrals in space and time appearing in (2.8) are computed by numerical quadrature rules for the SLDG method proposed in [45] on the fully discrete level. For example, to compute the third integral on the RHS of (2.8), we locate  $p + 1$  quadrature nodes over  $[t^n, t^{n+1}]$ , denoted as  $t^{n,s}$ ,  $s = 0, \dots, p$ . Then,

$$\int_{t^n}^{t^{n+1}} \left( a(x, t) S_t(u^n) \Psi \Big|_{x_{j+\frac{1}{2}}^-} \right) dt \approx \sum_s a\left(x_{j+\frac{1}{2}}, t^{n,s}\right) S_t\left(u^n\left(x_{j+\frac{1}{2},s}^\star\right)\right) \Psi \Big|_{x_{j+\frac{1}{2}}^-} w_{n,s},$$

where  $x_{j+\frac{1}{2},s}^\star$  denotes the feet of the characteristic trajectories emanating from the points  $(x_{j+\frac{1}{2}}, t^{n,s})$ ,  $s = 0, \dots, p$ , see the left panel in Fig. 2, and  $w_{n,s}$  denotes the associated quadrature weights. As pointed out in [44],  $S_t(u^n)$  may be a discontinuous function on  $[t^n, t^{n+1}]$  for large time step evolution, e.g.,  $\text{CFL} \geq 1$ , due to the discontinuous nature of  $u^n$ . In such a case, a numerical quadrature rule in time may suffer from some stability issue, making the SLDG formulation only conditionally stable. To remedy the drawback, an alternative SLDG method is developed in [44], which makes use of the divergence theorem to convert the integrals in time into the integrals in space. We still consider the third integral on the RHS of (2.8) as an example. A direct application of the divergence theorem to the integral form of the DG formulation (2.7) over domain  $\Omega_{j+\frac{1}{2}}$  yields

$$\int_{t^n}^{t^{n+1}} \left( a(x, t) S_t(u^n) \Psi \Big|_{x_{j+\frac{1}{2}}^-} \right) dt = \int_{x_{j+\frac{1}{2}}^\star}^{x_{j+\frac{1}{2}}} u^n(x) \, dx \cdot \Psi \Big|_{x_{j+\frac{1}{2}}^-}.$$



**Fig. 2** Schematic illustration for the 1D SLDG scheme in [45] (a) and the scheme in [43] (b)

Hence, numerical quadrature rules are applied to the integrals in space only and in the subinterval-by-subinterval manner, see the right panel in Fig. 2, leading to an unconditionally stable SLDG method.

**Remark 2.3** Since the SLDG method in [45] is only conditionally stable, the efficiency is limited to some extent. On the other hand, such a formulation can be directly generalized to the multi-dimensional case without operator splitting. For the SLDG method with the divergence theorem [44], its direct generalization to a multi-dimensional problem is much more difficult and technically involved, thus dimensional splitting is often used as discussed in Sect. 2.2.

**Remark 2.4** It can be shown that the SLDG formulation (2.5) is equivalent to (2.8) if all the integrals are computed exactly. In general, we have to resort to numerical quadrature rules or polynomial interpolation, and hence these SLDG formulations have slightly different performance in simulations.

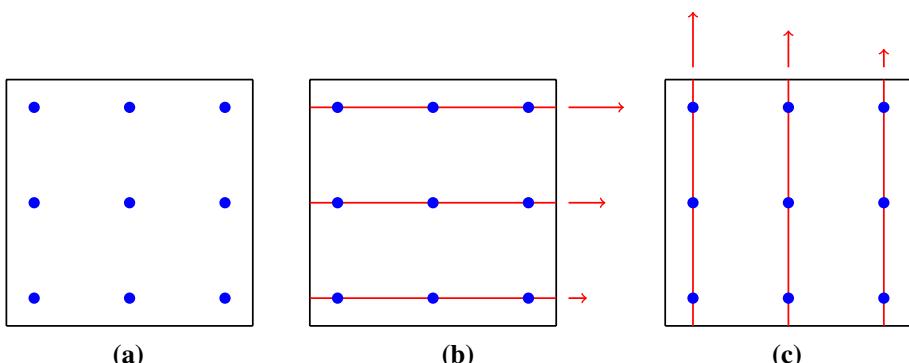
## 2.2 2D SLDG Methods with Operator Splitting

Next we discuss the extension of the 1D SLDG formulations to the 2D case via dimensional splitting. Consider the following 2D transport equation:

$$u_t + (a(x, y, t)u)_x + (b(x, y, t)u)_y = 0, \quad (x, y) \in \Omega \quad (2.9)$$

with a proper initial condition  $u(x, y, 0) = u_0(x, y)$ . Here,  $(a(x, y, t), b(x, y, t))$  is a prescribed velocity field depending on time and space. Boundary conditions are periodic for simplicity. The rectangular domain  $\Omega$  is partitioned in terms of a Cartesian mesh with each computational cell denoted as  $A_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$ . In the dimensional splitting setting, we use the piecewise  $Q^k$  tensor-product polynomial spaces. Hence, there are  $(k+1)^2$  degrees of freedom per computational cell.

Note that all the 1D SLDG formulations introduced in Sect. 2.1 can be extended to the 2D case via operator splitting. Below, we present a general framework of splitting SLDG algorithms. We first locate  $(k+1)^2$  tensor-product Gaussian nodes on cell  $A_{ij}$ :  $(x_{i,p}, y_{j,q})$ ,



**Fig. 3** Schematic illustration of the 2D SLDG scheme via Strang splitting,  $k = 2$

$p, q = 0, \dots, k$ . For example, see Fig. 3 (left) for the case of  $k = 2$ . Then (2.9) is split into two 1D advection problems based on the quadrature nodes in both  $x$ - and  $y$ -directions:

$$u_t + (a(x, y, t)u)_x = 0, \quad (2.10)$$

$$u_t + (b(x, y, t)u)_y = 0. \quad (2.11)$$

Based on a 1D SLDG formulation, the split equations (2.10)–(2.11) are evolved via Strang splitting over a time step  $\Delta t$  as follows. (a) Evolve the 1D equation (2.10) at each quadrature node  $y_{j,q}$  for a half time-step  $\Delta t/2$ , see Fig. 3 (middle); (b) evolve the 1D equation (2.11) at each quadrature node  $x_{i,p}$  for a full time-step  $\Delta t$ , see Fig. 3 (right); (c) evolve the 1D equation (2.10) at each quadrature node  $y_{j,q}$  for another half time-step  $\Delta t/2$ , see Fig. 3 (middle).

**Remark 2.5** The splitting 2D SLDG formulation inherits many desired properties from the base 1D formulation, such as the high-order accuracy in space, the unconditional stability, and the mass conservation. Meanwhile, a splitting error in time is introduced. For the Strang splitting, the error is of second order. Higher order splitting methods can be constructed in the spirit of composition methods [28, 57], while the number of intermediate stages; hence, the computational cost, increases exponentially with the order of the splitting method. For example, a fourth-order splitting SLDG method is developed in [46] for solving the VP system.

### 2.3 2D SLDG Method Without Operator Splitting

Recently, a class of high-order non-splitting SLDG methods has been under great development [7–9] and in [34] for unstructured meshes. They are unconditionally stable and mass conservative. Below we briefly describe the 2D SLDG algorithm proposed in [7], which is a direct generalization of the 1D algorithm in Sect. 2.1. We assume that the domain  $\Omega$  is rectangular and is partitioned by a set of non-overlapping tensor-product rectangular elements  $A_j$ ,  $j = 1, \dots, J$ . We remark that, unlike the splitting formulation, such a non-splitting SLDG method is based on a truly multi-dimensional formulation and hence allows for an unstructured mesh [34]. We then define the finite dimensional DG approximation space,  $\mathbf{V}_h^k = \{v_h : v_h|_{A_j} \in P^k(A_j)\}$ , where  $P^k(A_j)$  denotes the set of polynomials of degree at most  $k$  over  $A_j$ .

Similar to the 1D case, we need the weak formulation of characteristic Galerkin formulation [7, 26]. Specifically, let the test function  $\psi(x, y, t)$  satisfy the adjoint problem with  $\Psi \in P^k(A_j)$ ,

$$\begin{cases} \psi_t + a(x, y, t)\psi_x + b(x, y, t)\psi_y = 0, \\ \psi(t = t_{n+1}) = \Psi. \end{cases} \quad (2.12)$$

Then we have the identity

$$\frac{d}{dt} \int_{\tilde{A}_j(t)} u(x, y, t)\psi(x, y, t) dx dy = 0, \quad (2.13)$$

where  $\tilde{A}_j(t)$  is the dynamic cell, moving from the Eulerian cell  $A_j$  at  $t^{n+1}$ , i.e.,  $A_j = \tilde{A}_j(t^{n+1})$ , backward in time by following the characteristics trajectories. We denote  $\tilde{A}_j(t^n)$  as  $A_j^\star$ , i.e.,

the upstream cell bounded by the red curves in Fig. 4. The SLDG method is defined as follows. Given  $u^n \in \mathbf{V}_h^k$ , we seek  $u^{n+1} \in \mathbf{V}_h^k$ , such that  $\forall \Psi \in P^k(A_j)$ ,  $j = 1, \dots, J$ ,

$$\int_{A_j} u^{n+1} \Psi dx dy = \int_{A_j^*} u^n \psi(x, y, t^n) dx dy. \quad (2.14)$$

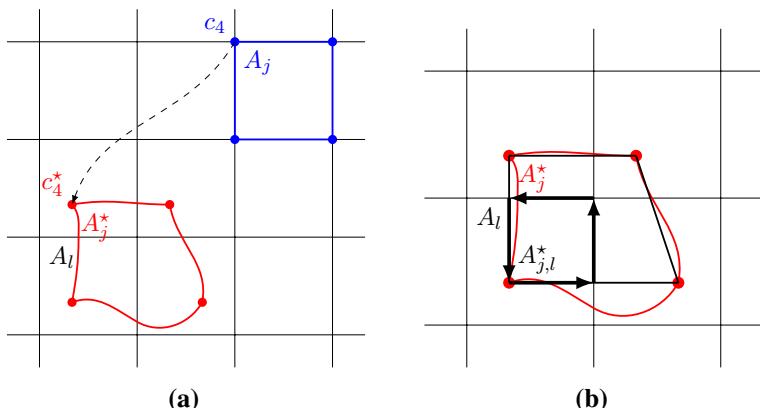
As in the 1D case, the main difficulty lies in the evaluation of the RHS in (2.14). Below, we briefly discuss the procedure.

Denote  $c_q$ ,  $q = 1, \dots, 4$  as the four vertices of  $A_j$  with the coordinates  $(x_{j,q}, y_{j,q})$ . We trace characteristics backward in time to  $t^n$  for the four vertices and obtain  $c_q^*$  with the new coordinate  $(x_{j,q}^*, y_{j,q}^*)$ ,  $q = 1, \dots, 4$ . For example, see  $c_4$  and  $c_4^*$  in Fig. 4. The upstream cell  $A_j^*$  can be approximated by a quadrilateral determined by the four vertices  $c_q^*$ . Note that  $u^n$  is a piecewise polynomial based on the partition. Then the integral over  $A_j^*$  in (2.14) has to be evaluated sub-region-by-subregion. To this end, we denote  $A_{j,l}^*$  as a non-empty overlapping region between the upstream cell  $A_j^*$  and the background grid cell  $A_l$ , i.e.,  $A_{j,l}^* = A_j^* \cap A_l$ ,  $A_{j,l}^* \neq \emptyset$ , and define the index set  $\varepsilon_j^* := \{l | A_{j,l}^* \neq \emptyset\}$ , see Fig. 4b. The detailed procedure of detecting  $A_{j,l}^*$  can be found in [7]. The integral over the upstream cell  $A_j^*$  is broken up into the following integrals:

$$\int_{A_j} u^{n+1} \Psi dx dy = \sum_{l \in \varepsilon_j^*} \int_{A_{j,l}^*} u^n \psi(x, y, t^n) dx dy. \quad (2.15)$$

Furthermore, as in the 1D case, even though  $\psi(x, y, t^n)$  is not a polynomial in general, it can be well approximated by a polynomial given the velocity field  $(a, b)$  is smooth. In particular, based on the fact that  $\psi$  stays constant along characteristics, we construct a least-squares approximation  $\Psi^*(x, y)$  to  $\psi(x, y, t^n)$ , which is a  $P^1$  polynomial for  $k = 1$ . To further simplify the implementation of evaluating the RHS of (2.15), we make use of Green's theorem. We first introduce two auxiliary polynomial functions  $P(x, y)$  and  $Q(x, y)$  such that

$$-\frac{\partial P}{\partial y} + \frac{\partial Q}{\partial x} = u(x, y, t^n) \Psi^*(x, y).$$



**Fig. 4** Schematic illustration of the SLDG formulation in two dimension: quadrilateral approximation to an upstream cell

Then the area integral  $\int_{A_{jl}^*} u^n \Psi^*(x, y) dx dy$  can be converted into line integrals via Green's theorem, i.e.,

$$\int_{A_{jl}^*} u(x, y, t^n) \Psi^*(x, y) dx dy = \oint_{\partial A_{jl}^*} P dx + Q dy, \quad (2.16)$$

see Fig. 4b. Note that the choices of  $P$  and  $Q$  are not unique, but the value of the line integrals is independent of the choices. For the implementation details, see [7].

**Remark 2.6** The quadrilateral approximation to upstream cells is second order accurate. To achieve third order accuracy, one can use the quadratic-curved quadrilateral approximation, see [7] for more details.

## 2.4 SLDG Schemes for Nonlinear Models

In this subsection, we discuss the application of the SLDG methods in nonlinear modelings, e.g., the nonlinear VP system, the guiding center Vlasov equation, and the incompressible Euler equation.

- i) Vlasov-Poisson system. We consider the 1D1V VP system,

$$f_t + v f_x + E(x, t) f_v = 0, \quad (2.17)$$

$$E(x, t) = -\phi_x, \quad -\phi_{xx}(x, t) = \rho(x, t). \quad (2.18)$$

Here  $x$  and  $v$  are the coordinates in the phase space  $(x, v) \in \Omega_x \times \mathbb{R}$ , and  $f(x, v, t)$  is the probability distribution function describing the probability of finding a particle with velocity  $v$  at position  $x$  and at time  $t$ . Note that in the Vlasov equation (2.17)  $f$  is accelerated by the electric field  $E$  which is determined by Poisson's equation (2.18).  $\phi$  is the self-consistent electrostatic potential, and  $\rho(x, t) = \int_{\mathbb{R}} f(x, v, t) dv - 1$  denotes the charge density. Here, we assume that infinitely massive ions are uniformly distributed in the background. We recall several conserved quantities in the VP system below, which should remain constant in time.

- $L^p$  norm,  $1 \leq p \leq \infty$ :

$$\|f\|_p = \left( \int_v \int_x |f(x, v, t)|^p dx dv \right)^{\frac{1}{p}}. \quad (2.19)$$

- Energy:

$$\text{Energy} = \int_v \int_x f(x, v, t) v^2 dx dv + \int_x E^2(x, t) dx. \quad (2.20)$$

- Entropy:

$$\text{Entropy} = \int_v \int_x f(x, v, t) \log(f(x, v, t)) dx dv. \quad (2.21)$$

- ii) Incompressible Euler equations and the guiding center Vlasov model. The 2D time-dependent incompressible Euler equations in the vorticity-stream function formulation read

$$\begin{cases} \omega_t + \nabla \cdot (\mathbf{u}\omega) = 0, \\ \Delta\Phi = \omega, \mathbf{u} = (-\Phi_y, \Phi_x), \end{cases} \quad (2.22)$$

where  $\mathbf{u}$  is the velocity field,  $\omega$  is the vorticity of the fluid, and  $\Phi$  is the stream-function determined by Poisson's equation. The other closely related model concerned is the guiding center approximation of the 2D Vlasov model, which describes a highly magnetized plasma in the transverse plane of a Tokamak [14, 24, 47, 55]:

$$\rho_t + \nabla \cdot (\mathbf{E}^\perp \rho) = 0, \quad (2.23)$$

$$-\Delta\Phi = \rho, \mathbf{E}^\perp = (-\Phi_y, \Phi_x), \quad (2.24)$$

where the unknown variable  $\rho$  denotes the charge density of the plasma, and  $\mathbf{E}$ , determined by  $\mathbf{E} = -\nabla\Phi$ , is the electric field. Despite their different application backgrounds, the above two models indeed have an equivalent mathematical formulation up to a sign difference in Poisson's equation. For both models, the following physical quantities remain constant over time.

- Energy:

$$\|\mathbf{u}\|_{L^2}^2 = \int_\Omega \mathbf{u} \cdot \mathbf{u} dx dy, \quad (\text{Euler}), \quad \|\mathbf{E}\|_{L^2}^2 = \int_\Omega \mathbf{E} \cdot \mathbf{E} dx dy, \quad (\text{Vlasov}).$$

- Enstrophy:

$$\|\omega\|_{L^2}^2 = \int_\Omega \omega^2 dx dy, \quad (\text{Euler}), \quad \|\rho\|_{L^2}^2 = \int_\Omega \rho^2 dx dy, \quad (\text{Vlasov}).$$

The splitting SLDG methods for linear transport can be directly applied to both nonlinear models. Below, we summarize the procedures. The notation is consistent with that used in Sect. 2.2.

- For the VP system (2.17), we follow the classic SL approach with Strang splitting [12, 17, 43, 44, 46]. Over one time step  $\Delta t$ , we perform the following steps.
  - ① Evolve the split 1D equation  $f_t + vf_x = 0$  at each quadrature node  $v_{j,q}$  for a half time-step  $\Delta t/2$ .
  - ② Solve Poisson's equation  $-\phi_{xx}(x, t^{n+\frac{1}{2}}) = \rho(x, t^{n+\frac{1}{2}})$ , and compute  $E^{n+\frac{1}{2}} = -\phi_x(x, t^{n+\frac{1}{2}})$ .
  - ③ Evolve the split 1D equation  $f_t + E^{n+\frac{1}{2}}f_v = 0$  at each quadrature node  $x_{i,p}$  for a full time-step  $\Delta t$ .

- ④ Evolve the 1D equation  $f_t + vf_x = 0$  at each quadrature node  $v_{j,q}$  for a half time-step  $\Delta t/2$ .

The Poisson's equation above can be solved by any proper elliptic solvers, such as the LDG method [2, 46]. The Strang splitting strategy is subject to a second-order splitting error in time.

- (ii) If we directly apply the Strang splitting idea to the 2D incompressible Euler equation in the vorticity-stream function formulation as well as the guiding center Vlasov model, the procedures are outlined below.

- ① Obtain  $\Phi_y$  by solving  $\Delta\Phi = \omega^n$ , where  $\omega^n$  denotes the numerical solution of  $\omega(x, y, t^n)$ .
- ② Evolve the split 1D equation  $\omega_t - (\Phi_y \omega)_x = 0$  at each quadrature node  $y_{j,q}$  for a half time-step  $\Delta t/2$  and then obtain the solution  $\omega^*$ .
- ③ Obtain  $\Phi_x$  by solving  $\Delta\Phi = \omega^*$ .
- ④ Evolve the 1D equation  $\omega_t + (\Phi_x \omega)_y = 0$  at each quadrature node  $x_{i,p}$  for a full time-step  $\Delta t$  and obtain the solution  $\omega^{**}$ .
- ⑤ Obtain  $\Phi_y$  by solving  $\Delta\Phi = \omega^{**}$ .
- ⑥ Evolve the 1D equation  $\omega_t - (\Phi_y \omega)_x = 0$  at each quadrature node  $y_{j,q}$  for a half time-step  $\Delta t/2$ , and obtain the solution  $\omega^{n+1}$  on next time step  $t^{n+1}$ .

**Remark 2.7** Despite the Strang splitting procedure, the above algorithm is only of first-order accuracy in time. The reason is that the field  $\Phi$  is time dependent for the split 1D equation. When the field equation and transport equation are solved in a step-by-step manner, numerical tracing of characteristics feet is subject to a first-order error in time, which is difficult to enhance beyond first-order accuracy. Note that for the case of the VP system, the field equation  $\phi$  and hence the electric field  $E$  does not change in time, when the split 1D equation  $f_t + Ef_v = 0$  is evolved.

When applying the 2D SLDG methods for the nonlinear Vlasov and incompressible Euler models, the main difficulty is to track nonlinear characteristics with high-order temporal accuracy. We adopt the two-stage multi-derivative prediction-correction algorithms as proposed in [41] for the VP system and in [52] for the incompressible Euler equations and the guiding center model. We refer the reader to [8, 9] for details.

### 3 Numerical Tests

In this section, we perform extensive numerical tests to compare the SLDG methods reviewed in the previous section for solving linear and nonlinear transport problems. Note that the 1D SLDG methods based on the characteristic Galerkin formulation in Sect. 2.1.1 and based on the flux difference form with the unconditional stability in Sect. 2.1.2 perform virtually the same in terms of accuracy and CPU cost. Hence, we only apply the SLDG formulation in Sect. 2.1.1 with dimensional splitting for its comparison against the non-splitting SLDG method for 2D transport simulations. Examples include 2D linear passive-transport problems in Sect. 3.1, the nonlinear VP system in Sect. 3.2, and the incompressible

Euler equation and the guiding center Vlasov model in Sect. 3.3. The performance of the two SLDG methods is benchmarked in terms of error magnitude, order of convergence in space and time, CPU cost, as well as resolution of complex solution structures.

For CPU time comparison, all simulations are performed on a laptop, Intel(R) Core (TM) i7-6300HQ CPU @ 2.60 GHz with 2.59 GB ram. In our notation, we denote the splitting SLDG method using  $Q^k$  approximation spaces as  $Q^k$  SLDG-split, and denote the non-splitting SLDG method using the  $P^k$  approximation space as  $P^k$  SLDG. For the non-splitting SLDG method, when the quadratic-curves quadrilateral approximation is used to approximate upstream cells, such a method is denoted as  $P^k$  SLDG-QC. For the splitting method, we always employ the second-order Strang splitting as discussed in Sect. 2.4. The time step is set as

$$\Delta t = \frac{\text{CFL}}{\frac{a}{\Delta x} + \frac{b}{\Delta y}}, \quad (3.1)$$

where  $a$  and  $b$  are maximum transport speeds in  $x$ - and  $y$ -directions, respectively. The CFL number is specified for comparison with the RKDG formulation, whose CFL upper bound is  $1/(2k + 1)$  with  $k$  being the degree of the polynomial.

### 3.1 2D Linear Passive-Transport Problems

**Example 3.1** (Linear transport equation with constant coefficient) Consider

$$u_t + u_x + u_y = 0, \quad (x, y) \in [-\pi, \pi]^2 \quad (3.2)$$

with periodic boundary conditions and the initial condition  $u(x, y, 0) = \sin(x + y)$ . The exact solution is  $u(x, y, t) = \sin(x + y - 2t)$ . Note that for this example, since the convection operators in  $x$ - and  $y$ -directions commute, there is no splitting error. Table 1 summarizes the  $L^2$  and  $L^\infty$  errors, the associated orders of convergence, and the CPU cost for  $Q^k$  SLDG-split and  $P^k$  SLDG for  $k = 1, 2$ . We let the final time  $T = \pi$ , and consider CFL = 2.5 and CFL = 10.5. The expected  $(k + 1)$ -th order convergence is observed for all cases. It is also observed that the error magnitude by  $Q^k$  SLDG-split is smaller than that by  $P^k$  SLDG. This is because  $Q^k$  has more degrees of freedom than  $P^k$  and hence enjoys better approximation property, even though their approximation rates are the same. By comparing the CPU cost of  $Q^k$  SLDG-split and  $P^k$  SLDG with the same CFL, we observe that  $Q^k$  SLDG-split is more efficient than  $P^k$  SLDG, which is ascribed to the absence of the splitting error as well as the simplicity of the 1D implementation procedure for  $Q^k$  SLDG-split.

**Example 3.2** (Rigid body rotation) Consider

$$u_t - (yu)_x + (xu)_y = 0, \quad (x, y) \in [-2\pi, 2\pi]^2. \quad (3.3)$$

We first consider a circular symmetry initial condition  $u(x, y, 0) = \exp(-x^2 - y^2)$  and run the simulations to  $T = 20\pi$  (10 periods of rotation) with CFL = 2.5 and CFL = 10.5. Table 2 summarizes the  $L^2$  and  $L^\infty$  errors, the associated orders of the convergence and the CPU cost of both SLDG methods. It is observed that  $Q^k$  SLDG-split produces results with smaller error magnitude and at the same time requires less CPU time compared with  $P^k$  SLDG. Indeed, even with a large CFL = 10.5, the spatial error from  $Q^k$  SLDG-split

**Table 1**  $P^k$  SLDG and  $Q^k$  SLDG-split ( $k = 1, 2$ ) for (3.2) with  $u(x, y, 0) = \sin(x + y)$  at  $T = \pi$ 

Mesh	CFL = 2.5					CFL = 10.5				
	$L^2$ error	Order	$L^\infty$ error	Order	CPU	$L^2$ error	Order	$L^\infty$ error	Order	CPU
<b><math>P^1</math> SLDG</b>										
20 <sup>2</sup>	7.24E–03	–	4.07E–02	–	0.02	7.43E–03	–	4.33E–02	–	0.01
40 <sup>2</sup>	1.82E–03	1.99	1.04E–02	1.96	0.14	1.82E–03	2.03	1.05E–02	2.04	0.03
80 <sup>2</sup>	4.55E–04	2.00	2.63E–03	1.99	1.02	4.55E–04	2.00	2.64E–03	1.99	0.31
160 <sup>2</sup>	1.14E–04	2.00	6.60E–04	2.00	7.89	1.14E–04	2.00	6.61E–04	2.00	2.14
<b><math>Q^1</math> SLDG-split</b>										
20 <sup>2</sup>	1.91E–03	–	4.04E–03	–	0.02	2.21E–03	–	3.92E–03	–	0.01
40 <sup>2</sup>	4.71E–04	2.02	9.55E–04	2.08	0.13	6.76E–04	1.71	1.35E–03	1.53	0.03
80 <sup>2</sup>	1.17E–04	2.00	2.31E–04	2.05	0.84	1.17E–04	2.53	2.24E–04	2.59	0.25
160 <sup>2</sup>	2.93E–05	2.00	5.67E–05	2.03	6.31	2.93E–05	2.00	5.59E–05	2.01	1.56
<b><math>P^2</math> SLDG</b>										
20 <sup>2</sup>	3.54E–04	–	2.34E–03	–	0.05	3.64E–04	–	2.25E–03	–	0.02
40 <sup>2</sup>	4.42E–05	3.00	2.92E–04	3.01	0.30	4.41E–05	3.05	2.89E–04	2.96	0.09
80 <sup>2</sup>	5.53E–06	3.00	3.63E–05	3.00	2.25	5.52E–06	3.00	3.62E–05	3.00	0.62
160 <sup>2</sup>	6.91E–07	3.00	4.54E–06	3.00	17.59	6.91E–07	3.00	4.54E–06	2.99	4.58
<b><math>Q^2</math> SLDG-split</b>										
20 <sup>2</sup>	4.18E–05	–	1.08E–04	–	0.05	7.08E–05	–	1.71E–04	–	0.01
40 <sup>2</sup>	5.43E–06	2.94	1.38E–05	2.96	0.30	8.38E–06	3.08	2.28E–05	2.91	0.09
80 <sup>2</sup>	6.79E–07	3.00	1.73E–06	3.00	2.22	6.60E–07	3.67	1.71E–06	3.73	0.61
160 <sup>2</sup>	8.49E–08	3.00	2.16E–07	3.00	17.83	8.51E–08	2.96	2.17E–07	2.98	4.30

discretization still dominates the splitting error due to the symmetry of the solution. Then we take another initial condition  $u(x, y, 0) = \exp(-x^2 - 10y^2)$ , for which the circular symmetry no longer holds. Table 3 summarizes the  $L^2$  and  $L^\infty$  errors, the associated orders of the convergence and the CPU cost of both methods with CFL = 2.5 and CFL = 10.5 at  $T = 20\pi$ . We observe that, for CFL = 2.5,  $Q^k$  SLDG-split performs better than  $P^k$  SLDG in terms of CPU efficiency, due to the fact that the spatial error still dominates the splitting error. While for a large CFL = 10.5, the splitting error in time becomes significant and hence dominates the total numerical error for  $Q^2$  SLDG-split. In particular, the second-order convergence due to the Strang splitting is observed, and the error magnitude of  $Q^2$  SLDG-split is much larger than that of  $P^k$  SLDG with the same configuration. To better understand the splitting error in time, we test the temporal accuracy of both schemes by varying CFL with a fixed mesh of  $160 \times 160$  cells, see Table 4. It is observed that  $Q^2$  SLDG-split suffers a second-order splitting error, and the approximation quality deteriorates quickly as CFL increases. On the other hand,  $P^2$  SLDG with CFL as large as 25 can still provide an accurate approximation to the solution. The error magnitude stays nearly the same as CFL increases, indicating the dominating spatial error and relatively small temporal error even with large CFLs.

We finally compare the performance on shape preservation for various schemes by simulating the rigid body rotation (3.3) with the following setting: the computational domain is  $[-\pi, \pi]$ ; we consider the periodic boundary conditions and an initial condition plotted in

**Table 2**  $P^k$  SLDG and  $Q^k$  SLDG-split ( $k = 1, 2$ ) for (3.3) with  $u(x, y, 0) = \exp(-x^2 - y^2)$  at  $T = 20\pi$ 

Mesh	CFL = 2.5					CFL = 10.5				
	$L^2$ error	Order	$L^\infty$ error	Order	CPU	$L^2$ error	Order	$L^\infty$ error	Order	CPU
<b><math>P^1</math> SLDG</b>										
$20^2$	4.60E-02	—	5.25E-01	—	1.00	2.88E-02	—	3.62E-01	—	0.34
$40^2$	1.50E-02	1.62	1.95E-01	1.43	7.73	7.29E-03	1.98	1.20E-01	1.59	2.06
$80^2$	2.70E-03	2.47	3.76E-02	2.38	59.30	1.19E-03	2.62	2.71E-02	2.15	15.48
$160^2$	3.88E-04	2.80	6.85E-03	2.46	468.88	1.82E-04	2.71	6.05E-03	2.17	120.94
<b><math>Q^1</math> SLDG-split</b>										
$20^2$	2.88E-02	—	2.12E-01	—	0.88	2.01E-02	—	1.78E-01	—	0.17
$40^2$	8.43E-03	1.77	9.61E-02	1.14	7.08	5.72E-03	1.81	6.80E-02	1.39	1.58
$80^2$	1.27E-03	2.74	1.56E-02	2.62	55.94	8.49E-04	2.75	1.30E-02	2.39	11.58
$160^2$	1.77E-04	2.84	2.59E-03	2.60	452.00	1.21E-04	2.81	2.43E-03	2.42	91.19
<b><math>P^2</math> SLDG</b>										
$20^2$	5.86E-03	—	7.59E-02	—	2.17	2.13E-03	—	3.45E-02	—	0.63
$40^2$	3.07E-04	4.25	5.84E-03	3.70	16.64	1.53E-04	3.81	3.39E-03	3.35	4.45
$80^2$	1.84E-05	4.06	4.23E-04	3.79	129.36	1.50E-05	3.35	4.50E-04	2.91	33.55
$160^2$	1.92E-06	3.26	6.00E-05	2.82	1 032.59	1.72E-06	3.12	5.87E-05	2.94	262.78
<b><math>Q^2</math> SLDG-split</b>										
$20^2$	2.01E-03	—	2.95E-02	—	2.34	2.13E-03	—	1.78E-02	—	0.52
$40^2$	1.32E-04	3.93	3.50E-03	3.08	17.94	1.29E-04	4.04	1.71E-03	3.38	4.03
$80^2$	8.96E-06	3.88	2.83E-04	3.63	142.02	7.54E-06	4.10	2.22E-04	2.95	31.92
$160^2$	9.19E-07	3.29	1.90E-05	3.90	1 116.86	9.90E-07	2.93	1.73E-05	3.68	265.55

Fig. 5, which consists of a slotted disk, a cone as well as a smooth hump, similar to the one used in [35]. It is a challenging test. For instance, the oscillations will be introduced for high-order schemes for solving this problem since its solution is discontinuous. Because of this, we adopt a simple WENO limiter in [59] for all schemes. We simulate this problem after six full revolutions and report the numerical solutions in Fig. 6. To better compare the performance of schemes, we plot 1D cuts of the numerical solutions along with the exact solution in Fig. 7. From Figs. 6 and 7, we make a few observations: (i) the solutions of all schemes with the WENO limiter are non-oscillatory; (ii) the nonsplitting  $P^2$  SLDG-QC is able to better resolve solution structure, compared with the splitting  $Q^2$  SLDG-split; (iii) the solution of the SLDG scheme with larger time step (CFL = 10.2) dissipates less than that of the SLDG scheme with smaller time step (CFL = 2.2).

**Example 3.3** (Swirling deformation flow) We consider solving

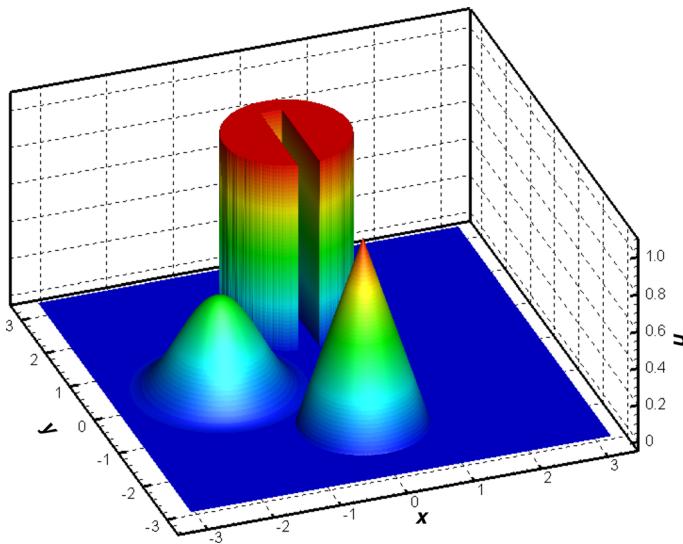
$$u_t - \left( \cos^2 \left( \frac{x}{2} \right) \sin(y) g(t) u \right)_x + \left( \sin(x) \cos^2 \left( \frac{y}{2} \right) g(t) u \right)_y = 0, \quad (x, y) \in [-\pi, \pi]^2, \quad (3.4)$$

**Table 3**  $P^k$  SLDG and  $Q^k$  SLDG-split ( $k = 1, 2$ ) for (3.3) with  $u(x, y, 0) = \exp(-x^2 - 10y^2)$  at  $T = 20\pi$ 

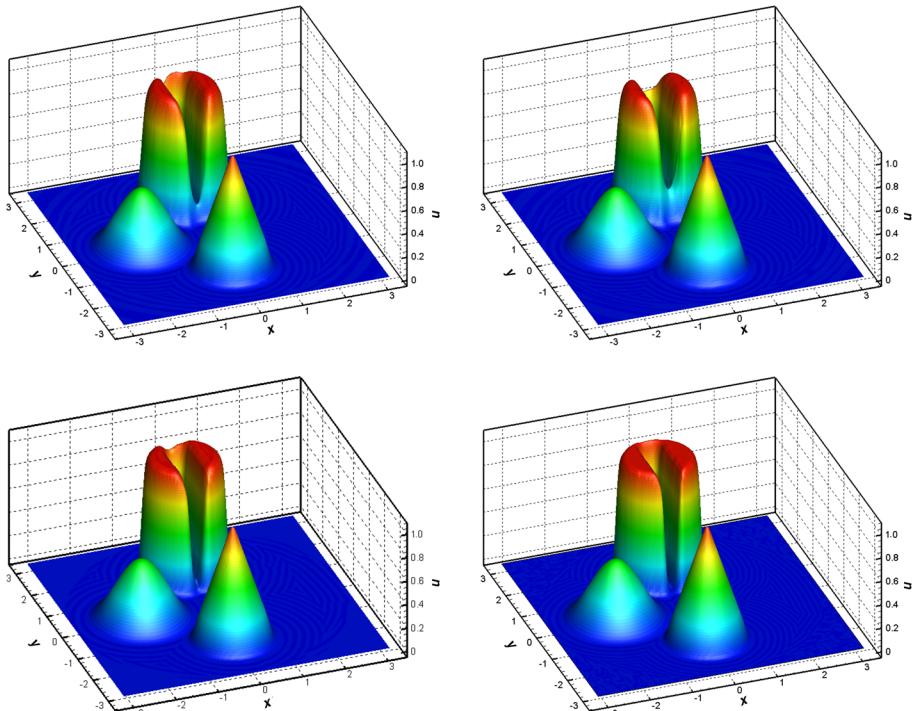
Mesh	CFL = 2.5					CFL = 10.5				
	$L^2$ error	Order	$L^\infty$ error	Order	CPU	$L^2$ error	Order	$L^\infty$ error	Order	CPU
<b><math>P^1</math> SLDG</b>										
20 <sup>2</sup>	4.46E–02	–	8.11E–01	–	1.05	3.99E–02	–	7.25E–01	–	0.33
40 <sup>2</sup>	3.27E–02	0.45	5.77E–01	0.49	7.86	2.69E–02	0.57	4.87E–01	0.57	2.05
80 <sup>2</sup>	1.78E–02	0.88	2.87E–01	1.00	61.98	1.27E–02	1.08	2.23E–01	1.13	15.42
160 <sup>2</sup>	5.93E–03	1.59	1.03E–01	1.48	478.59	3.43E–03	1.89	6.65E–02	1.74	119.97
<b><math>Q^1</math> SLDG-split</b>										
20 <sup>2</sup>	3.14E–02	–	3.86E–01	–	0.89	3.46E–02	–	3.80E–01	–	0.14
40 <sup>2</sup>	2.62E–02	0.26	4.79E–01	–0.31	7.11	2.42E–02	0.52	4.33E–01	–0.19	1.73
80 <sup>2</sup>	1.39E–02	0.92	2.32E–01	1.05	53.70	1.15E–02	1.07	2.03E–01	1.09	12.86
160 <sup>2</sup>	4.17E–03	1.73	7.54E–02	1.62	422.23	3.00E–03	1.94	5.77E–02	1.81	103.75
<b><math>P^2</math> SLDG</b>										
20 <sup>2</sup>	2.73E–02	–	4.39E–01	–	2.19	3.49E–02	–	3.56E–01	–	0.63
40 <sup>2</sup>	1.09E–02	1.33	1.75E–01	1.33	16.61	7.06E–03	1.68	1.25E–01	1.50	4.45
80 <sup>2</sup>	1.57E–03	2.80	2.79E–02	2.65	137.16	7.09E–04	3.31	1.48E–02	3.08	33.59
160 <sup>2</sup>	7.93E–05	4.30	1.77E–03	3.98	1 095.58	3.17E–05	4.48	9.09E–04	4.02	261.95
<b><math>Q^2</math> SLDG-split</b>										
20 <sup>2</sup>	2.23E–02	–	3.67E–01	–	2.38	3.63E–02	–	4.06E–01	–	0.55
40 <sup>2</sup>	7.10E–03	1.65	1.15E–01	1.67	18.47	1.44E–02	1.33	2.01E–01	1.02	4.44
80 <sup>2</sup>	7.69E–04	3.21	1.37E–02	3.07	146.53	3.62E–03	1.99	5.00E–02	2.01	35.33
160 <sup>2</sup>	6.09E–05	3.66	1.13E–03	3.60	1 107.16	9.00E–04	2.01	1.20E–02	2.06	274.30

**Table 4**  $P^2$  SLDG and  $Q^2$  SLDG-split for (3.3) with  $u(x, y, 0) = \exp(-x^2 - 10y^2)$  at  $T = 20\pi$ . A mesh of  $160 \times 160$  cells is used

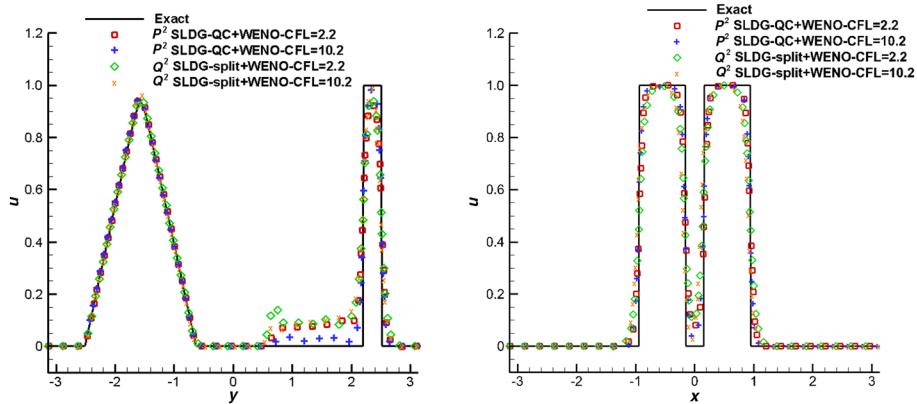
CFL	$L^2$ error	Order	$L^\infty$ error	Order	CPU
<b><math>P^2</math> SLDG</b>					
5	5.62E–05	–	1.33E–03	–	487.08
10	3.30E–05	–0.77	9.15E–04	–0.55	251.50
15	2.45E–05	–0.73	8.63E–04	–0.14	165.55
20	2.09E–05	–0.55	8.16E–04	–0.20	129.88
25	1.95E–05	–0.31	8.56E–04	0.22	109.30
<b><math>Q^2</math> SLDG-split</b>					
5	2.06E–04	–	3.01E–03	–	542.84
10	8.16E–04	1.99	1.09E–02	1.86	271.59
15	1.84E–03	2.00	2.43E–02	1.98	184.47
20	3.26E–03	2.00	4.27E–02	1.96	137.52
25	5.11E–03	2.01	6.70E–02	2.02	109.41



**Fig. 5** Plots of the initial profile. The mesh of  $400 \times 400$  is used



**Fig. 6** Plots of the numerical solutions of SLDG schemes for solving Eq. (3.3) with initial data Fig. 5 up to  $T = 12\pi$ . The mesh of  $100 \times 100$  is used. Upper left:  $Q^2$  SLDG-split+WENO with CFL = 2.2. Upper right:  $Q^2$  SLDG-split+WENO with CFL = 10.2. Bottom left:  $P^2$  SLDG-QC+WENO with CFL = 2.2. Bottom right:  $P^2$  SLDG-QC+WENO with CFL = 10.2



**Fig. 7** Plots of 1D cuts of the numerical solution for SLDG schemes for solving Eq. (3.3) with initial data Fig. 5 up to  $T = 12\pi$ . The mesh of  $100 \times 100$  is used. Left: numerical solution at  $x = 0 + \pi/50$ . Right: numerical solution at  $y = \pi/2 + \pi/50$

where  $g(t) = \cos\left(\frac{\pi t}{T}\right)\pi$ ,  $T = 1.5$ . The initial condition is set to be the following smooth cosine bells (with  $C^5$  smoothness):

$$u(x, y, 0) = \begin{cases} r_0^b \cos^6\left(\frac{r^b}{2r_0^b}\pi\right), & \text{if } r^b < r_0^b, \\ 0, & \text{otherwise,} \end{cases} \quad (3.5)$$

where  $r_0^b = 0.3\pi$ , and  $r^b = \sqrt{(x - x_0^b)^2 + (y - y_0^b)^2}$  denotes the distance between  $(x, y)$  and the center of the cosine bell  $(x_0^b, y_0^b) = (0.3\pi, 0)$ . Note that this swirling deformation flow introduced in [35] is more challenging than the rigid body rotation due to the space- and time-dependent flow field. In particular, along the direction of the flow, the initial function becomes largely deformed at  $t = T/2$ , then goes back to its initial shape at  $t = T$  as the flow reverses. If this problem is solved up to  $T$ , we call such a procedure one full evolution. Note that unlike the previous two examples, the upstream cells are no longer rectangular, and hence the quadratic-curved quadrilateral approximation is expected to capture the geometry of the upstream cells more accurately than the quadrilateral approximation. However, regarding the efficiency, the quadrilateral approximation is already adequate for  $k = 1$  since the DG approximation is second-order accurate. The efficiency benefit of using more costly quadratic-curved quadrilateral approximation is more evident for  $k = 2$ .

We test accuracy for  $P^k$  SLDG(-QC) and  $Q^k$  SLDG-split for  $k = 1, 2$  up to  $T/2$ , i.e., half of one full evolution, and summarize the results in Table 5. Note that as the exact solution is not available, we pre-compute a reference solution by  $P^2$  SLDG-QC with a refined mesh of  $320 \times 320$  cells and CFL = 2.5. As expected, the  $(k+1)$ th-order convergence is observed for  $P^k$  SLDG(-QC) and the second-order convergence is observed due to the splitting error. More specifically, when comparing  $P^1$  SLDG and  $Q^1$  SLDG-split, error magnitude from the splitting one is relatively smaller, which again is ascribed to the fact that despite the same convergence rate,  $Q^1$  delivers better approximation performance than  $P^1$ . Further, when comparing  $P^2$  SLDG-QC and  $Q^2$  SLDG-split, the splitting method does not perform as well as the non-splitting one in terms of error magnitude, due to its splitting error.

**Table 5** Swirling deformation flow.  $P^k$  SLDG(-QC) and  $Q^k$  SLDG-split ( $k = 1, 2$ ) for (3.4) with the smooth cosine bells (3.5) at  $T/2 = 0.75$ . A reference solution is solved by  $P^2$  SLDG-QC with CFL = 2.5 and a mesh of  $320 \times 320$  cells

Mesh	CFL = 2.5					CFL = 10.5				
	$L^2$ error	Order	$L^\infty$ error	Order	CPU	$L^2$ error	Order	$L^\infty$ error	Order	CPU
$P^1$ SLDG										
20 <sup>2</sup>	1.08E–02	–	2.49E–01	–	0.03	9.27E–03	–	2.34E–01	–	0.02
40 <sup>2</sup>	2.98E–03	1.86	1.04E–01	1.26	0.25	3.50E–03	1.40	1.31E–01	0.83	0.09
80 <sup>2</sup>	7.88E–04	1.92	3.23E–02	1.69	1.70	8.47E–04	2.05	3.78E–02	1.80	0.53
160 <sup>2</sup>	2.14E–04	1.88	9.02E–03	1.84	12.98	2.30E–04	1.88	1.28E–02	1.56	3.81
$Q^1$ SLDG-split										
20 <sup>2</sup>	6.68E–03	–	1.42E–01	–	0.03	3.18E–02	–	4.19E–01	–	0.02
40 <sup>2</sup>	1.66E–03	2.01	4.37E–02	1.70	0.23	6.27E–03	2.34	9.15E–02	2.20	0.06
80 <sup>2</sup>	4.32E–04	1.95	1.10E–02	1.99	1.75	1.48E–03	2.08	2.27E–02	2.01	0.44
160 <sup>2</sup>	1.13E–04	1.94	2.68E–03	2.04	13.88	3.69E–04	2.01	6.00E–03	1.92	3.48
$P^2$ SLDG-QC										
20 <sup>2</sup>	2.43E–03	–	9.30E–02	–	0.06	2.76E–03	–	9.55E–02	–	0.02
40 <sup>2</sup>	3.44E–04	2.82	2.13E–02	2.13	0.42	4.54E–04	2.60	1.71E–02	2.48	0.14
80 <sup>2</sup>	4.56E–05	2.91	3.08E–03	2.79	3.27	5.52E–05	3.04	2.41E–03	2.83	0.86
160 <sup>2</sup>	5.80E–06	2.98	3.70E–04	3.06	25.58	6.21E–06	3.15	3.04E–04	2.99	6.36
$Q^2$ SLDG-split										
20 <sup>2</sup>	1.56E–03	–	2.60E–02	–	0.06	3.32E–02	–	4.36E–01	–	0.02
40 <sup>2</sup>	3.53E–04	2.14	6.47E–03	2.01	0.45	6.26E–03	2.41	8.89E–02	2.30	0.11
80 <sup>2</sup>	8.36E–05	2.08	1.43E–03	2.18	3.69	1.47E–03	2.09	2.12E–02	2.06	0.92
160 <sup>2</sup>	2.05E–05	2.03	3.29E–04	2.11	29.17	3.60E–04	2.03	5.18E–03	2.03	7.30

**Table 6** Swirling deformation flow. Order of accuracy in time for  $P^2$  SLDG-QC and  $Q^2$  SLDG-split for (3.4) with the smooth cosine bells (3.5) at  $T/2 = 0.75$  by comparing numerical solutions with a reference solution from the corresponding scheme with CFL = 0.1. A mesh of  $160 \times 160$  is used

CFL	$L^2$ error	Order	$L^\infty$ error	Order	CPU
$P^2$ SLDG-QC					
5	1.85E–06	–	7.34E–05	–	8.48
10	4.13E–06	1.16	2.16E–04	1.56	4.38
15	7.22E–06	1.38	4.74E–04	1.94	3.05
20	1.13E–05	1.55	8.10E–04	1.86	2.34
25	1.36E–05	0.84	8.33E–04	0.13	1.98
$Q^2$ SLDG-split					
5	8.09E–05	–	1.18E–03	–	9.45
10	3.23E–04	2.00	4.67E–03	1.99	4.75
15	7.28E–04	2.00	1.06E–02	2.01	3.17
20	1.30E–03	2.01	1.88E–02	2.01	2.36
25	2.08E–03	2.11	3.02E–02	2.11	2.00

We then perform the accuracy test in time with a fixed mesh of  $160 \times 160$  cells and different CFLs ranging from 5 to 25. The reference solution is computed by  $P^2$  SLDG-QC with the same mesh but using a small CFL = 0.1. We report the results in Table 6. The

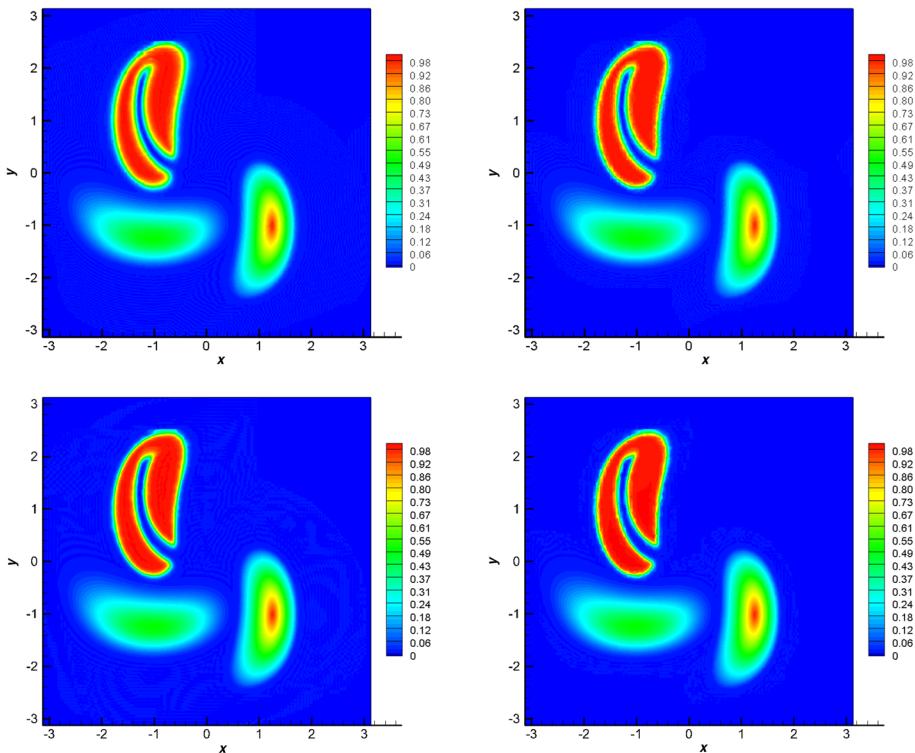
**Table 7** Swirling deformation flow.  $P^k$  SLDG(-QC) and  $Q^k$  SLDG-split ( $k = 1, 2$ ) for (3.4) with the smooth cosine bells (3.5) at  $T = 1.5$ 

Mesh	CFL = 2.5					CFL = 10.5				
	$L^2$ error	Order	$L^\infty$ error	Order	CPU	$L^2$ error	Order	$L^\infty$ error	Order	CPU
<i>P</i> <sup>1</sup> SLDG										
20 <sup>2</sup>	1.41E–02	–	2.94E–01	–	0.08	1.06E–02	–	2.11E–01	–	0.01
40 <sup>2</sup>	3.85E–03	1.87	9.33E–02	1.65	0.44	5.47E–03	0.95	2.03E–01	0.05	0.13
80 <sup>2</sup>	9.55E–04	2.01	3.12E–02	1.58	3.55	1.57E–03	1.80	9.23E–02	1.14	0.89
160 <sup>2</sup>	2.60E–04	1.87	1.33E–02	1.23	27.58	6.13E–04	1.36	3.15E–02	1.55	6.61
<i>Q</i> <sup>1</sup> SLDG-split										
20 <sup>2</sup>	9.88E–03	–	1.87E–01	–	0.06	1.10E–02	–	1.55E–01	–	0.02
40 <sup>2</sup>	2.15E–03	2.20	5.41E–02	1.79	0.50	2.73E–03	2.02	4.75E–02	1.71	0.14
80 <sup>2</sup>	4.03E–04	2.41	1.28E–02	2.08	4.04	3.86E–04	2.82	7.45E–03	2.67	1.00
160 <sup>2</sup>	8.47E–05	2.25	2.75E–03	2.22	31.63	5.50E–05	2.81	1.41E–03	2.40	7.56
<i>P</i> <sup>2</sup> SLDG-QC										
20 <sup>2</sup>	2.68E–03	–	5.78E–02	–	0.14	3.38E–03	–	1.11E–01	–	3.13
40 <sup>2</sup>	3.72E–04	2.85	1.23E–02	2.23	0.83	5.50E–04	2.62	1.64E–02	2.75	0.22
80 <sup>2</sup>	5.69E–05	2.71	3.12E–03	1.98	6.38	7.99E–05	2.78	4.50E–03	1.87	1.64
160 <sup>2</sup>	8.56E–06	2.73	6.63E–04	2.23	50.78	1.21E–05	2.73	6.46E–04	2.80	12.30
<i>Q</i> <sup>2</sup> SLDG-split										
20 <sup>2</sup>	1.13E–03	–	1.97E–02	–	0.14	1.16E–02	–	1.74E–01	–	0.03
40 <sup>2</sup>	1.26E–04	3.17	2.85E–03	2.79	1.05	2.55E–03	2.19	3.57E–02	2.28	0.25
80 <sup>2</sup>	1.49E–05	3.08	4.15E–04	2.78	8.19	2.95E–04	3.11	4.19E–03	3.09	2.03
160 <sup>2</sup>	1.86E–06	3.01	4.91E–05	3.08	65.86	2.39E–05	3.63	3.84E–04	3.45	15.53

second-order Strang splitting error is clearly observed in Table 6. It is also observed that,  $P^2$  SLDG-QC significantly outperforms  $Q^2$  SLDG-split in terms of error magnitude. The two methods consume comparable CPU time with the same simulation configuration.

We also test the convergence for one full evolution and summarize the results in Table 7 for  $P^k$  SLDG(-QC) and  $Q^k$  SLDG-split,  $k = 1, 2$ , with CFL = 2.5 and CFL = 10.5. It is observed that the errors after one full evolution are less than those at one half evolution. Such a super performance is ascribed to the special error cancellations (coming from some symmetry in the solution movement along the flow) as explained in [4]. In addition, the error cancellation also makes the splitting error less pronounced for  $Q^k$  SLDG-split. This is an example, in which special cancellation of errors plays a role in comparing schemes' performance.

Finally, we study the performance of the splitting and nonsplitting SLDG methods for solving the swirling deformation flow (3.4) with the initial data in Fig. 5. As the previous example, we adopt the WENO limiter for all schemes. We present the numerical results in Figs. 8, 9, 10. The similar observations as Example 3.2 can be made.

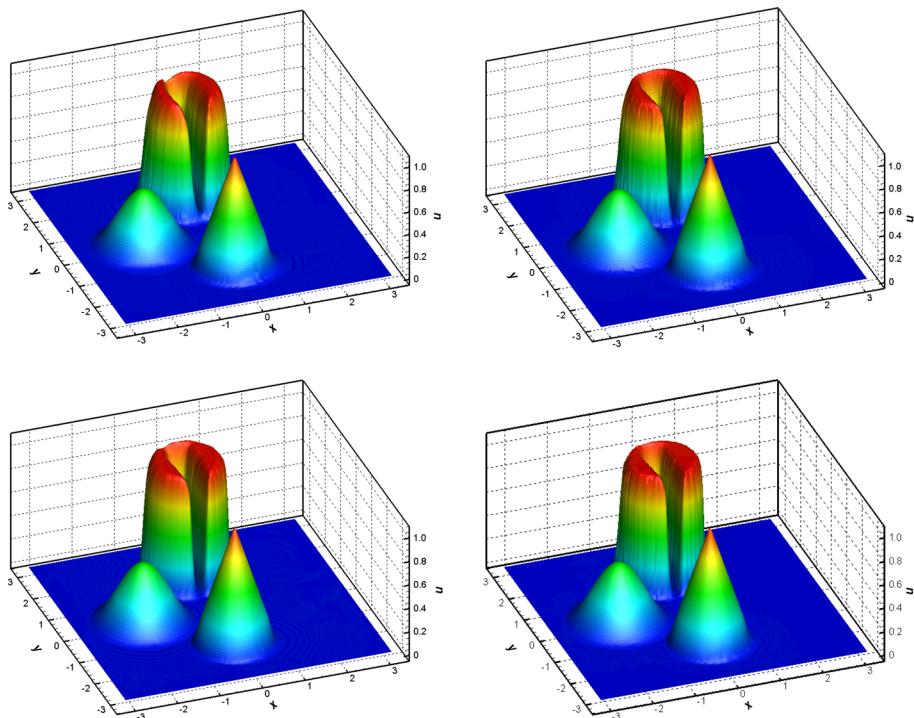


**Fig. 8** Plots of the numerical solutions of SLDG schemes for solving the swirling deformation flow (3.4) with initial data Fig. 5.  $T = 1.5$  and the final integration time is 0.75. The mesh of  $100 \times 100$  is used. Upper left:  $Q^2$  SLDG-split+WENO with CFL = 2.2. Upper right:  $Q^2$  SLDG-split+WENO with CFL = 10.2. Bottom left:  $P^2$  SLDG-QC+WENO with CFL = 2.2. Bottom right:  $P^2$  SLDG-QC+WENO with CFL = 10.2

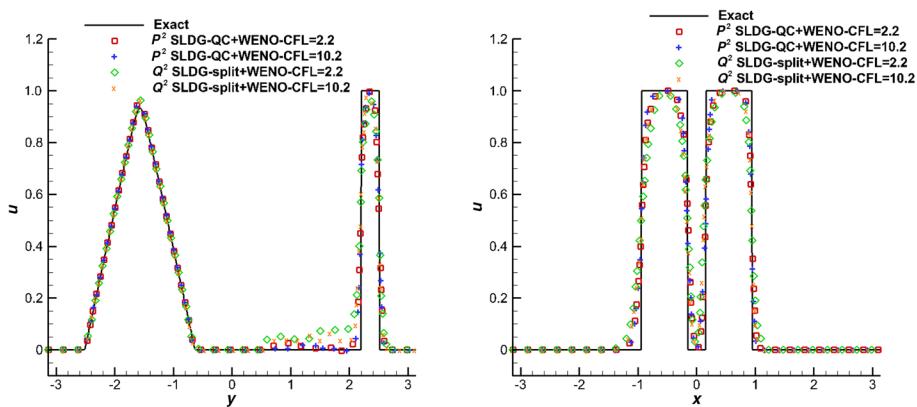
### 3.2 Vlasov-Poisson System

In this subsection, we compare the performance of the splitting and non-splitting SLDG methods for solving the 1D1V VP system (2.17). We focus on the strong Landau damping test. The observations for other benchmark tests including the two stream instability I [21], two stream instability II [49] and bump-on-tail instability [1, 25] are similar, thus are omitted for brevity. For the non-splitting SLDG method, we employ high-order characteristics tracing schemes as proposed in [8]. In particular, we couple  $P^1$  SLDG with a second order characteristic tracing scheme and couple  $P^2$  SLDG-QC with a third order characteristic tracing scheme, and denote the resulting methods as  $P^1$  SLDG-time2 and  $P^2$  SLDG-QC-time3, respectively. We apply the positivity-preserving limiter [58] for all nonlinear test examples.

**Example 3.4** (Strong Landau damping) Consider the strong Landau damping for the VP system with the initial condition being a perturbed Maxwellian equilibrium



**Fig. 9** Plots of the numerical solutions of SLDG schemes for solving the swirling deformation flow (3.4) with initial data Fig. 5.  $T = 1.5$  and the final integration time is 1.5. The mesh of  $100 \times 100$  is used. Upper left:  $Q^2$  SLDG-split+WENO with CFL = 2.2. Upper right:  $Q^2$  SLDG-split+WENO with CFL = 10.2. Bottom left:  $P^2$  SLDG-QC+WENO with CFL = 2.2. Bottom right:  $P^2$  SLDG-QC+WENO with CFL = 10.2



**Fig. 10** Plots of 1D cuts of the numerical solution for SLDG schemes for solving the swirling deformation flow (3.4) with initial data Fig. 5. The mesh of  $100 \times 100$  is used. Left: numerical solution at  $x = 0 + \pi/100$ . Right: numerical solution at  $y = \pi/2 + \pi/100$

**Table 8** Strong Landau damping. Temporal order of convergence and CPU comparison between  $Q^k$  SLDG-split and  $P^k$  SLDG-(QC)-time ( $k+1$ ) by comparing numerical solutions with a reference solution from the corresponding scheme with CFL = 0.01.  $T = 2$ . The mesh of  $200 \times 200$  cells is used

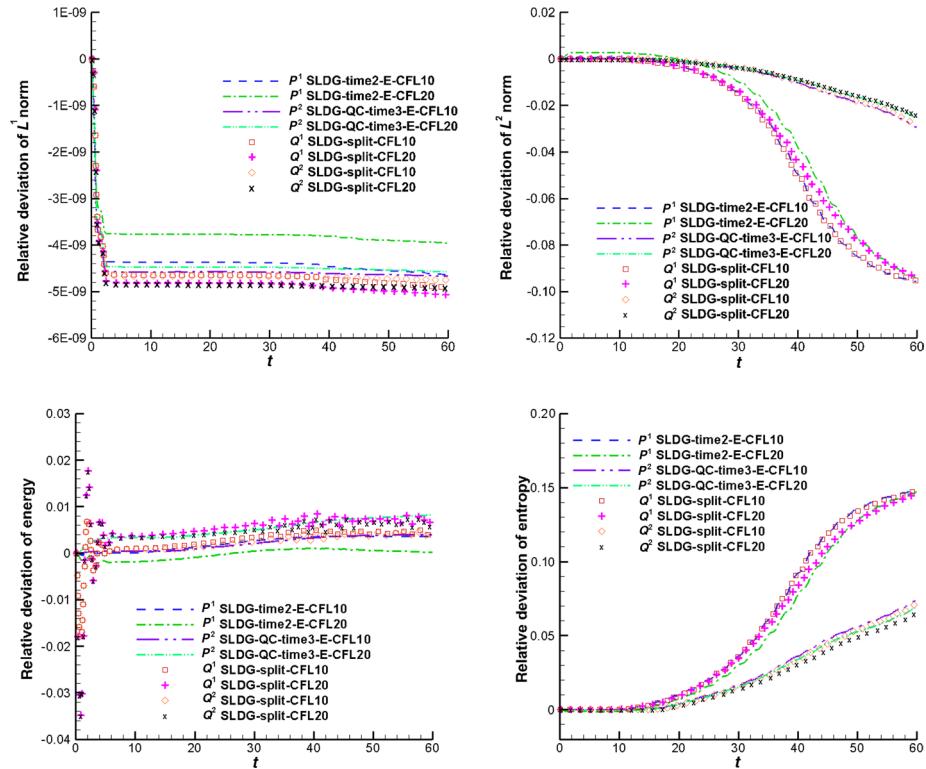
CFL	$L^2$ error	Order	$L^\infty$ error	Order	CPU	$L^2$ error	Order	$L^\infty$ error	Order	CPU
$Q^1$ SLDG-split										
5	1.70E-05		1.10E-04		4.70	1.44E-04		7.09E-04		10.49
10	4.67E-05	1.46	2.71E-04	1.31	2.39	5.63E-04	1.97	2.56E-03	1.85	6.29
15	9.03E-05	1.63	4.01E-04	0.96	1.64	1.26E-03	1.98	5.60E-03	1.93	3.74
20	1.61E-04	2.01	7.53E-04	2.19	1.21	2.24E-03	2.01	1.01E-02	2.04	2.77
25	2.51E-04	2.00	1.04E-03	1.47	1.02	3.44E-03	1.93	1.54E-02	1.91	2.49
$Q^2$ SLDG-split										
5	9.40E-06		3.67E-05		16.29	5.07E-06		5.40E-05		28.10
10	3.80E-05	2.01	1.46E-04	1.99	8.16	1.83E-05	1.85	1.66E-04	1.62	11.40
15	8.70E-05	2.04	3.31E-04	2.03	5.60	5.54E-05	2.73	3.39E-04	1.76	7.85
20	1.44E-04	1.76	5.54E-04	1.78	4.11	1.32E-04	3.03	7.39E-04	2.71	5.89
25	2.41E-04	2.30	9.10E-04	2.22	3.40	2.55E-04	2.94	1.38E-03	2.78	4.77

$$f(x, v, t=0) = \frac{1}{\sqrt{2\pi}}(1 + \alpha \cos(k_0 x)) \exp\left(-\frac{v^2}{2}\right), \quad (3.6)$$

where  $\alpha = 0.5$  and  $k_0 = 0.5$ . This problem has been numerically investigated by several authors, e.g., see [15, 32, 44, 51, 60]. First, we perform the comparison for  $Q^k$  SLDG-split and  $P^k$  SLDG-(QC)-time( $k+1$ ),  $k = 1, 2$ . The results are reported in Table 8. In the simulation, we use a fixed mesh of  $200 \times 200$  cells. We let  $T = 2$  and CFL range from 5 to 25. The errors are computed by comparing numerical solutions with the reference solution by  $P^2$  SLDG-QC-time3 with the same mesh but using a small CFL = 0.01. Expected orders of convergence are observed. By taking a closer look at the error table, we observe that the error magnitude from both methods is comparable, while the splitting method takes less CPU time. The methods also perform similarly in conserving physical invariants and in resolving solution structures, see Figs. 11 and 12, respectively. The splitting scheme outperforms the non-splitting one, as the temporal splitting error is not dominating, and  $Q^k$  provides better approximations than  $P^k$  polynomials.

### 3.3 2D Incompressible Euler Equations and the Guiding Center Vlasov Model

In this subsection, we compare the performance of the SLDG methods for solving the 2D incompressible Euler equation in the vorticity-stream function formulation (2.22) and the guiding center Vlasov model (2.23). As with the VP system, we have to couple a high-order characteristics tracing mechanism for the non-splitting SLDG method. For the notational convenience, below  $P^k$  SLDG-(QC)- $P^r$  LDG-time( $k+1$ )-CFLs denotes the non-splitting SLDG scheme with  $P^k$  polynomial space, using the  $P^r$  LDG scheme for solving Poisson's equation, the  $(k+1)$ th order scheme in characteristics tracing, and CFL =  $s$ . As



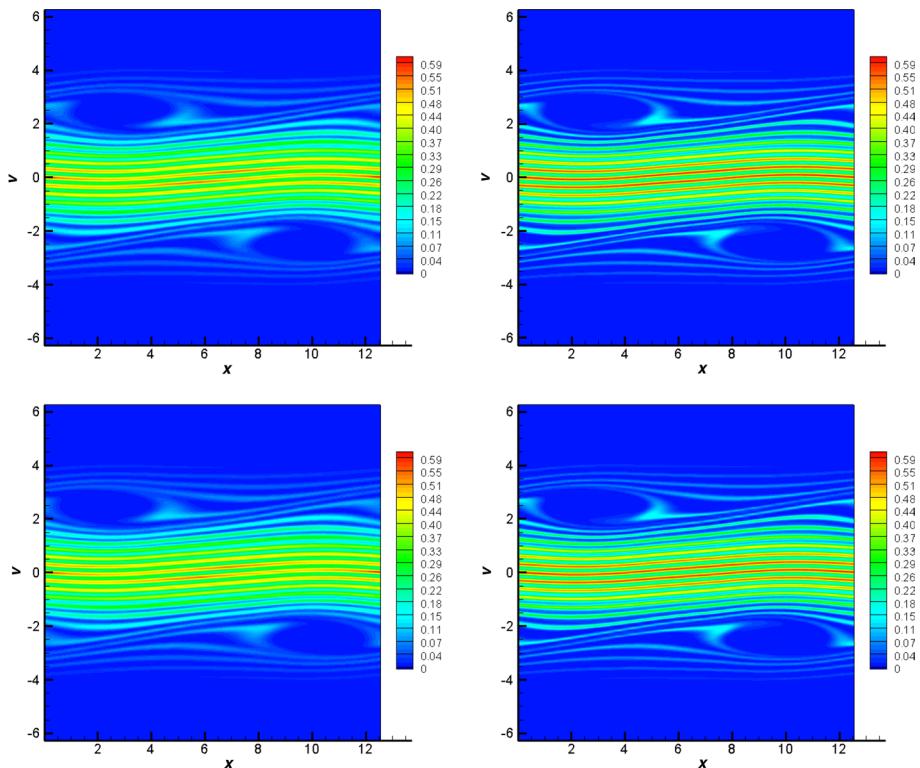
**Fig. 11** Strong Landau damping. Time evolution of the relative deviations of  $L^1$  (upper left) and  $L^2$  (upper right) norms of the solution as well as the discrete kinetic energy (lower left) and entropy (lower right)

mentioned in [9], the choice of using the  $P^k$  or  $P^{k+1}$  LDG solver, i.e.,  $r = k$  or  $k + 1$ , for Poisson's equation is a trade-off between accuracy (effectiveness in resolving solutions) and CPU cost. In the simulations, we let  $r = k + 1$  for the accuracy test only and let  $r = k$  otherwise. See [9] for more detailed discussion.

**Example 3.5** (Accuracy and convergence test) Consider the incompressible Euler equation (2.22) on the domain  $[0, 2\pi] \times [0, 2\pi]$  with the initial condition

$$\omega(x, y, 0) = -2 \sin(x) \sin(y) \quad (3.7)$$

and periodic boundary conditions. The exact solution stays stationary as  $\omega(x, y, t) = -2 \sin(x) \sin(y)$ . Here we solve the problem up to  $T = 1$  with  $\text{CFL} = 1$ . We test the accuracy and CPU cost for  $Q^k$  SLDG-split- $P^{k+1}$  LDG-CFL1 and  $P^k$  SLDG(-QC)- $P^{k+1}$  LDG-time( $k + 1$ )-CFL1,  $k = 1, 2$ , and summarize results in Table 9. Expected second- and third-order convergence is observed for  $P^1$  SLDG- $P^2$  LDG-time2 and  $P^2$  SLDG-QC- $P^3$  LDG-time3, respectively. Only first order of convergence is observed for the splitting scheme due to the first-order error in tracing characteristics, even if the second-order Strang splitting is used. It is evident that the non-splitting method is more efficient due to its genuine high-order accuracy in both space and time.



**Fig. 12** Strang Landau damping with the spatial mesh of  $160 \times 160$ .  $T = 40$ , CFL = 10. Upper left:  $Q^1$  SLDG-split. Upper right:  $Q^2$  SLDG-split. Bottom left:  $P^1$  SLDG-time2. Bottom right:  $P^2$  SLDG-QC-time3

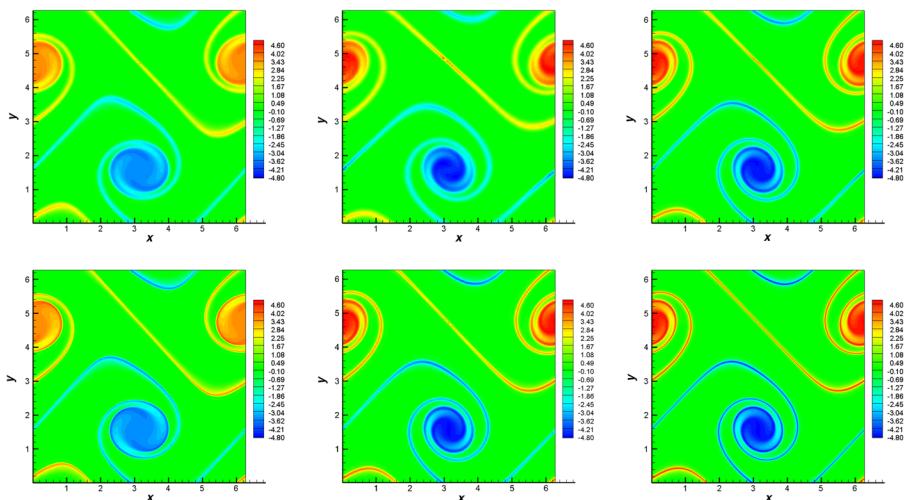
**Table 9** The incompressible Euler equations. Errors, orders and CPU times (sec) comparison between  $Q^k$  SLDG-split and  $P^k$  SLDG(-QC) ( $k = 1, 2$ ). CFL = 1.  $T = 1$

Mesh	$L^2$ error	Order	$L^\infty$ error	Order	CPU	$L^2$ error	Order	$L^\infty$ error	Order	CPU
$Q^1$ SLDG-split+ $P^2$ LDG					$P^1$ SLDG+ $P^2$ LDG+time2					
$20^2$	4.06E-02		1.30E-01		0.23	1.57E-02		8.55E-02		0.21
$40^2$	2.00E-02	1.03	6.18E-02	1.07	2.97	8.55E-02	1.97	2.48E-02	1.78	2.47
$60^2$	1.34E-02	0.98	4.09E-02	1.02	16.55	2.48E-02	1.98	1.16E-02	1.87	12.68
$80^2$	1.00E-02	1.02	3.03E-02	1.05	75.91	1.02E-03	1.96	6.71E-03	1.91	49.12
$100^2$	8.03E-03	0.98	2.42E-02	1.00	193.45	6.49E-04	2.01	4.34E-03	1.95	131.11
$Q^2$ SLDG-split+ $P^3$ LDG					$P^2$ SLDG-QC+ $P^3$ LDG+time3					
$20^2$	4.04E-02		1.17E-01		0.67	2.82E-03		1.36E-02		1.63
$40^2$	1.99E-02	1.02	5.80E-02	1.02	8.82	3.56E-04	2.99	1.93E-03	2.81	17.63
$60^2$	1.34E-02	0.98	3.91E-02	0.97	59.58	1.04E-04	3.02	5.33E-04	3.18	113.77
$80^2$	1.00E-02	1.02	2.91E-02	1.02	205.18	4.44E-05	2.97	2.28E-04	2.96	338.95
$100^2$	8.03E-03	0.98	2.34E-02	0.98	667.46	2.24E-05	3.08	1.14E-04	3.09	1 003.01

**Example 3.6** (The double shear layer problem [3, 38, 58]) We consider the model problem (2.22) in the domain  $[0, 2\pi] \times [0, 2\pi]$ , with periodic boundary conditions and the initial condition

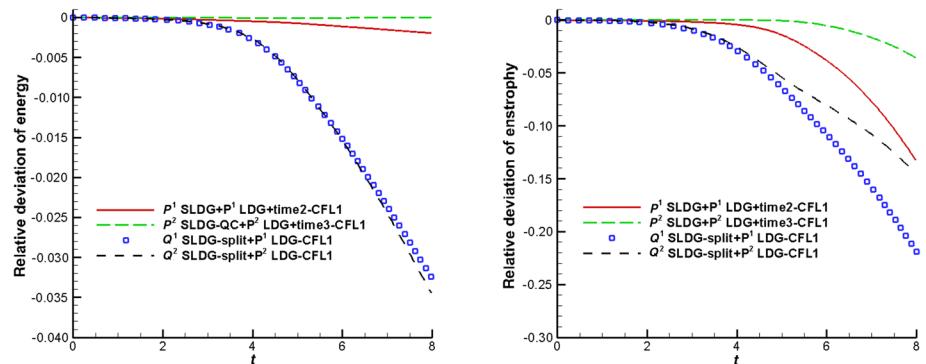
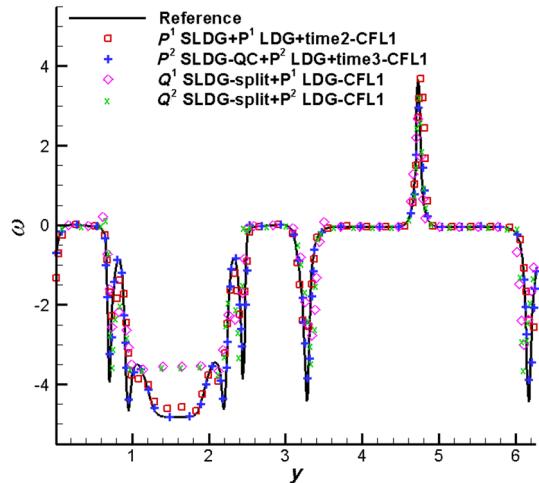
$$\omega(x, y, 0) = \begin{cases} \delta \cos(x) - \frac{1}{\rho} \operatorname{sech}^2\left(\frac{y-\pi/2}{\rho}\right), & \text{if } y \leq \pi, \\ \delta \cos(x) + \frac{1}{\rho} \operatorname{sech}^2\left(\frac{3\pi/2-y}{\rho}\right), & \text{if } y > \pi, \end{cases} \quad (3.8)$$

where  $\delta = 0.05$  and  $\rho = \pi/15$ . As time evolves, the shear layers quickly develop into roll-ups with thinner and thinner scales. Eventually, with a fixed mesh, the full resolution of the structures will be lost for any methods. This problem has been tested by the high-order Eulerian finite difference ENO/WENO method in [16, 27], the high-order SL WENO scheme in [13, 42, 52], the DG method in [37, 58, 61] and the spectral element method in [23, 54]. We solve this problem up to  $T = 8$  and present the surface plots of  $\omega$  for both SLDG methods in Fig. 13. The solutions by the splitting scheme (left) and non-splitting scheme (middle) are compared against the reference solution (right) computed by the same methods with a doubly refined mesh. In Fig. 14, we further compare the 1D cuts of the numerical solutions along  $x = \pi$ . It is observed that the non-splitting SLDG method performs much better in correctly resolving solution structures than the splitting counterpart. We also plot the time history of the relative deviations of energy and enstrophy in Fig. 15, using a mesh of  $100 \times 100$  cells and  $\text{CFL} = 1$ . Again, the non-splitting SLDG method does a better job of conserving the physical invariants than the splitting counterpart.



**Fig. 13** Surface plots of the numerical solutions for the shear flow at  $T = 8$ . A mesh of  $100 \times 100$  cells is used for the left and middle plots, in comparison to the reference solution using a mesh of  $200 \times 200$  cells (right plot).  $\text{CFL} = 1$ . Upper left:  $Q^1$  SLDG-split- $P^1$  LDG. Upper middle:  $P^1$  SLDG- $P^1$  LDG-time2. Upper right:  $P^1$  SLDG- $P^1$  LDG-time2. Bottom left:  $Q^2$  SLDG-split- $P^2$  LDG. Bottom middle:  $P^2$  SLDG-QC- $P^2$  LDG-time3. Bottom right:  $P^2$  SLDG-QC- $P^2$  LDG-time3

**Fig. 14** 1D cut along  $x = \pi$  for the shear flow problem at  $T = 8$ . The mesh of  $100 \times 100$  cells is used. Lines: a reference solution is obtained by  $P^2$  SLDG-QC+ $P^2$  LDG+time3 with CFL = 1 and using a mesh of  $200 \times 200$  cells

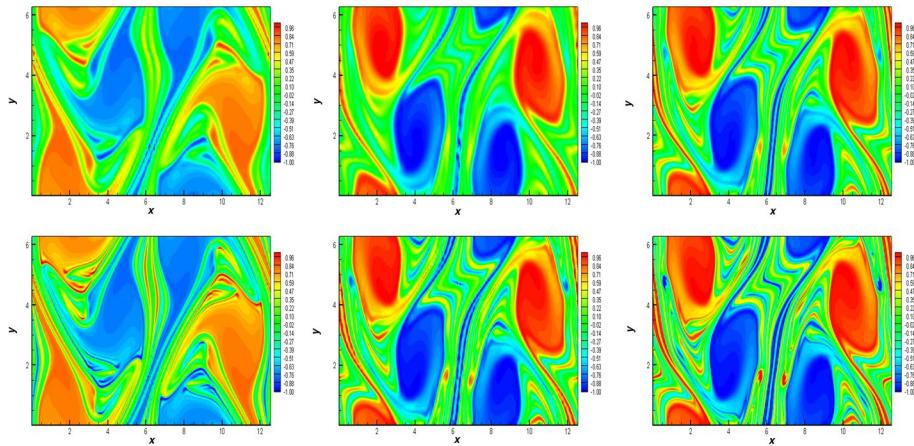


**Fig. 15** Time evolution of the relative deviation of energy (left) and enstrophy (right) for the SLDG method with or without operator splitting for the double shear layer problem. The mesh of  $100 \times 100$  cells is used

**Example 3.7** (Kelvin-Helmholtz instability) The last example is the 2D guiding center model problem with the initial condition

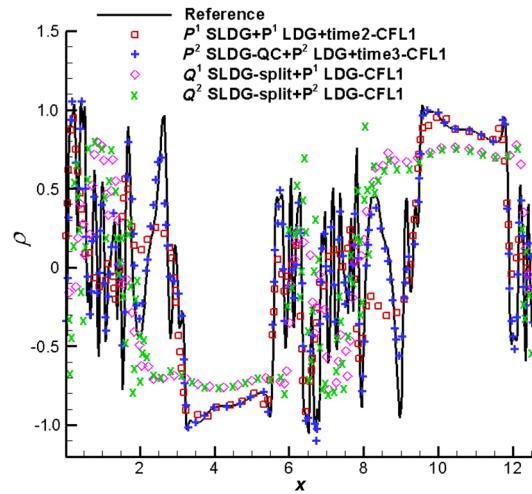
$$\rho_0(x, y) = \sin(y) + 0.015 \cos(k_0 x) \quad (3.9)$$

and periodic boundary conditions on the domain  $[0, 4\pi] \times [0, 2\pi]$ . We let  $k_0 = 0.5$ , and hence create a Kelvin-Helmholtz instability [47]. In the literature, this problem was well studied by many authors. In Fig. 16, we present the surface plots of the charge density  $\rho$  at  $T = 40$  by the two SLDG methods with a mesh of  $100 \times 100$  elements. The reference solution with a doubly refined mesh is provided as well. We further compare the numerical solutions via their 1D cuts along the line  $y = \pi$  in Fig. 17. As with the previous example, we observe that the solutions by the non-splitting SLDG method match the reference

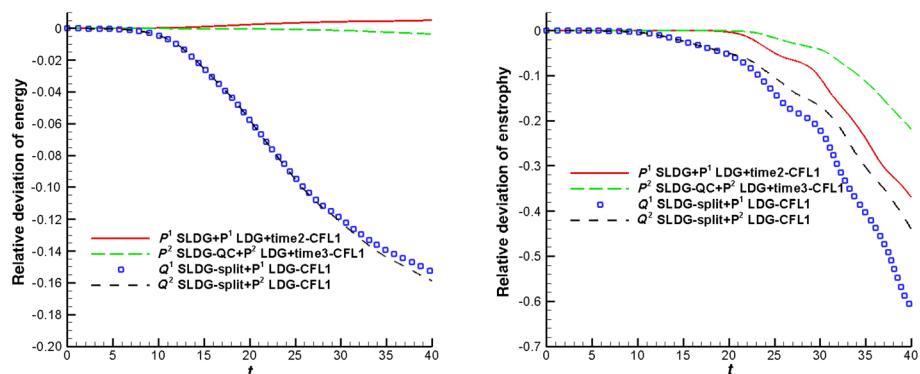


**Fig. 16** Surface plots of the numerical solutions for the Kelvin-Helmholtz instability at  $T = 40$ . A mesh of  $100 \times 100$  cells is used for the left and middle plots, in comparison to the reference solution using a mesh of  $200 \times 200$  cells (right plot). CFL = 1. Upper left:  $Q^1$  SLDG-split- $P^1$  LDG. Upper middle:  $P^1$  SLDG- $P^1$  LDG-time2. Upper right:  $P^1$  SLDG- $P^1$  LDG-time2. Bottom left:  $Q^2$  SLDG-split- $P^2$  LDG. Bottom middle:  $P^2$  SLDG-QC- $P^2$  LDG-time3. Bottom right:  $P^2$  SLDG-QC- $P^2$  LDG-time3

**Fig. 17** 1D cut along  $y = \pi$  for the Kelvin-Helmholtz instability problem at  $T = 40$ . Lines: a reference solution is obtained by  $P^2$  SLDG-QC- $P^2$  LDG-time3 with CFL = 1 and using a mesh of  $200 \times 200$  cells



solution more closely. Furthermore, the non-splitting method is able to better conserve energy and enstrophy of the system than the splitting counterpart, see Fig. 18.



**Fig. 18** Time evolution of the relative deviation of enstrophy for the SLDG method with or without operator splitting for the Kelvin-Helmholtz instability problem. The mesh of  $100 \times 100$  cells is used

## 4 Conclusion

In this paper, we review several SLDG methods and compare their performance for solving 2D linear transport problems, and nonlinear Vlasov and incompressible Euler equations. The performance of SLDG schemes in terms of error magnitude, order of accuracy, CPU cost, resolution of complicated solution structures, as well as their ability to conserve important physical invariants was benchmarked through extensive numerical experiments. In general, when the geometry is simple, and a smaller CFL is needed for accuracy, the scheme based on dimensional splitting together with 1D SLDG formulation is preferred for its simplicity in implementation; when a larger CFL is desired for efficiency without significantly sacrificing accuracy, the truly multi-dimensional SLDG scheme is preferred for its efficiency and efficacy in resolving solutions using a relatively coarse mesh and extra large time stepping sizes.

## Compliance with Ethical Standards

**Conflict of interest** On behalf of all authors, the corresponding author states that there is no conflict of interest.

## References

- Arber, T., Vann, R.: A critical comparison of Eulerian-grid-based Vlasov solvers. *J. Comput. Phys.* **180**(1), 339–357 (2002)
- Arnold, D., Brezzi, F., Cockburn, B., Marini, L.: Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Nume. Anal.* **39**(5), 1749–1779 (2002)
- Bell, J., Colella, P., Glaz, H.: A second-order projection method for the incompressible Navier-Stokes equations. *J. Comput. Phys.* **85**(2), 257–283 (1989)
- Blossey, P., Durran, D.: Selective monotonicity preservation in scalar advection. *J. Comput. Phys.* **227**(10), 5160–5183 (2008)
- Bochev, P., Moe, S., Peterson, K., Ridzal, D.: A conservative, optimization-based semi-Lagrangian spectral element method for passive tracer transport. In: COUPLED PROBLEMS 2015, VI International Conference on Computational Methods for Coupled Problems in Science and Engineering, Barcelona, Spain, pp. 23–34 (2015)

6. Bradley, A.M., Bosler, P.A., Guba, O., Taylor, M.A., Barnett, G.A.: Communication-efficient property preservation in tracer transport. *SIAM J. Sci. Comput.* **41**(3), C161–C193 (2019)
7. Cai, X., Guo, W., Qiu, J.-M.: A high order conservative semi-Lagrangian discontinuous Galerkin method for two-dimensional transport simulations. *J. Sci. Comput.* **73**(2/3), 514–542 (2017)
8. Cai, X., Guo, W., Qiu, J.-M.: A high order semi-Lagrangian discontinuous Galerkin method for Vlasov-Poisson simulations without operator splitting. *J. Comput. Phys.* **354**, 529–551 (2018)
9. Cai, X., Guo, W., Qiu, J.-M.: A high order semi-Lagrangian discontinuous Galerkin method for the two-dimensional incompressible Euler equations and the guiding center Vlasov model without operator splitting. *J. Sci. Comput.* **79**(2), 1111–1134 (2019)
10. Cai, X., Qiu, J., Qiu, J.-M.: A conservative semi-Lagrangian HWENO method for the Vlasov equation. *J. Comput. Phys.* **323**, 95–114 (2016)
11. Celia, M.A., Russell, T.F., Herrera, I., Ewing, R.E.: An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation. *Adv. Water Resources* **13**(4), 187–206 (1990)
12. Cheng, C.-Z., Knorr, G.: The integration of the Vlasov equation in configuration space. *J. Comput. Phys.* **22**(3), 330–351 (1976)
13. Christlieb, A., Guo, W., Morton, M., Qiu, J.-M.: A high order time splitting method based on integral deferred correction for semi-Lagrangian Vlasov simulations. *J. Comput. Phys.* **267**, 7–27 (2014)
14. Crouseilles, N., Mehrenberger, M., Sonnendrücker, E.: Conservative semi-Lagrangian schemes for Vlasov equations. *J. Comput. Phys.* **229**(6), 1927–1953 (2010)
15. Crouseilles, N., Mehrenberger, M., Vecil, F.: Discontinuous Galerkin semi-Lagrangian method for Vlasov-Poisson. In: *ESAIM: Proceedings*, vol. 32, pp. 211–230. EDP Sciences, (2011)
16. W. E., Shu, C.-W.: A numerical resolution study of high order essentially non-oscillatory schemes applied to incompressible flow. *J. Comput. Phys.* **110**(1), 39–46 (1994)
17. Einkemmer, L.: A study on conserving invariants of the Vlasov equation in semi-Lagrangian computer simulations. *J. Plasma Phys.* **83**(2), 705830203 (2017)
18. Einkemmer, L.: A performance comparison of semi-Lagrangian discontinuous Galerkin and spline based Vlasov solvers in four dimensions. *J. Comput. Phys.* **376**, 937–951 (2019)
19. Erath, C., Nair, R.D.: A conservative multi-tracer transport scheme for spectral-element spherical grids. *J. Comput. Phys.* **256**, 118–134 (2014)
20. Ewing, R.E., Wang, H.: Eulerian-Lagrangian localized adjoint methods for linear advection or advection-reaction equations and their convergence analysis. *Comput. Mech.* **12**(1/2), 97–121 (1993)
21. Filbet, F., Sonnendrücker, E.: Comparison of Eulerian Vlasov solvers. *Comput. Phys. Commun.* **150**(3), 247–266 (2003)
22. Filbet, F., Sonnendrücker, E., Bertrand, P.: Conservative numerical schemes for the Vlasov equation. *J. Comput. Phys.* **172**(1), 166–187 (2001)
23. Fischer, P., Mullen, J.: Filter-based stabilization of spectral element methods. *Comptes Rendus de l'Académie des Sci.-Ser. I-Math.* **332**(3), 265–270 (2001)
24. Frenod, E., Hirstoaga, S.A., Lutz, M., Sonnendrücker, E.: Long time behaviour of an exponential integrator for a Vlasov-Poisson system with strong magnetic field. *Commun. Comput. Phys.* **18**(2), 263–296 (2015)
25. Güçlü, Y., Christlieb, A.J., Hitchon, W.N.: Arbitrarily high order convected scheme solution of the Vlasov-Poisson system. *J. Comput. Phys.* **270**, 711–752 (2014)
26. Guo, W., Nair, R., Qiu, J.-M.: A conservative semi-Lagrangian discontinuous Galerkin scheme on the cubed-sphere. *Monthly Weather Rev.* **142**(1), 457–475 (2013)
27. Guo, Y., Xiong, T., Shi, Y.: A maximum-principle-satisfying high-order finite volume compact WENO scheme for scalar conservation laws with applications in incompressible flows. *J. Sci. Comput.* **65**(1), 83–109 (2015)
28. Hairer, E., Lubich, C., Wanner, G.: *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*, vol. 31. Springer Science & Business Media (2006)
29. Harris, L., Lauritzen, P., Mittal, R.: A flux-form version of the conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed sphere grid. *J. Comput. Phys.* **230**(4), 1215–1237 (2011)
30. Herrera, I., Ewing, R.E., Celia, M.A., Russell, T.F.: Eulerian-Lagrangian localized adjoint method: the theoretical framework. *Numer. Methods Partial Differ. Equ.* **9**(4), 431–457 (1993)
31. Heumann, H., Hiptmair, R., Li, K., Xu, J.: Fully discrete semi-Lagrangian methods for advection of differential forms. *BIT Numer. Math.* **52**(4), 981–1007 (2012)
32. Huang, C.-S., Arbogast, T., Hung, C.-H.: A semi-Lagrangian finite difference WENO scheme for scalar nonlinear conservation laws. *J. Comput. Phys.* **322**, 559–585 (2016)
33. Lauritzen, P., Nair, R., Ullrich, P.: A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid. *J. Comput. Phys.* **229**(5), 1401–1424 (2010)

34. Lee, D., Lowrie, R., Petersen, M., Ringler, T., Hecht, M.: A high order characteristic discontinuous Galerkin scheme for advection on unstructured meshes. *J. Comput. Phys.* **324**, 289–302 (2016)
35. LeVeque, R.: High-resolution conservative algorithms for advection in incompressible flow. *SIAM J. Numer. Anal.* **33**(2), 627–665 (1996)
36. Lin, S., Rood, R.: An explicit flux-form semi-Lagrangian shallow-water model on the sphere. *Q. J. R. Meteorol. Soc.* **123**(544), 2477–2498 (1997)
37. Liu, J.-G., Shu, C.-W.: A high-order discontinuous Galerkin method for 2D incompressible flows. *J. Comput. Phys.* **160**(2), 577–596 (2000)
38. Minion, M.L., Brown, D.L.: Performance of under-resolved two-dimensional incompressible flow simulations, ii. *J. Comput. Phys.* **138**(2), 734–765 (1997)
39. Qiu, J.-M.: High-order mass-conservative semi-Lagrangian methods for transport problems. In: *Handbook of Numerical Analysis*, vol. 17, pp. 353–382. Elsevier (2016)
40. Qiu, J.-M., Christlieb, A.: A conservative high order semi-Lagrangian WENO method for the Vlasov equation. *J. Comput. Phys.* **229**, 1130–1149 (2010)
41. Qiu, J.-M., Russo, G.: A high order multi-dimensional characteristic tracing strategy for the Vlasov-Poisson system. *J. Sci. Comput.* **71**(1), 414–434 (2017)
42. Qiu, J.-M., Shu, C.-W.: Conservative high order semi-Lagrangian finite difference WENO methods for advection in incompressible flow. *J. Comput. Phys.* **230**(4), 863–889 (2011)
43. Qiu, J.-M., Shu, C.-W.: Conservative semi-Lagrangian finite difference WENO formulations with applications to the Vlasov equation. *Commun. Comput. Phys.* **10**(4), 979–1000 (2011)
44. Qiu, J.-M., Shu, C.-W.: Positivity preserving semi-Lagrangian discontinuous Galerkin formulation: theoretical analysis and application to the Vlasov-Poisson system. *J. Comput. Phys.* **230**(23), 8386–8409 (2011)
45. Restelli, M., Bonaventura, L., Sacco, R.: A semi-Lagrangian discontinuous Galerkin method for scalar advection by incompressible flows. *J. Comput. Phys.* **216**(1), 195–215 (2006)
46. Rossmanith, J.A., Seal, D.C.: A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov-Poisson equations. *J. Comput. Phys.* **230**(16), 6203–6232 (2011)
47. Shoucri, M.M.: A two-level implicit scheme for the numerical solution of the linearized vorticity equation. *Int. J. Numer. Methods Eng.* **17**(10), 1525–1538 (1981)
48. Sonnendrücker, E., Roche, J., Bertrand, P., Ghizzo, A.: The semi-Lagrangian method for the numerical resolution of the Vlasov equation. *J. Comput. Phys.* **149**(2), 201–220 (1999)
49. Umeda, T.: A conservative and non-oscillatory scheme for Vlasov code simulations. *Earth Planets Space* **60**(7), 773–779 (2008)
50. Wang, H., Dahle, H., Ewing, R., Espedal, M., Sharpley, R., Man, S.: An ELLAM scheme for advection-diffusion equations in two dimensions. *SIAM J. Sci. Comput.* **20**(6), 2160–2194 (1999)
51. Xiong, T., Qiu, J.-M., Xu, Z., Christlieb, A.: High order maximum principle preserving semi-Lagrangian finite difference WENO schemes for the Vlasov equation. *J. Comput. Phys.* **273**, 618–639 (2014)
52. Xiong, T., Russo, G., Qiu, J.-M.: High order multi-dimensional characteristics tracing for the incompressible Euler equation and the guiding-center Vlasov equation. *J. Sci. Comput.* **77**(1), 263–282 (2018)
53. Xiu, D., Karniadakis, G.: A semi-Lagrangian high-order method for Navier-Stokes equations. *J. Comput. Phys.* **172**(2), 658–684 (2001)
54. Xu, C.: Stabilization methods for spectral element computations of incompressible flows. *J. Sci. Comput.* **27**(1/2/3), 495–505 (2006)
55. Yang, C., Filbet, F.: Conservative and non-conservative methods based on Hermite weighted essentially non-oscillatory reconstruction for Vlasov equations. *J. Comput. Phys.* **279**, 18–36 (2014)
56. Yang, Y., Cai, X., Qiu, J.-M.: Optimal convergence and superconvergence of semi-Lagrangian discontinuous Galerkin methods for linear convection equations in one space dimension. *Math. Comput.* **89**(325), 2113–2139 (2020)
57. Yoshida, H.: Construction of higher order symplectic integrators. *Phys. Lett. A* **150**(5/6/7), 262–268 (1990)
58. Zhang, X., Shu, C.-W.: On maximum-principle-satisfying high order schemes for scalar conservation laws. *J. Comput. Phys.* **229**, 3091–3120 (2010)
59. Zhong, X., Shu, C.-W.: A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods. *J. Comput. Phys.* **232**(1), 397–415 (2013)
60. Zhu, H., Qiu, J., Qiu, J.-M.: An  $h$ -adaptive RKDG method for the Vlasov-Poisson system. *J. Sci. Comput.* **69**(3), 1346–1365 (2016)
61. Zhu, H., Qiu, J., Qiu, J.-M.: An  $h$ -adaptive RKDG method for the two-dimensional incompressible Euler equations and the guiding center Vlasov model. *J. Sci. Comput.* **73**(2/3), 1316–1337 (2017)