



# A semi-Lagrangian discontinuous Galerkin (DG) – local DG method for solving convection-diffusion equations

Mingchang Ding<sup>a</sup>, Xiaofeng Cai<sup>a</sup>, Wei Guo<sup>b,1</sup>, Jing-Mei Qiu<sup>a,\*,2</sup>

<sup>a</sup> Department of Mathematical Sciences, University of Delaware, Newark, DE, 19716, United States of America

<sup>b</sup> Department of Mathematics and Statistics, Texas Tech University, Lubbock, TX, 79409, United States of America

## ARTICLE INFO

### Article history:

Received 12 July 2019

Received in revised form 22 January 2020

Accepted 26 January 2020

Available online 7 February 2020

### Keywords:

Convection-diffusion equation

Semi-Lagrangian

Discontinuous Galerkin (DG) method

Local DG method

Implicit Runge-Kutta method

Stability analysis

## ABSTRACT

In this paper, we propose an efficient high order semi-Lagrangian (SL) discontinuous Galerkin (DG) method for solving linear convection-diffusion equations. The method generalizes our previous work on developing the SLDG method for transport equations [5], making it capable of handling additional diffusion and source terms. Within the DG framework, the solution is evolved along the characteristics; while the diffusion term is discretized by the local DG (LDG) method and integrated along characteristics by implicit Runge-Kutta methods together with source terms. The proposed method is named the ‘SLDG-LDG’ method and enjoys many attractive features of the DG and SL methods. These include the uniformly high order accuracy (e.g. third order) in space and in time, compact, mass conservative, and stability under large time stepping size. An  $L^2$  stability analysis is provided when the method is coupled with the first order backward Euler discretization. Effectiveness of the method are demonstrated by a group of numerical tests in one and two dimensions.

© 2020 Elsevier Inc. All rights reserved.

## 1. Introduction

In this paper, we are concerned with solving the time dependent convection-diffusion problems in the form of

$$\begin{cases} u_t + \nabla_{\mathbf{x}} \cdot (\mathbf{a}(\mathbf{x}, t)u) = \epsilon \Delta u + g, & \mathbf{x} \in \Omega, \quad t > 0, \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega \end{cases} \quad (1.1)$$

with  $\epsilon \geq 0$ . For the scope of our current research, we assume the velocity field  $\mathbf{a}(\mathbf{x}, t)$  to be continuous with respect to  $\mathbf{x}$  and  $t$ .

A popular computational method for finding approximate solutions to transport dominant problems in the form (1.1) is the semi-Lagrangian (SL) method, which has a long history in computational fluid dynamics, e.g. for convection-diffusion problems [21,16], climate modeling [15,18,9], plasma simulations [17], as well as linear and Hamilton-Jacobi equations [10]. For transport dominant problems, the method is designed via tracking the characteristics forward or backward in time, thus

\* Corresponding author.

E-mail addresses: dmcvamos@udel.edu (M. Ding), xfcai@udel.edu (X. Cai), weimath.guo@ttu.edu (W. Guo), jingqiu@udel.edu (J.-M. Qiu).

<sup>1</sup> Research is supported by NSF grant NSF-DMS-1830838 (Program manager: Dr. Leland M. Jameson).

<sup>2</sup> Research of the first, second and last author is supported by NSF grant NSF-DMS-1522777 and NSF-DMS-1818924 (Program manager: Dr. Leland M. Jameson), Air Force Office of Scientific Research FA9550-18-1-0257 (Program manager: Dr. Fariba Fahroo).

avoiding the time step restriction, and can be coupled with various spatial discretization, such as the finite element method [16], the finite difference method with polynomial and spline interpolations [17], the spectral element method [11], the discontinuous Galerkin (DG) method [5]. In the presence of diffusion and source terms, usually time integration should be performed along characteristics, e.g. see [12] for the BGK model, and [7,2,21] for linear and nonlinear convection-diffusion models.

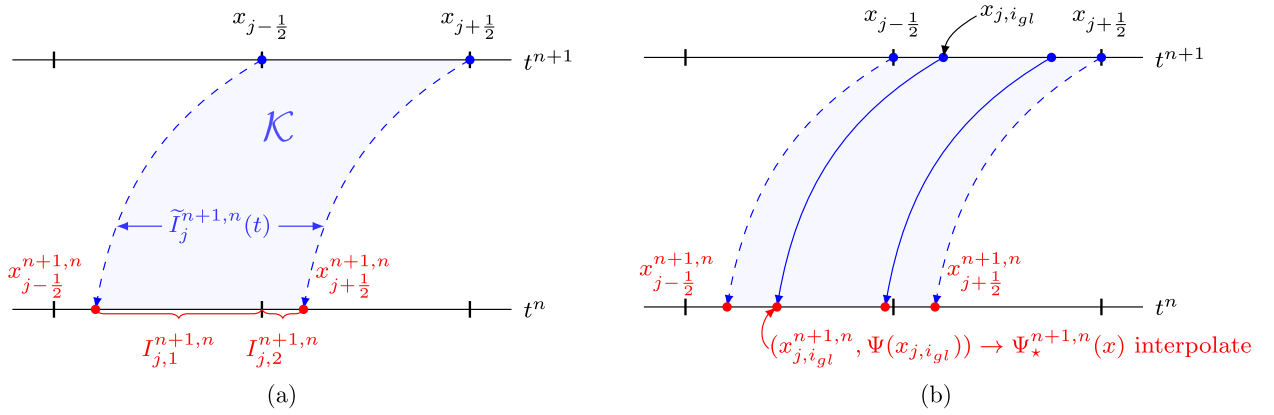
The objective of this paper is to develop an efficient high order SL method for (1.1) under the DG framework. The DG discretization approach is a class of finite element methods that use piecewise continuous approximations and enjoy many attractive computational advantages for transport dominant problems. In this paper, we propose to evolve the convection term by the SLDG method recently proposed in [5], and treat the diffusion term by the local DG (LDG) method coupled with a diagonally implicit (DI) Runge-Kutta (RK) method along dynamic characteristics elements. The proposed method is termed as the SLDG-LDG method. In the scheme formulation, we introduce the adjoint problem for the test function in the same spirit of ELLAM [16], and project the DG solution and LDG approximation to second derivative terms onto a set of time-dependent characteristics elements, based on the procedure developed in our earlier work [5]. There are a few key novelties of this work, compared with existing methods in the literature. First, thanks to the DG framework together with the backward characteristics tracing mechanism, our proposed scheme is inherently locally mass conservative. In particular, when compared with the SL finite difference framework in which a high order interpolation is employed, the DG finite element and the finite volume schemes are known to be a more natural framework for mass conservation; the authors in [3] propose an SLDG method with forward characteristics tracing. In their work the global mass conservation is enforced by an extra step of constrained optimization, i.e., a mass fixer, and hence the local mass conservation as well as the original order accuracy are not guaranteed. We note that the SLDG work [5] is an extension of the CSLAM [14] from the finite volume setting to the DG setting by introducing an adjoint problem for the test function; and it shares the same local mass conservation property with the CSLAM. Second, we inherit advantages of the DG in the SL framework. These include the schemes' ability to resolve solution structures and to evaluate the diffusion term by the LDG method. In the LDG method, by introducing auxiliary variables the second spatial derivative is rewritten into a system of first order equations, and proper choices of fluxes are made for numerical stability and accuracy. Third, unlike the method-of-lines approach in an Eulerian framework, the time integration has to be performed for the material derivative which is not necessarily aligned with the background grid. Hence, extra effort has to be made. Second order Crank-Nicolson and BDF methods have been proposed and used in the SL setting [21]; yet there is little existing work that employs higher than second order multi-stage RK method for the integration of non-convection terms. In this work, on each RK stage, we propose to update the solution and the diffusion term on the background elements; and then project them to the characteristics elements by the same SLDG algorithm in a purely convective setting [5]. The RK implementation can be done in a stage-by-stage manner as that of first order backward Euler method. As we use the SL method for transport and implicit RK method along characteristics for other terms, our scheme is highly accurate and unconditionally stable for linear problems. Last, our scheme formulation does not employ operator splitting and thus is free of splitting error.

Another class of very popular solvers for (1.1) is the Eulerian method, among which the most relevant high order methods related to this work is the Eulerian DG method. Typically, an implicit-explicit (IMEX) RK time discretization is used for time discretization of (1.1), i.e. the convection term is handled by explicit RK methods, while the diffusion term is discretized by an LDG [8] method in space along with an implicit RK method in time. From the stability analysis via the energy method in [19], there is a very strong result stating that "such IMEX LDG schemes are unconditionally stable for the linear problems in the sense that the time-step size is only required to be upper-bounded by a constant which depends on the ratio of the diffusion and the square of the advection coefficients and is independent of the spatial mesh-size  $h$ , even though the advection term is treated explicitly." We remark that, under the same setting, our scheme is unconditionally stable with no time step constraint for stability, when a first order backward Euler method is used. Extension of the theoretical analysis, when a higher order DIRK method is used for diffusion term, is subject to future work.

The rest of the paper is organized as follows. In Section 2, we introduce the proposed methodology for one-dimensional (1D) and two-dimensional (2D) problems; theoretically we prove the mass conservation and  $L^2$  stability when the method is coupled with the first-order backward Euler method. In Section 3, we present numerical results to demonstrate the effectiveness of our proposed approach with high order accuracy, and stability under large time stepping sizes. Finally, a conclusion is given in Section 4.

## 2. The SLDG-LDG method for convection-diffusion problems

In this paper, we focus on problems in one and two dimensions on rectangular domains with zero or periodic boundary conditions. Notice that our problem (1.1) is in the conservative form, for which local mass conservation is desired at the discrete level for the numerical scheme. Below, we formulate the proposed scheme for 1D problems in Section 2.1 by first introducing the spatial discretization and the adjoint problem for the test function; then we introduce the proposed treatment of the diffusion and source terms with DIRK methods along characteristic elements. The extensions to 2D problems are then discussed briefly in Section 2.2.



**Fig. 1.** Schematic illustration of the SLDG-LDG formulation in 1D. Left: Integration region  $\mathcal{K}$ , dynamic interval  $\tilde{I}_j^{n+1,n}(t)$  and upstream interval  $I_j^{n+1,n} = I_{j,1}^{n+1,n} \cup I_{j,2}^{n+1,n}$ . Right: Interpolation of  $\psi^{n+1,n}$ .

### 2.1. Scheme formulation: 1D case

To introduce the algorithm, we start from the 1D case of (1.1):

$$u_t + (a(x, t)u)_x = \epsilon u_{xx} + g. \quad (2.1)$$

**I. Spatial discretization: DG solution and test function spaces.** We discretize the 1D domain  $[x_a, x_b]$  into  $N$  elements:  $x_a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \dots < x_{N+\frac{1}{2}} = x_b$ , with  $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$  denoting an element of length  $\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$  for  $j = 1, 2, \dots, N$ .  $\Delta t = t^{n+1} - t^n$  represents the time discretization step. In the framework of the DG method, we let numerical solutions and test functions belong to the finite dimensional piecewise approximation space

$$V_h^k = \{v_h : v_h|_{I_j} \in P^k(I_j), j = 1, 2, \dots, N\}, \quad (2.2)$$

where  $P^k(I_j)$  denotes the set of polynomials of degree at most  $k$  over  $I_j$ .

**II. Adjoint problem.** To formulate the SLDG-LDG scheme, we follow a similar idea in [13,5] by considering the following adjoint problem for the test function  $\psi(x, t)$  that satisfies

$$\psi_t + a(x, t)\psi_x = 0, \quad t \in [\tau_1, \tau_2] \quad (2.3)$$

with

$$\psi(x, \tau_2) = \Psi(x) \in V_h^k. \quad (2.4)$$

In other words,  $\psi$  satisfies a final-value problem with function values specified at  $\tau_2$ . For a pure convection problem [5], we have  $[\tau_1, \tau_2] = [t^n, t^{n+1}]$ ; while  $\tau_1$  and  $\tau_2$  could also correspond to different time stages in an implicit RK method when discretizing the diffusion and source terms along characteristics. Next, we make the following observations for the test function  $\psi(x, t)$ :

- (i) While the original problem (1.1) is in the conservative form, an adjoint problem for the test function is in the advective form (2.3). Along characteristics curves governed by

$$\frac{d\tilde{x}(t)}{dt} = a(\tilde{x}(t), t), \quad (2.5)$$

$\psi(\tilde{x}(t), t)$  stays constant. Hence  $\psi(x, t)$ ,  $\forall x \in [x_a, x_b]$ ,  $t \in [\tau_1, \tau_2]$  can be obtained by tracking characteristics based on (2.5).

- (ii) The test function satisfies a final value problem (2.4). In general,  $\psi(x, t)$  with  $t \in [\tau_1, \tau_2]$ , is not necessarily a polynomial. Yet, it can be approximated by polynomials with high order accuracy as presented in the algorithm flowchart Step 1.1 below.

**III. Time dependent characteristics interval, see Fig. 1(a).** Let

$$\tilde{I}_j^{n+1,n}(t) = [\tilde{x}_{j-\frac{1}{2}}(t), \tilde{x}_{j+\frac{1}{2}}(t)], \quad t \in [t^n, t^{n+1}],$$

be the dynamic interval bounded by characteristics curves  $\tilde{x}_{j-\frac{1}{2}}(t)$  and  $\tilde{x}_{j+\frac{1}{2}}(t)$  emanating from cell boundaries of  $I_j$  at  $t^{n+1}$ , where  $\tilde{x}_{j\pm\frac{1}{2}}(t)$  satisfy the final value problems

$$\frac{d\tilde{x}_{j\pm\frac{1}{2}}(t)}{dt} = a(\tilde{x}_{j\pm\frac{1}{2}}(t), t), \quad \tilde{x}_{j\pm\frac{1}{2}}(t^{n+1}) = x_{j\pm\frac{1}{2}}. \quad (2.6)$$

Let  $[\tilde{x}_{j-\frac{1}{2}}(t^n), \tilde{x}_{j+\frac{1}{2}}(t^n)] = [x_{j-\frac{1}{2}}^{n+1,n}, x_{j+\frac{1}{2}}^{n+1,n}]$ , which will be referred to as the “upstream cell” at  $t^n$  later. Here the superscripts  $n+1, n$  refer to the interval from  $t^{n+1}$ , being tracked backward in time to  $t^n$ . See Fig. 1(a) for illustration of  $\tilde{x}_{j-\frac{1}{2}}(t)$ ,  $\tilde{x}_{j+\frac{1}{2}}(t)$  and  $[x_{j-\frac{1}{2}}^{n+1,n}, x_{j+\frac{1}{2}}^{n+1,n}]$ .

**IV. SLDG-LDG scheme formulation and discretization.** Let  $u^n$  be the numerical solution at time  $t^n$ , the SLDG-LDG scheme is formulated based on Proposition 2.1 presented below.

**Proposition 2.1.** Consider the 1D problem (2.1) and the adjoint problem (2.3) for the test function  $\psi$ , then the following identity holds

$$\int_{I_j} u^{n+1} \Psi \, dx - \underbrace{\int_{I_j^{n+1,n}} u^n \psi^{n+1,n} \, dx}_{\text{Term I}} = \underbrace{\int_{t^n}^{t^{n+1}} \int_{\tilde{I}_j^{n+1,n}(t)} [\epsilon u_{xx} + g] \psi \, dx \, dt}_{\text{Term II}}, \quad \Psi \in P^k(I_j) \quad (2.7)$$

where  $\psi^{n+1,n}$  is the solution to the adjoint problem (2.3) with  $\psi(x, t^{n+1}) = \Psi$  at  $t^n$ .

**Proof.**  $\int_{\mathcal{K}} (2.1) \cdot \psi + (2.3) \cdot u \, dx \, dt$  leads to

$$\int_{\mathcal{K}} ([u_t + (a(x, t)u)_x] \psi + [\psi_t + a(x, t)\psi_x] u) \, dx \, dt = \int_{\mathcal{K}} [\epsilon u_{xx} + g] \psi \, dx \, dt, \quad (2.8)$$

where the domain  $\mathcal{K}$  (see Fig. 1(a)) is bounded by  $I_j$ ,  $I_j^{n+1,n}$  from above and below, and characteristic trajectories  $\tilde{x}_{j-\frac{1}{2}}(t)$  and  $\tilde{x}_{j+\frac{1}{2}}(t)$  from left and right, respectively. Rearranging terms in Eq. (2.8) gives

$$\int_{\mathcal{K}} [u\psi]_t + [a(x, t)u\psi]_x \, dx \, dt = \int_{\mathcal{K}} [\epsilon u_{xx} + g] \psi \, dx \, dt. \quad (2.9)$$

Applying the divergence theorem to the left-hand side (LHS) of Eq. (2.9) and due to the cancellation of the integrals along the characteristic curves, we prove (2.7).  $\square$

To update  $u^{n+1} \in V_h^k$ , one has to evaluate Term I and Term II in Eq. (2.7) by letting the test function  $\Psi$  go through all the basis functions in  $V_h^k$ . In particular, the proposed SLDG-LDG scheme consists of the following two steps: Step 1.1 and 1.2.

**Step 1.1: Evaluation of Term I of (2.7) as in the SLDG method [5].** To evaluate  $\int_{I_j^{n+1,n}} u^n \psi^{n+1,n} \, dx$ , we propose the procedures below.

**Step 1.1a: Reconstruct test function  $\psi^{n+1,n}$  through interpolation.** Choose  $k+1$  interpolation points  $\{x_{j,i_{gl}}\}_{i_{gl}=1}^{k+1}$  such as the Gauss-Lobatto (GL) points over  $I_j$  at  $t^{n+1}$  and locate the characteristic feet  $\{x_{j,i_{gl}}^{n+1,n}\}_{i_{gl}=1}^{k+1}$  at  $t^n$  by solving

$$\begin{cases} \frac{d\tilde{x}(t)}{dt} = a(\tilde{x}(t), t) \\ \tilde{x}(t^{n+1}) = x_{j,i_{gl}} \end{cases} \quad (2.10)$$

with high order numerical integrators. In our implementation, a fourth order RK method is applied. Thus,  $\psi^{n+1,n}(x_{j,i_{gl}}^{n+1,n}) = \Psi(x_{j,i_{gl}})$ , for  $i_{gl} = 1, \dots, k+1$ . We then construct a degree  $k$  polynomial  $\Psi_{\star}^{n+1,n}(x)$  interpolating the  $\psi^{n+1,n}$  function at characteristic feet  $\{x_{j,i_{gl}}^{n+1,n}\}_{i_{gl}=1}^{k+1}$  located over upstream interval  $I_j^{n+1,n}$  in Step 1.1a, see Fig. 1(b).

**Step 1.1b: Integrating Term I by summation over sub-intervals.** From Fig. 1(a), we can see that there are two inter-sections  $I_{j,1}^{n+1,n} = [x_{j-\frac{1}{2}}^{n+1,n}, x_{j-\frac{1}{2}}]$  and  $I_{j,2}^{n+1,n} = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}^{n+1,n}]$  between  $I_j^{n+1,n}$  and the background element  $I_j$ . In general,  $I_j^{n+1,n} = \bigcup_l I_{j,l}^{n+1,n}$  where  $l$  is the index for sub-intervals. Term I is approximated by

$$\int_{I_j^{n+1,n}} u^n \psi^{n+1,n} dx \approx \sum_l \int_{I_{j,l}^{n+1,n}} u^n \Psi_\star^{n+1,n} dx. \quad (2.11)$$

On each of these subintervals  $I_{j,l}^{n+1,n}$ ,  $u^n \Psi_\star^{n+1,n}$  is continuous and its integration can be approximated by quadrature rules. Notice that  $u^n$  is discontinuous across cell boundaries.

**Step 1.2: Evaluation of Term II of (2.7) along characteristics intervals.** There are two technical components involved in this step: one is an LDG approximation to the second order derivative term  $u_{xx}$  together with a proper evaluation of  $\int_{\tilde{I}_j^{n+1,n}(t)} [\epsilon u_{xx} + g] \psi dx$ ; the other is the high order temporal discretization by a RK method for

$$\frac{d}{dt} \int_{\tilde{I}_j^{n+1,n}(t)} u \psi dx = \int_{\tilde{I}_j^{n+1,n}(t)} [\epsilon u_{xx} + g] \psi dx, \quad (2.12)$$

which is the time differential form of Eq. (2.7). We will first discuss the evaluation of Term II coupled with a simple backward Euler time discretization. Then we will extend the idea to high order time integration by employing diagonally implicit RK (DIRK) methods. The diagonally implicit property allows one to solve a linear system for the current RK stage only, greatly reducing computational complexity and cost.

**Step 1.2a: LDG approximation of  $u_{xx}$  [8].** We use the LDG formulation to seek  $p \in V_h^k$  approximating  $u_{xx}$ . In particular,  $p = u_{xx}$  can be rewritten as a first order system

$$\begin{cases} p = q_x, \\ q = u_x. \end{cases} \quad (2.13)$$

Then, we seek  $p, q \in V_h^k$  such that, for all test functions  $v, w \in P^k(I_j)$ ,

$$(p, v)_{I_j} = \hat{q}_{j+\frac{1}{2}} v_{j+\frac{1}{2}}^- - \hat{q}_{j-\frac{1}{2}} v_{j-\frac{1}{2}}^+ - (q, v_x)_{I_j}, \quad (2.14a)$$

$$(q, w)_{I_j} = \hat{u}_{j+\frac{1}{2}} w_{j+\frac{1}{2}}^- - \hat{u}_{j-\frac{1}{2}} w_{j-\frac{1}{2}}^+ - (u, w_x)_{I_j}, \quad (2.14b)$$

where  $(\cdot, \cdot)_{I_j}$  stands for the  $L^2$  inner product on interval  $I_j$ , and  $\hat{\cdot}$  denotes the numerical fluxes defined at the cell interfaces, which are taken as the alternating fluxes for stability consideration

$$\hat{q} = q^-, \quad \hat{u} = u^+; \quad \text{or} \quad \hat{q} = q^+, \quad \hat{u} = u^-. \quad (2.15)$$

Notice that  $q$  can be solved explicitly in terms of  $u$  from (2.14b); and also  $p$  can be solved explicitly from  $q$  from (2.14a). In short,  $p = u_{xx}$  can be computed locally by using  $u$  from three nearby elements, namely  $I_{j-1}$ ,  $I_j$  and  $I_{j+1}$ .

**Step 1.2b. DIRK methods for accurate evaluations of the time integral.**

We start from a first order backward Euler time discretization of (2.12):

$$(u^{n+1}, \Psi)_{I_j} - (u^n, \psi^{n+1,n})_{I_j^{n+1,n}} = \Delta t (\epsilon u_{xx}^{n+1} + g^{n+1}, \Psi)_{I_j}. \quad (2.16)$$

After rearranging the terms in Eq. (2.16), we obtain

$$(u^{n+1}, \Psi)_{I_j} - \epsilon \Delta t (u_{xx}^{n+1}, \Psi)_{I_j} = (u^n, \psi^{n+1,n})_{I_j^{n+1,n}} + \Delta t (g^{n+1}, \Psi)_{I_j}. \quad (2.17)$$

For notational simplicity of the presentation, above we let  $(\epsilon u_{xx}, \Psi)_{I_j}$  represent the LDG discretization of the diffusion term, without writing out all the flux and volume integral terms from integration-by-part in an LDG formulation.

With the test function  $\Psi$  going through all basis functions in  $V_h^k$ , we can formulate a linear system for degrees of freedom (i.e., the coefficients of the basis) of  $\mathbf{u}^{n+1}$  as

$$B_1 \mathbf{u}^{n+1} = \mathbf{f}_1, \quad (2.18)$$

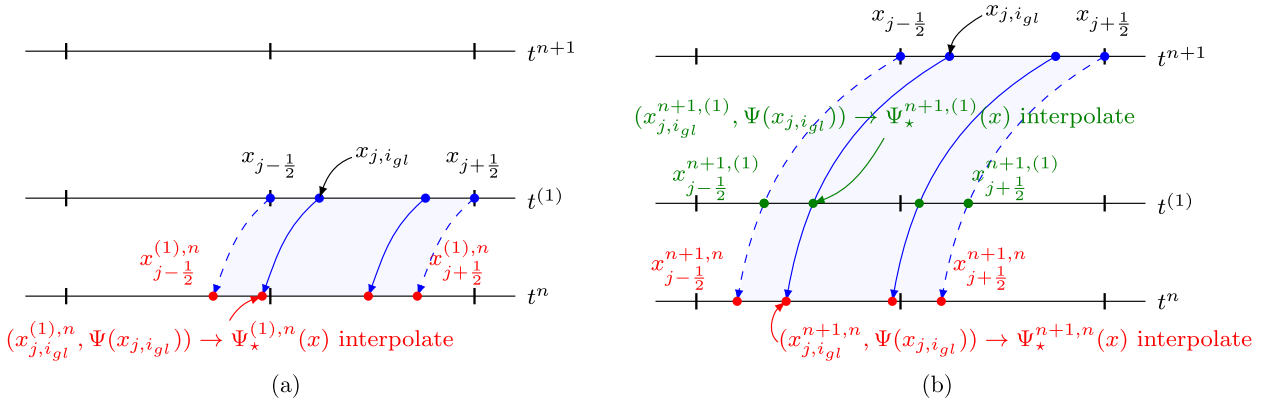
which can be solved by an iterative method, e.g. GMRES. Here the matrix  $B_1 = I - \epsilon \Delta t D_\Delta$ , where  $D_\Delta$  comes from an LDG discretization of  $u_{xx}$ ; and  $\mathbf{f}_1$  can be obtained from evaluating right-hand side (RHS) terms of (2.17). The details in constructing the sparse matrix  $B_1$  are provided in Appendix A.

To attain higher order accuracy in time, we propose to employ high order DIRK methods. Here, we demonstrate the scheme with an L-stable, two-stage, second-order DIRK method, termed as DIRK2 [1] (as in Table 1) that involves two stages:  $t^{(1)} = t^n + \nu \Delta t$  and  $t^{(2)} = t^{n+1}$ .

**Table 1**  
DIRK2.

$v$	$v$	0
1	$1-v$	$v$
	$1-v$	$v$

$$v = 1 - \sqrt{2}/2.$$

**Fig. 2.** Schematic illustration of 1D SLDG-LDG formulation coupled with DIRK2. Left: First time stage  $t^{(1)}$ . Right: Second time stage  $t^{n+1}$ .

For the convenience of our presentations for DIRK discretization along characteristics, we introduce the following notations

$$\psi^{\tau_2, \tau}(x), \quad I_j^{\tau_2, \tau}. \quad (2.19)$$

- (i)  $\psi^{\tau_2, \tau}(x)$  denotes the solution  $\psi(x, \tau)$  satisfying the final value problem (2.4). Here  $\tau_2$  and  $\tau$  may refer to intermediate RK stages in a DIRK discretization. Assuming DIRK has  $s$  stage  $t^n < t^{(1)} < \dots < t^{(s)} = t^{n+1}$ ,  $\psi^{t^{(ii)}, t^{(jj)}}(x)$  with  $1 \leq jj \leq ii \leq s$ , denotes the function  $\psi(x)$  at  $t = t^{(ii)}$  satisfying the adjoint problem (2.4) with the final value  $\psi(x, t^{(ii)}) = \Psi(x) \in V_h^k$ . For notational simplicity, we let  $\psi^{(ii), (jj)} \doteq \psi^{t^{(ii)}, t^{(jj)}}(x)$ .
- (ii)  $I_j^{\tau_2, \tau} = [x_{j-\frac{1}{2}}^{\tau_2, \tau}, x_{j+\frac{1}{2}}^{\tau_2, \tau}]$  with  $x_{j\pm\frac{1}{2}}^{\tau_2, \tau}$  being the solution to Eq. (2.6) at time  $\tau$  with  $\tilde{x}_{j\pm\frac{1}{2}}(\tau_2) = x_{j\pm\frac{1}{2}}$ . For example,  $I_j^{t^{(ii)}, t^{(jj)}} \doteq [\tilde{x}_{j-\frac{1}{2}}(t^{(jj)}), \tilde{x}_{j+\frac{1}{2}}(t^{(jj)})]$  with  $\tilde{x}_{j\pm\frac{1}{2}}(t)$  satisfying (2.6) and the final value  $\tilde{x}_{j\pm\frac{1}{2}}(t^{(ii)}) = x_{j\pm\frac{1}{2}}$ , respectively. For simplicity, we let  $I_j^{(ii), (jj)} \doteq I_j^{t^{(ii)}, t^{(jj)}}$ .

Following the above notations, the proposed SLDG-LDG scheme when coupled with a DIRK2 method (see Table 1) along characteristics curves can be implemented as below.

- (i) In the **first time stage**  $\tau_2 = t^{(1)}$ , as shown in Fig. 2(a), for each Eulerian background cell  $I_j$ , we solve the numerical solution  $u^{(1)} \in V_h^k$  at intermediate stage  $t^{(1)}$  from the following formulation

$$(u^{(1)}, \Psi)_{I_j} - (u^n, \psi^{(1), n})_{I_j^{(1), n}} = \Delta t \cdot v \left( \epsilon u_{xx}^{(1)} + g^{(1)}, \Psi \right)_{I_j}. \quad (2.20)$$

Note that the formulation is equivalent to applying a first order backward Euler method with  $v\Delta t$ . Implementation-wise, (2.20) can be written as  $B_2 \mathbf{u}^{(1)} = \mathbf{f}_2$ , where  $B_2$  and  $\mathbf{f}_2$  can be collected in a similar fashion as those for matrix  $B_1$  and vector  $\mathbf{f}_1$  in Eq. (2.18).

- (ii) In the **second time stage**  $\tau_2 = t^{n+1}$ , as shown in Fig. 2(b), we have the following formulation

$$(u^{n+1}, \Psi)_{I_j} - (u^n, \psi^{n+1, n})_{I_j^{n+1, n}} = \Delta t \left[ (1-v) \cdot \left( \epsilon u_{xx}^{(1)} + g^{(1)}, \psi^{n+1, (1)} \right)_{I_j^{n+1, (1)}} + v \cdot \left( \epsilon u_{xx}^{n+1} + g^{n+1}, \Psi \right)_{I_j} \right]. \quad (2.21)$$

Notice that  $\psi^{n+1, n}, \psi^{n+1, (1)}$  are in general not polynomials, yet can be well approximated by polynomials as in Step 1.1b. Reorganizing terms in (2.21) gives

$$(u^{n+1}, \Psi)_{I_j} - \Delta t \cdot v \epsilon (u_{xx}^{n+1}, \Psi)_{I_j} = (u^n, \psi^{n+1, n})_{I_j^{n+1, n}} + \Delta t \cdot (1-v) \epsilon (u_{xx}^{(1)}, \psi^{n+1, (1)})_{I_j^{n+1, (1)}} + \Delta t \cdot (1-v) \left( g^{(1)}, \psi^{n+1, (1)} \right)_{I_j^{n+1, (1)}} + \Delta t \cdot v \left( g^{n+1}, \Psi \right)_{I_j}. \quad (2.22)$$

Notice that the first term on RHS of (2.22) can be evaluated as in Step 1.1; the second term on RHS of (2.22) can be evaluated by first computing  $u_{xx}^{(1)}$  in an LDG fashion with  $u^{(1)}$  given from the first stage of RK computation, and then applying the same procedure as in Step 1.1 to evaluate  $(u_{xx}^{(1)}, \psi^{n+1,(1)})_{I_j^{n+1,(1)}}$ ; the latter two terms involving  $g$  can be directly evaluated by quadrature rules. Implementation-wise, (2.22) can be written as  $B_3 \mathbf{u}^{n+1} = \mathbf{f}_3$  with  $B_3$  the same matrix as  $B_2$  in the first time stage.

**Remark 2.2.** The above procedure is for a two-stage second-order DIRK discretization of diffusion and source terms. Such a procedure can be generalized to any DIRK methods. For some high order DIRK discretization methods we use for the numerical experiment, the associated Butcher tableaus are provided in Appendix A including the L-stable, three-stage, third-order DIRK method in Table 7 [6], the L-stable, five-stage, fourth-order method in Table 8 [20]. Notice that, we use the SLDG method for the convection term and an implicit discretization for the diffusion and source terms; thus the time stepping size allowed could be much larger than that of an explicit Eulerian RKDG method.

**Remark 2.3.** All DIRK time discretization methods we employ in the paper have the property that  $a_{ii} \neq 0$ ,  $\forall i = 1, \dots, s$  and the method are stiffly accurate; these properties are important for numerical stability.

## 2.2. Scheme formulation: 2D case

In this subsection, we generalize the above 1D SLDG-LDG scheme for solving the following 2D problem

$$u_t + (a(x, y, t)u)_x + (b(x, y, t)u)_y = \epsilon \Delta u + g. \quad (2.23)$$

We begin with a partition of the 2D domain as  $\Omega = \{E_j\}_{j=1}^J$ . The numerical solutions and test functions belong to the finite dimensional piecewise approximation space

$$V_h^k = \{v_h : v_h|_{E_j} \in P^k(E_j), j = 1, 2, \dots, J\} \quad (2.24)$$

where  $P^k(E_j)$  denotes the set of polynomials of degree at most  $k$  over each element  $E_j$ .

Similar to the 1D case and the strategy in [5], we consider the adjoint problem for the test function  $\psi = \psi(x, y, t)$  satisfying

$$\psi_t + a(x, y, t)\psi_x + b(x, y, t)\psi_y = 0, \quad t \in [\tau_1, \tau_2], \quad (2.25)$$

with  $\psi(x, y, \tau_2) = \Psi(x, y) \in V_h^k$ . A similar observation as in the 1D case is that the solution to (2.25) stays constant along the characteristic curves governed by

$$\frac{d\tilde{x}(t)}{dt} = a(\tilde{x}(t), \tilde{y}(t), t), \quad \frac{d\tilde{y}(t)}{dt} = b(\tilde{x}(t), \tilde{y}(t), t).$$

Let  $\tilde{E}_j(t)$  be the dynamic moving cell bounded by characteristics curves emanating from the edges of Eulerian cell  $E_j$  at  $t^{n+1}$  and  $E_j^{n+1,n}$  be the upstream cell as  $\tilde{E}_j(t = t^n)$ , see Fig. 3(a). A 2D generalization of Proposition 2.1 is established in the following.

**Proposition 2.4.** Consider the 2D problem (2.23) and the adjoint problem (2.25) for the test function  $\psi$ , then the following identity holds

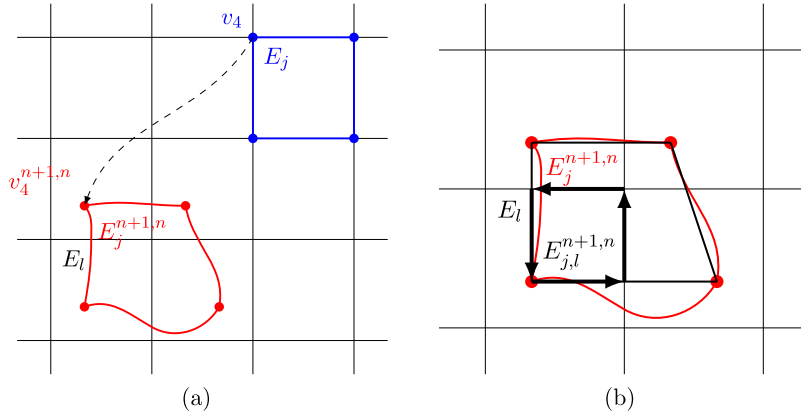
$$\underbrace{\iint_{E_j} u^{n+1} \Psi \, dx \, dy - \iint_{E_j^{n+1,n}} u^n \psi^{n+1,n} \, dx \, dy}_{\text{Term I}} = \underbrace{\int_{t^n}^{t^{n+1}} \iint_{\tilde{E}_j(t)} [\epsilon \Delta u + g] \psi \, dx \, dy \, dt}_{\text{Term II}}, \quad (2.26)$$

where  $\psi^{n+1,n}$  is the solution to the adjoint problem (2.25) at  $t^n$  with  $\psi(x, y, t^{n+1}) = \Psi \in V_h^k$ .

Similar to the 1D case, the update of  $u^{n+1} \in V_h^k$  depends on proper evaluations of Term I and Term II of Eq. (2.26). We again refer to [5] for detailed procedures of evaluating Term I and only summarize main steps below. The computation of Term II consists of two parts: the first part is approximating  $\Delta u$  by using 2D LDG spatial discretization, and the second part is high order time integration over the dynamic moving cell  $\tilde{E}_j(t)$  along the characteristics. These two parts share the same spirit with Step 1.2 in Subsection 2.1. Below we outline the main procedures for a 2D problem.

**Step 2.1: Evaluation of Term I of (2.26).** It's worth noting that when the velocity field is space and time dependent, the upstream cell  $E_j^{n+1,n}$  might not be of quadrilateral shape. When they are approximated by a quadrilateral or a quadratic-curved





**Fig. 3.** Schematic illustration of the SLDG formulation with  $P^1$  polynomial spaces in 2D. Left: upstream cell  $E_j^{n+1,n}$ . Right: Quadrilateral upstream cells.

(QC) quadrilateral, second or third-order spatial accuracy can be achieved, respectively. Here, as an example, we discuss the formulation with  $P^1$  polynomial spaces.

**Step 2.1a: Characteristic tracing.** Locate four vertices  $\{v_q^{n+1,n}\}_{q=1}^4$  of upstream cell  $E_j^{n+1,n}$  at  $t^n$  by solving the final value problem

$$\begin{cases} \frac{d\tilde{x}(t)}{dt} = a(\tilde{x}(t), \tilde{y}(t), t), & \tilde{x}(t^{n+1}) = x_{v_q}, \\ \frac{d\tilde{y}(t)}{dt} = b(\tilde{x}(t), \tilde{y}(t), t), & \tilde{y}(t^{n+1}) = y_{v_q} \end{cases} \quad (2.27)$$

by high order numerical integrators the same way as in the 1D case, and  $\{v_q\}_{q=1}^4$  with coordinates  $(x_{v_q}, y_{v_q})$  are the four vertices of  $E_j$ .

**Step 2.1b: Reconstruction of the test function  $\psi^{n+1,n}$  and decomposition of Term I.** It is known that  $\psi(x, y, t)$  with adjoint problem (2.25) stays constant along characteristics,

$$\psi^{n+1,n}(x(v_q^{n+1,n}), y(v_q^{n+1,n})) = \Psi(x(v_q), y(v_q)), \quad q = 1, \dots, 4. \quad (2.28)$$

We can reconstruct  $P^k$  ( $k = 1$ ) polynomial  $\Psi_\star^{n+1,n}(x)$  to approximate  $\psi^{n+1,n}$  by a least-square strategy. Let  $E_{j,l}^{n+1,n}$  be intersections between the upstream cell  $E_j^{n+1,n}$  and the background cell  $E_l$ , see in Fig. 3(b). In general,  $E_j^{n+1,n} = \bigcup_{l \in \epsilon_j^{n+1,n}} E_{j,l}^{n+1,n}$

where  $\epsilon_j^{n+1,n} \doteq \{l | E_{j,l}^{n+1,n} \neq \emptyset \text{ where } E_{j,l}^{n+1,n} \doteq E_j^{n+1,n} \cap E_l\}$ , and then Term I in (2.26) can be approximated with

$$\iint_{E_j^{n+1,n}} u^n \psi^{n+1,n} dx dy \approx \sum_{l \in \epsilon_j^{n+1,n}} \iint_{E_{j,l}^{n+1,n}} u^n \Psi_\star^{n+1,n}(x, y) dx dy. \quad (2.29)$$

**Step 2.1c: Line integral evaluation.** To evaluate the area integral  $\iint_{E_{j,l}^{n+1,n}} u^n \Psi_\star^{n+1,n}(x, y) dx dy$ , we can introduce two auxiliary functions  $P(x, y)$  and  $Q(x, y)$  satisfying

$$-\frac{\partial P}{\partial y} + \frac{\partial Q}{\partial x} = u^n \Psi_\star^{n+1,n}(x, y).$$

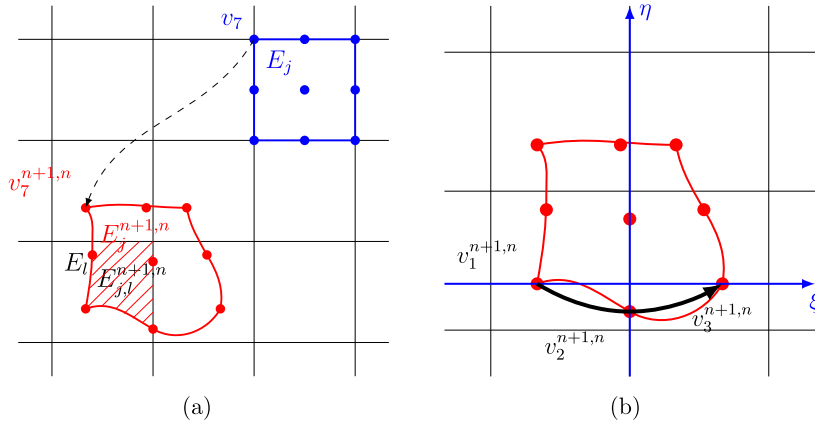
Due to the Green's theorem, the area integral can be converted into the line integrals as

$$\iint_{E_{j,l}^{n+1,n}} u^n \Psi_\star^{n+1,n}(x, y) dx dy = \oint_{\partial E_{j,l}^{n+1,n}} P dx + Q dy, \quad (2.30)$$

where quadrature rules can be directly applied along  $\partial E_{j,l}^{n+1,n}$ . This evaluation procedure is the same as in [14].

**Remark 2.5.** Applying quadrilateral approximation to  $P^2$  polynomial spaces will restrict us with the second-order accuracy in a general setting. This motivates us to use QC quadrilateral approximation to the upstream cells for higher order accuracy. There are two additional key steps. First, locate nine upstream points  $\{v_q^{n+1,n}\}_{q=1}^9$  belonging to the upstream cell  $E_j^{n+1,n}$ , see Fig. 4(a), by solving (2.27) with final values  $\{v_q\}_{q=1}^9$  (nine uniformly distributed points at  $E_j$ ). Second, approximate each





**Fig. 4.** Schematic illustration of the SLDG formulation with  $P^2$  polynomial spaces in 2D. Left: upstream cell  $E_j^{n+1,n}$  and subregion  $E_{j,l}^{n+1,n}$ . Right: QC quadrilateral upstream cells.

side of the upstream cell with a quadratic curve by a parameterization and evaluation of the area integral through the line integrals. For more details, we refer to [5].

**Step 2.2: Evaluation of Term II.** Evaluation of Term II in Eq. (2.26) can be realized by similar steps as in Step 1.2. First, the LDG approximation to  $u_{xx}$  in Step 1.2a can be directly generalized to evaluate  $\Delta u$  in the 2D setting.  $p = \Delta u$  is rewritten into a system of first order equations

$$p = q_x + h_y, \quad q = u_x, \quad h = u_y. \quad (2.31)$$

Weak formulations of (2.31) can be discretized by an LDG method as in (2.14) to compute  $p = \Delta u$  from five nearby elements. High order time discretizations of Term II can be fulfilled with DIRK methods on  $\tilde{E}_j(t)$  in the same fashion as Step 1.2b, by applying DIRK methods to the time differential form of the scheme (2.26)

$$\frac{d}{dt} \iint_{\tilde{E}_j(t)} u \psi \, dx \, dy = \iint_{\tilde{E}_j(t)} [\epsilon \Delta u + g] \psi \, dx \, dy. \quad (2.32)$$

We summarize the flowchart below in **Algorithm 1**, assuming the DIRK method has  $s$ -stages with the Butcher tableau

$$\begin{array}{c|c} \mathbf{c} & A \\ \hline & \mathbf{b}^T \end{array},$$

where  $A = (a_{ii,jj}) \in \mathbb{R}^{s \times s}$ ,  $\mathbf{b} \in \mathbb{R}^s$ , and  $\mathbf{c} \in \mathbb{R}^s$ . In this paper, we assume that the DIRK method is stiffly accurate, i.e. the last row of the  $A$  matrix is the same as the  $\mathbf{b}^T$  vector. Below we adopt a similar set of notations for the  $E_j^{\tau_2, \tau}$  and  $\psi^{\tau_2, \tau}(x, y)$  as in the 1D case, see (2.19). In particular,  $E_j^{\tau_2, \tau}$  is the upstream cell at time  $\tau$ , traced from the Eulerian cell  $E_j$  at time  $\tau_2$ ; and  $\psi^{\tau_2, \tau}(x, y)$  is the approximation to the test function  $\psi$  satisfying the adjoint problem (2.25) with final value  $\psi(x, y, \tau_2) = \Psi(x, y) \in V_h^k$ .

(i) In each DIRK stage  $t^{(ii)}$ ,  $1 \leq ii \leq s$ , we have

$$(u^{(ii)}, \Psi)_{E_j} = (u^n, \psi^{(ii),n})_{E_j^{(ii),n}} + \sum_{jj=1}^{ii} a_{ii,jj} \Delta t \left( \epsilon \Delta u^{(jj)} + g^{(jj)}, \psi^{(ii),(jj)} \right)_{E_j^{(ii),(jj)}}.$$

Rearranging the terms gives

$$\begin{aligned} (u^{(ii)}, \Psi)_{E_j} - a_{ii,ii} \Delta t \epsilon (\Delta u^{(ii)}, \Psi)_{E_j} \\ = (u^n, \psi^{(ii),n})_{E_j^{(ii),n}} + \sum_{jj=1}^{ii-1} a_{ii,jj} \Delta t \left( \epsilon \Delta u^{(jj)} + g^{(jj)}, \psi^{(ii),(jj)} \right)_{E_j^{(ii),(jj)}} + a_{ii,ii} \Delta t \left( g^{(ii)}, \Psi \right)_{E_j}, \end{aligned} \quad (2.33)$$

which is a generalization from the scheme for 1D problems (2.17) and (2.21).

(ii) Since the DIRK method we used is stiffly accurate,  $u^{n+1} = u^{(s)}$ .

**Algorithm 1** The SLDG-LDG scheme coupled with DIRK methods.

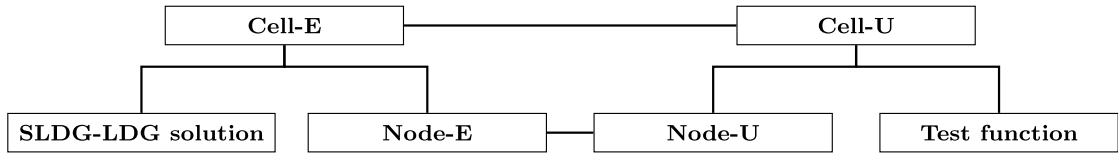
---

```

for  $ii = 1 \rightarrow s$  do
  for  $jj = 1 \rightarrow ii$  do
    Compute the RHS of Eq. (2.33) by performing the following:
    1. Find upstream cell  $E_j^{(ii),(jj)}$  via tracing characteristic backwards by  $(c_{ii} - c_{jj})\Delta t$ 
    as in Step 2.1a.
    2. Approximate test function  $\psi^{(ii),(jj)}$  by a least-square strategy as in Step 2.1b.
    3. Compute  $(u^n, \psi^{(ii),(jj)})_{E_j^{(ii),(jj)}}$ ,  $(\Delta u^{(jj)}, \psi^{(ii),(jj)})_{E_j^{(ii),(jj)}}$  and  $(g^{(jj)}, \psi^{(ii),(jj)})_{E_j^{(ii),(jj)}}$ 
    by SLDG method as in Step 2.1c.
    4. Evaluate  $(g^{(ii)}, \Psi)_{E_j}$  by quadrature rules.
  end for
  Compute  $u^{(ii)}$  from Eq. (2.33).
  Compute  $\Delta u^{(ii)}$  with LDG method.
end for
 $u^{n+1} = u^{(s)}$ .

```

---



- (i) Cell-E: Eulerian Cell  $E_j$ .  
 Node-E:  $\{v_q\}_{q=1}^4$  for  $P^0 \& P^1$  or  $\{v_q\}_{q=1}^9$  for  $P^2$ .  
 SLDG-LDG solution:  $u^n \in P^k(E_j)$ .
- (ii) Cell-U: Upstream Cell  $E_j^{(ii),(jj)}$ .  
 Node-U:  $\{v_q^{(ii),(jj)}\}_{q=1}^4$  for  $P^0 \& P^1$  or  $\{v_q^{(ii),(jj)}\}_{q=1}^9$  for  $P^2$ .  
 Test function:  $\psi^{(ii),(jj)}$ .  
 ( $ii, jj$  here refer to the indexes in the **Algorithm 1**).

**Fig. 5.** Data structure of 2D SLDG-LDG schemes.

Finally, we present the data structure for setting up the 2D SLDG-LDG implementation, which is similar to that in [5]. There are two main classes as specified below. Please see Fig. 5 in which the lines in the figure indicate connections between classes.

- (i) **Cell-E** representing Eulerian cells, e.g.  $E_j$  in Fig. 4(a). Main variables are
- **Node-E**:  $\{v_q\}_{q=1}^4$  for  $P^0$  and  $P^1$ ;  $\{v_q\}_{q=1}^9$  for  $P^2$  as vertices of Eulerian cell.
  - **SLDG-LDG solution**.
- (ii) **Cell-U** representing upstream cells, e.g.  $E_j^{n+1,n}$  in Fig. 4(a). Main variables are
- **Node-U**:  $\{v_q^{(ii),(jj)}\}_{q=1}^4$  for  $P^0$  and  $P^1$ ;  $\{v_q^{(ii),(jj)}\}_{q=1}^9$  for  $P^2$  as vertices of upstream cell.
  - **Test function**: e.g.  $\psi^{n+1,n}(x, y)$  approximated by a least-square procedure and by following characteristics of the adjoint problem, see Step 2.1b.

### 2.3. Stability analysis

We now briefly discuss the mass conservation and stability properties of the proposed SLDG-LDG schemes when coupled with first order backward Euler method. Stability analysis of our scheme coupling with higher order time discretization will be pursued in the future.

**Proposition 2.6** (Mass conservative). *The SLDG-LDG method coupled with any DIRK time discretization methods for the linear convection-diffusion problems enjoy the mass conservation property, assuming the source term  $g = 0$  in Eq. (2.1) and periodic boundary condition.*

$$\int_{\Omega} u^{n+1} d\mathbf{x} = \int_{\Omega} u^n d\mathbf{x}.$$

**Proof.** This proposition can be easily proved by letting the test function  $\Psi = 1$  in Eq. (2.7) and Eq. (2.26) for 1D and 2D cases respectively, and then making use of the flux form of the LDG approximation of the diffusion term.  $\square$

**Proposition 2.7** ( $L^2$  stability). Consider the proposed SLDG-LDG scheme coupled with the first-order backward Euler time discretization for 1D linear convection-diffusion equation  $u_t + u_x = \epsilon u_{xx}$ ,  $\epsilon > 0$  and periodic boundary condition, then:

$$\|u^{n+1}\| \leq \|u^n\|, \quad (2.34)$$

where  $\|\cdot\|$  denotes the standard  $L^2$  norm over  $\Omega$ .

**Proof.** The SLDG-LDG scheme for the 1D linear problem writes

$$(u^{n+1}, \Psi)_{I_j} - (u^n, \psi^{n+1,n})_{I_j^{n+1,n}} = \epsilon \cdot \Delta t \left( (\hat{q}^{n+1} \Psi)|_{j-\frac{1}{2}}^{j+\frac{1}{2}} - (q^{n+1}, \Psi_x)_{I_j} \right), \quad (2.35a)$$

$$(q^{n+1}, \varphi)_{I_j} = (\hat{u}^{n+1} \varphi)|_{j-\frac{1}{2}}^{j+\frac{1}{2}} - (u^{n+1}, \varphi_x)_{I_j} \quad (2.35b)$$

where  $\Psi, \varphi \in V_h^k$ . As a standard technique for proving the stability, we take the test function  $\Psi = u^{n+1}$  and  $\varphi = q^{n+1}$  on  $I_j$  in Eq. (2.35a) and Eq. (2.35b), respectively. According to (2.3), we have

$$\psi^{n+1,n} = u^{n+1}(x + \Delta t) \doteq \delta_{\Delta t} u^{n+1}.$$

This, together with the weak formulations in Eq. (2.14), yields

$$(u^{n+1}, u^{n+1})_{I_j} - (u^n, \delta_{\Delta t} u^{n+1})_{I_j^{n+1,n}} = \epsilon \cdot \Delta t \left( (\hat{q}^{n+1} u^{n+1})|_{j-\frac{1}{2}}^{j+\frac{1}{2}} - (q^{n+1}, u_x^{n+1})_{I_j} \right), \quad (2.36a)$$

$$(q^{n+1}, q^{n+1})_{I_j} = (\hat{u}^{n+1} q^{n+1})|_{j-\frac{1}{2}}^{j+\frac{1}{2}} - (u^{n+1}, q_x^{n+1})_{I_j}. \quad (2.36b)$$

(2.36a) +  $\epsilon \cdot \Delta t \cdot$  (2.36b) and summing up over  $j$  give us

$$\begin{aligned} \|u^{n+1}\|^2 - \int_{x_a}^{x_b} u^n \cdot \delta_{\Delta t} u^{n+1} dx + \epsilon \Delta t \|q^{n+1}\|^2 \\ = \epsilon \cdot \Delta t \cdot \sum_j \left\{ (\hat{q}^{n+1} u^{n+1})|_{j-\frac{1}{2}}^{j+\frac{1}{2}} + (\hat{u}^{n+1} q^{n+1})|_{j-\frac{1}{2}}^{j+\frac{1}{2}} - [(q^{n+1}, u_x^{n+1})_{I_j} + (u^{n+1}, q_x^{n+1})_{I_j}] \right\} \\ = \epsilon \cdot \Delta t \cdot \sum_j \left\{ (\hat{q}^{n+1} u^{n+1})|_{j-\frac{1}{2}}^{j+\frac{1}{2}} + (\hat{u}^{n+1} q^{n+1})|_{j-\frac{1}{2}}^{j+\frac{1}{2}} - \int_{I_j} (q^{n+1} u^{n+1})_x dx \right\} \\ = \epsilon \cdot \Delta t \cdot \sum_j \left\{ (\hat{q}^{n+1} u^{n+1})|_{j-\frac{1}{2}}^{j+\frac{1}{2}} + (\hat{u}^{n+1} q^{n+1})|_{j-\frac{1}{2}}^{j+\frac{1}{2}} - (q^{n+1} u^{n+1})|_{j-\frac{1}{2}}^{j+\frac{1}{2}} \right\} = 0, \end{aligned} \quad (2.37)$$

where the cancellation is due to the alternating fluxes used, see (2.15), and the periodicity.

Noting that  $\epsilon \Delta t \|q^{n+1}\|^2 \geq 0$  on the LHS of (2.37), we have

$$\|u^{n+1}\|^2 \leq \int_{x_a}^{x_b} u^n \cdot \delta_{\Delta t} u^{n+1} dx. \quad (2.38)$$

Then, applying the Cauchy-Schwarz inequality to (2.38) together with the identity  $\|\delta_{\Delta t} u^{n+1}\| = \|u^{n+1}\|$  yields

$$\|u^{n+1}\| \leq \|u^n\|. \quad (2.39)$$

This completes the proof.  $\square$

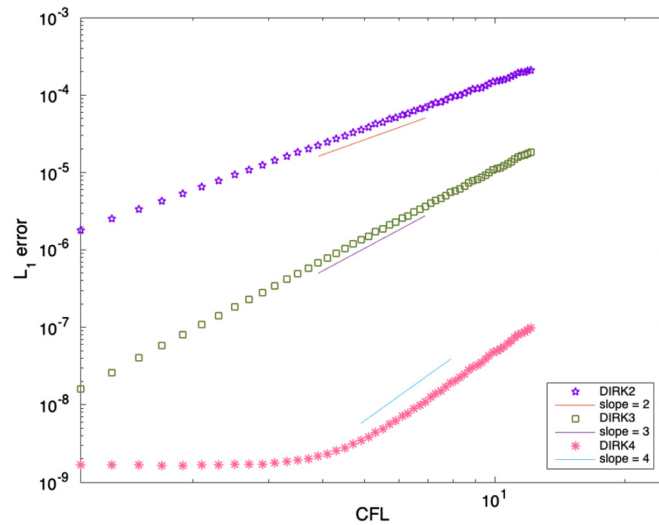
### 3. Numerical tests

In this section, we present the convergence study in terms of spatial and temporal orders of the proposed SLDG-LDG methods for a collection of 1D and 2D benchmark linear convection-diffusion equations. Mass conservation in Proposition 2.6 is also numerically verified. We assume uniform partition of the computational domain  $\Omega$ . In principle, the addressed schemes can be extended to general nonuniform meshes. We let  $\Delta t = \frac{\text{CFL}}{\max_{\Delta x} |a(x,t)|}$  and  $\Delta t = \frac{\text{CFL}}{\max_{\Delta x} |a(x,y,t)| + \max_{\Delta y} |b(x,y,t)|}$

for 1D and 2D tests respectively, in which the CFL number is to be specified. For the test of spatial accuracy, we choose DIRK4 with Butcher tableau specified in Table 8 in order to minimize the time discretization error. Likewise, for temporal accuracy tests, we use SLDG-LDG with piecewise  $P^2$  polynomial space unless otherwise specified.

**Table 2**Spatial order of accuracy for Example 3.1 with  $CFL = 1.0$ ,  $\epsilon = 1$  at  $T = 1.0$ .

$k = 0$						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
10	3.79E-02		4.78E-02		1.08E-01	
20	1.92E-02	0.98	2.40E-02	0.99	5.45E-02	0.99
40	9.41E-03	1.03	1.18E-02	1.02	2.70E-02	1.02
80	4.70E-03	1.00	5.90E-03	1.00	1.35E-02	1.00
160	2.35E-03	1.00	2.95E-03	1.00	6.74E-03	1.00
$k = 1$						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
10	4.60E-03		5.57E-03		1.15E-02	
20	1.21E-03	1.92	1.50E-03	1.90	4.27E-03	1.43
40	2.88E-04	2.07	3.70E-04	2.01	1.17E-03	1.87
80	7.01E-05	2.04	9.28E-05	2.00	3.04E-04	1.94
160	1.78E-05	1.98	2.39E-05	1.96	7.95E-05	1.94
$k = 2$						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
10	2.18E-04		3.19E-04		1.08E-03	
20	2.57E-04	3.09	3.92E-05	3.03	1.36E-04	2.99
40	3.32E-06	2.95	5.05E-06	2.96	1.77E-05	2.95
80	4.00E-07	3.05	6.02E-07	3.07	2.05E-06	3.11
160	5.10E-08	2.97	7.73E-08	2.96	2.68E-07	2.94

**Fig. 6.**  $L^1$  error with varying CFL numbers of Example 3.1. SLDG-LDG with  $P^2$  polynomial space.  $N = 500$  for spatial discretization.

### 3.1. One-dimensional tests

**Example 3.1** (1D linear convection-diffusion equation). Consider the following 1D convection-diffusion equation

$$u_t + u_x = \epsilon u_{xx}, \quad x \in [0, 2\pi] \quad (3.1)$$

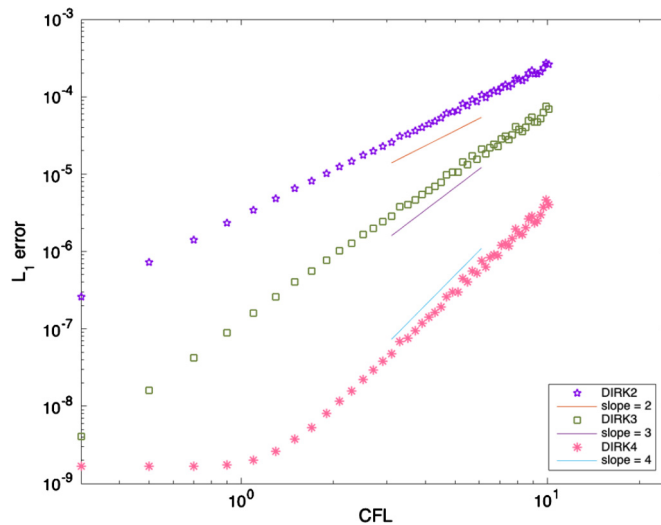
with exact solution  $u = \sin(x - t) \exp(-\epsilon t)$ . Table 2 provides  $L^1$ ,  $L^2$ ,  $L^\infty$  errors to verify the spatial performance of the SLDG-LDG method. One can observe the method is of  $k + 1$  order when  $V_h^k$  is used, as expected. To demonstrate the temporal orders of accuracy of the employed DIRK methods, we fix  $N = 500$  and let the CFL numbers vary from 1.1 to 12.1 (note the extra large values of CFL numbers). Fig. 6 shows the  $L^1$  error of the proposed scheme coupled with DIRK2, DIRK3 and DIRK4 and expected orders can be observed as compared with reference slopes. Note that for this problem there is no error incurred in time for the convection part, since the characteristics are tracked exactly in the proposed SL setting.

**Example 3.2** (1D equation with variable coefficient). Consider

$$u_t + (\sin(x)u)_x = \epsilon u_{xx} + g, \quad x \in [0, 2\pi] \quad (3.2)$$

**Table 3**Spatial order of accuracy for Example 3.2 with  $CFL = 1.0$ ,  $\epsilon = 1$  at  $T = 1.0$ .

$k = 0$						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
10	4.20E-02		4.96E-02		1.11E-01	
20	1.97E-02	1.09	2.42E-02	1.03	5.41E-02	1.05
40	9.96E-03	0.98	1.22E-02	0.99	2.71E-02	1.00
80	4.97E-03	1.00	6.11E-03	0.99	1.35E-02	1.00
160	2.50E-03	1.00	3.07E-03	1.00	6.80E-03	0.99
$k = 1$						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
10	6.24E-03		8.42E-03		3.11E-02	
20	1.33E-03	2.23	1.78E-03	2.25	6.53E-03	2.25
40	3.06E-04	2.12	3.20E-04	2.08	1.57E-03	2.05
80	7.39E-05	2.05	1.04E-04	2.02	3.91E-04	2.01
160	1.85E-05	2.00	2.62E-05	1.98	9.42E-05	2.05
$k = 2$						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
10	4.29E-04		5.38E-04		1.69E-03	
20	9.53E-05	2.17	1.09E-04	2.31	2.60E-04	2.70
40	8.16E-06	3.55	9.63E-06	3.49	2.70E-05	3.27
80	7.72E-07	3.40	9.37E-07	3.36	3.03E-06	3.15
160	7.57E-08	3.35	9.60E-08	3.29	3.39E-07	3.16

**Fig. 7.**  $L^1$  error with varying CFL numbers of Example 3.2. SLDG-LDG with  $P^2$  polynomial space.  $N = 500$  for spatial discretization.

with  $u = \sin(x) \exp(-\epsilon t)$  and  $g = \sin(2x) \exp(-\epsilon t)$ . Expected spatial orders of accuracy are observed in Table 3. With fixed mesh  $N = 500$  and CFL varying from 0.3 to 12.1, high order temporal convergences are observed in Fig. 7.

### 3.2. Two-dimensional tests

**Example 3.3** (2D linear convection-diffusion equation). Consider

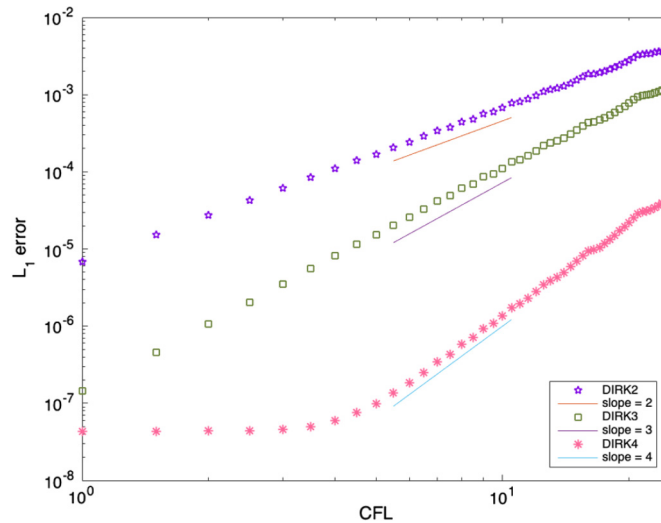
$$u_t + u_x + u_y = \epsilon \Delta u, \quad x, y \in [0, 2\pi] \quad (3.3)$$

with exact solution  $u = \sin(x + y - 2t) \exp(-2\epsilon t)$ .

We observe  $k + 1$  spatial orders of accuracy from Table 4 when  $V_h^k$  is employed. The temporal convergence study is summarized in Fig. 8 with fixed a mesh of  $J = 200^2$  cells. The observation is similar to that in the 1D case (Example 3.1). To verify the mass conservation property claimed in Proposition 2.6 for Example 3.3, we plot  $|\iint_{\Omega} u^n dx|$  with  $V_h^0$  computed using different thresholds  $10^{-10}$ ,  $10^{-12}$  and  $10^{-14}$  for the GMRES iterative method in Fig. 9. We see that the variation of  $|\iint_{\Omega} u^n dx|$  in time depends on the choice of threshold and it gets closer to the machine precision when a smaller threshold in GMRES is used.

**Table 4**Spatial order of accuracy for Example 3.3 with  $CFL = 1.0$ ,  $\epsilon = 1$  at  $T = 1.0$ .

$k = 0$ ; Quadrilateral						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
$20^2$	4.64E-02		5.15E-02		7.67E-02	
$60^2$	1.99E-02	0.77	2.21E-02	0.77	3.30E-02	0.77
$100^2$	1.26E-02	0.89	1.40E-02	0.89	2.09E-02	0.89
$140^2$	9.23E-03	0.93	1.03E-02	0.93	1.53E-02	0.93
$180^2$	7.27E-03	0.95	8.08E-03	0.95	1.20E-02	0.95
$k = 1$ ; Quadrilateral						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
$20^2$	1.10E-03		1.35E-03		5.23E-03	
$60^2$	9.59E-05	2.22	1.28E-04	2.14	6.94E-04	1.84
$100^2$	3.28E-05	2.10	4.52E-05	2.04	2.57E-04	1.95
$140^2$	1.65E-05	2.05	2.31E-05	2.00	1.33E-04	1.95
$180^2$	9.87E-06	2.04	1.40E-05	2.00	8.14E-05	1.96
$k = 2$ ; Quadrilateral						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
$20^2$	4.14E-05		6.06E-05		4.82E-04	
$60^2$	1.59E-06	2.96	2.35E-06	2.96	1.88E-05	2.95
$100^2$	3.45E-07	3.00	5.09E-07	2.99	4.08E-06	2.99
$140^2$	1.26E-07	2.99	1.86E-07	2.99	1.49E-06	3.00
$180^2$	5.96E-08	2.98	8.78E-08	2.99	7.03E-07	2.99

**Fig. 8.**  $L_1$  error with varying CFL number of Example 3.3. From top to bottom: DIRK2, DIRK3 and DIRK4.**Example 3.4** (Rigid body rotation). Consider

$$u_t - (yu)_x + (xu)_y = \epsilon \Delta u + g, \quad x, y \in [-2\pi, 2\pi]. \quad (3.4)$$

In order to test the spatial order of accuracy, we choose  $u = \exp[-(x^2 + 3y^2 + 2\epsilon t)]$  and  $g = [6\epsilon - 4xy - 4\epsilon(x^2 + 9y^2)]\exp[-(x^2 + 3y^2 + 2\epsilon t)]$ . The results are summarized in Table 5 for  $\epsilon = 1.0$ ,  $CFL = 10.0$ . The choice of the large CFL number once again supports our claim that the proposed methods are free of the stringent CFL restriction, leading to computational savings. Fig. 10 displays the temporal convergence study where we use two different ways to compute the numerical error: test1 is done by using the exact solution on mesh  $J = 200^2$ ; test2 uses the reference solution when  $CFL = 0.01$  on mesh  $J = 150^2$  and the final time  $T = 0.1$ . In Fig. 10, when test1 is used, expected orders for DIRK2 and DIRK3 are observed, while the numerical convergence order is not clear for DIRK4. In order to reduce the interference of the spatial error, we carry out test2 and observe the expected order of temporal accuracy.

We numerically solve Eq. (3.4) with an initial condition plotted in Fig. 11 which includes a slotted disk, a cone as well as a smooth hump.  $\epsilon = 0.01$ ,  $g = 0$  and mesh size  $J = 200^2$  are chosen here. The numerical solutions and the corresponding contour plots obtained by the proposed SLDG-LDG schemes with  $V_h^k$  space ( $k = 1, 2$ ) after  $T = 1.0$  are plotted in Fig. 12. DIRK4 time discretization method is applied.

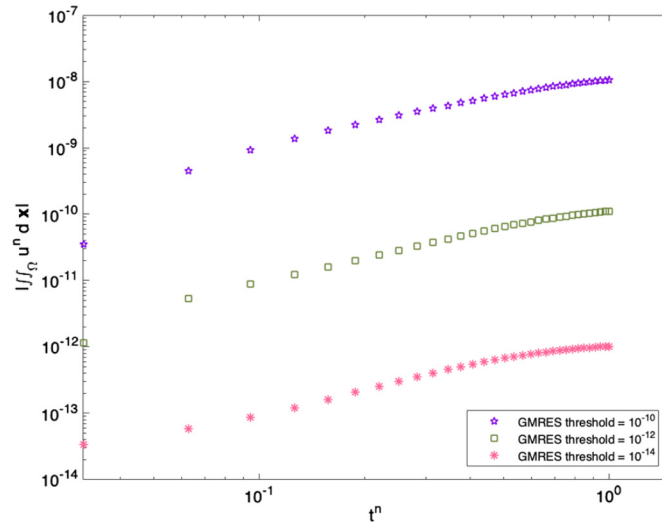


Fig. 9.  $|\int_{\Omega} u^n dx|$  at different  $t^n$  for Example 3.3 with  $V_h^0$  space. From top to bottom: GMRES iterative method with thresholds  $10^{-10}$ ,  $10^{-12}$  and  $10^{-14}$ .

Table 5

Spatial order of accuracy for Example 3.4 with CFL = 10.0,  $\epsilon = 1.0$  at  $T = 1.0$ .

$k = 0$ ; Quadrilateral						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
$20^2$	1.97E-03		6.53E-03		9.39E-02	
$60^2$	8.86E-04	0.73	3.08E-03	0.68	4.62E-02	0.65
$100^2$	5.73E-04	0.85	2.02E-03	0.82	2.99E-02	0.85
$140^2$	4.23E-04	0.90	1.51E-03	0.88	2.23E-02	0.87
$180^2$	3.36E-04	0.92	1.20E-03	0.91	1.78E-02	0.90
$k = 1$ ; Quadrilateral						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
$20^2$	2.76E-04		1.30E-03		2.81E-02	
$60^2$	2.68E-05	2.12	1.58E-04	1.92	5.75E-03	1.45
$100^2$	9.33E-06	2.07	5.72E-05	1.99	2.25E-03	1.83
$140^2$	4.70E-06	2.04	2.94E-05	1.98	1.19E-03	1.91
$180^2$	2.82E-06	2.03	1.78E-05	1.98	7.29E-04	1.94
$k = 2$ ; Quadrilateral						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
$20^2$	7.11E-05		3.51E-04		1.32E-02	
$60^2$	1.92E-06	3.29	1.18E-05	3.08	5.31E-04	2.92
$100^2$	4.04E-07	3.05	2.57E-06	2.99	1.14E-04	3.01
$140^2$	1.46E-07	3.03	9.39E-07	3.00	4.24E-05	2.94
$180^2$	6.82E-08	3.02	4.41E-07	3.00	2.00E-05	2.99

To better compare the performances of the schemes with  $V_h^1$  and  $V_h^2$  spaces for the DG and LDG discretization, we plot the 1D cuts of the numerical solutions at  $X = -1.0$  and  $Y = 1.0$  with mesh size  $J = 50^2$  along with the one computed on a refined mesh with  $J = 200^2$  as in Fig. 13 at  $T = 1.0$ .

**Example 3.5** (Swirling deformation flow). Consider

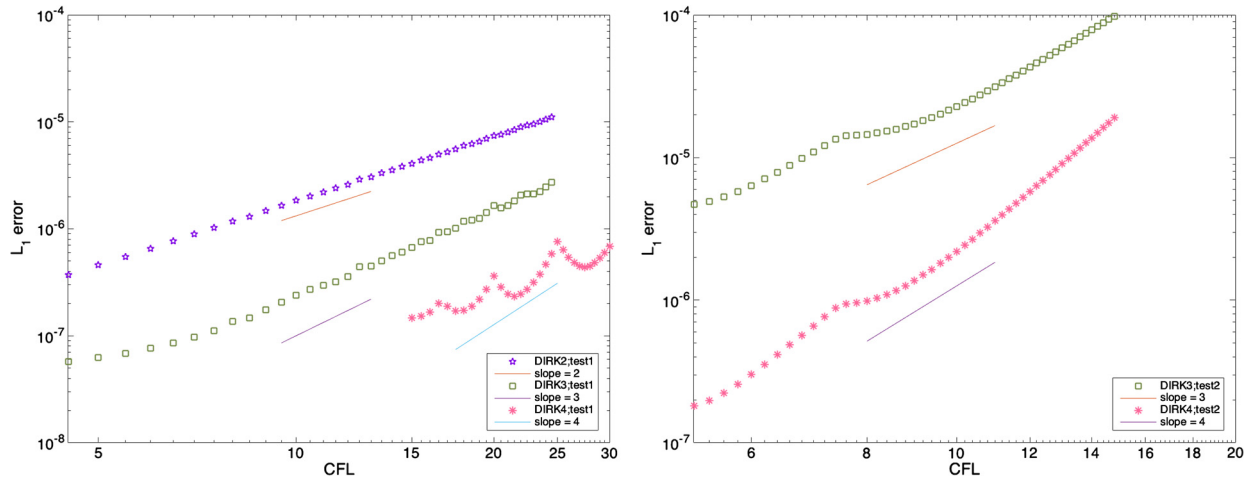
$$u_t - \left(\cos\left(\frac{x}{2}\right)^2 \sin(y) f(t) u\right)_x + \left(\sin(x) \cos\left(\frac{y}{2}\right)^2 f(t) u\right)_y = \epsilon \Delta u, \quad x, y \in [-\pi, \pi] \quad (3.5)$$

with  $f(t) = \cos\left(\frac{\pi t}{T}\right)\pi$ .

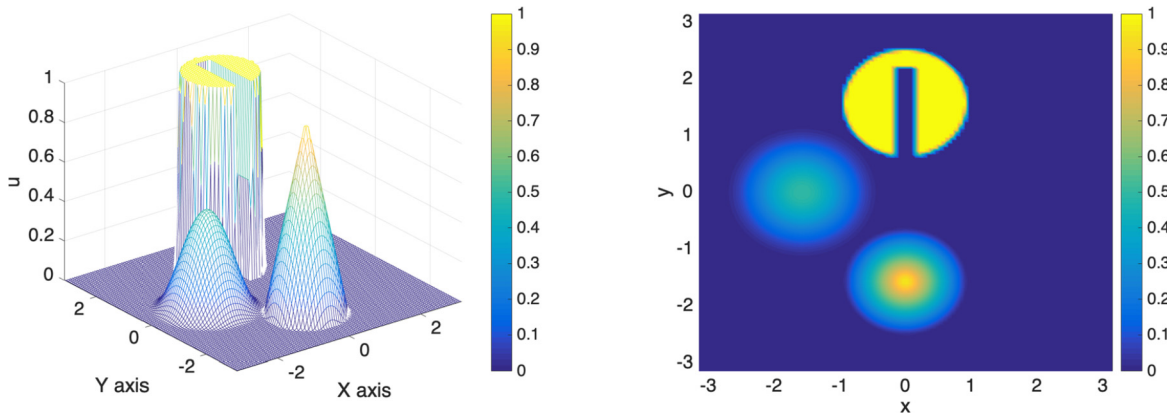
To test the spatial order of accuracy, the initial condition is set to be

$$u(x, y, 0) = \begin{cases} r_0^b \cos\left(\frac{r^b(\mathbf{x})\pi}{2r_0^b}\right)^6, & \text{if } r^b(\mathbf{x}) < r_0^b, \\ 0, & \text{otherwise,} \end{cases} \quad (3.6)$$

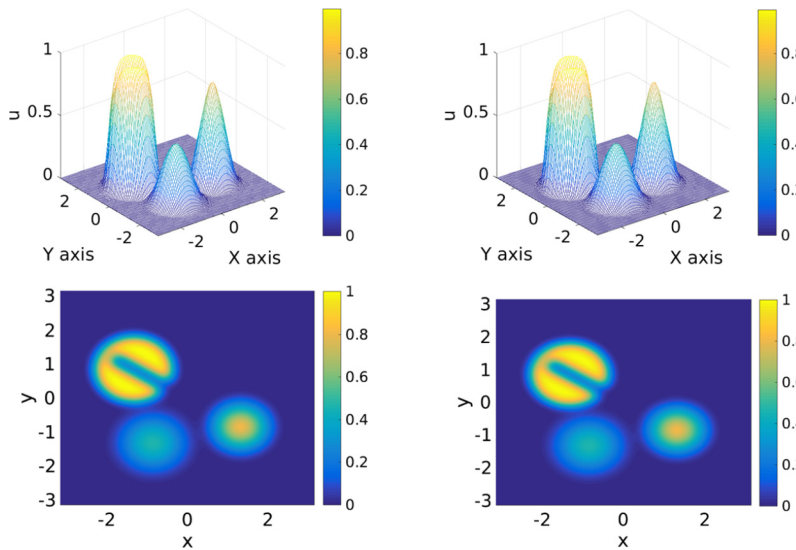




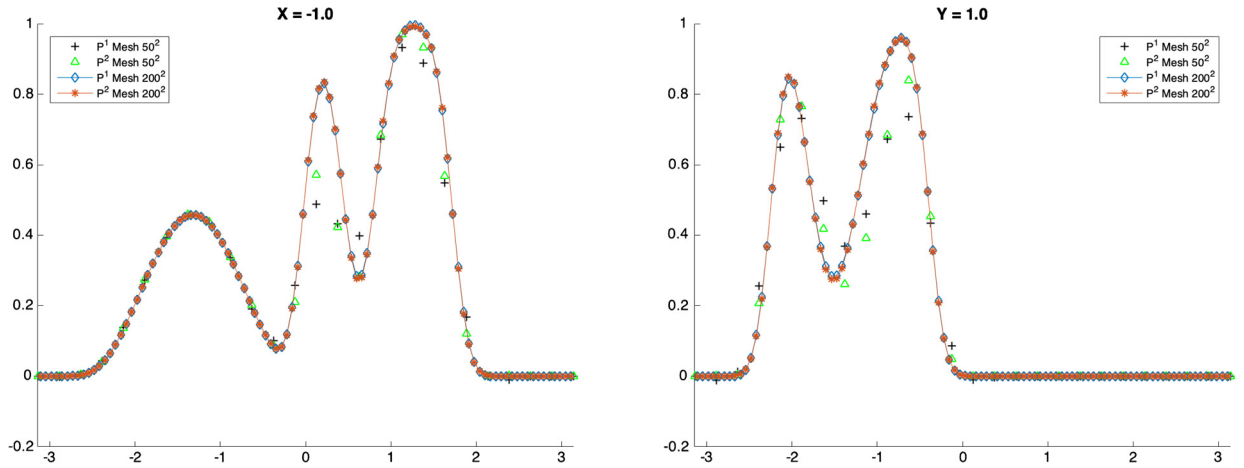
**Fig. 10.**  $L^1$  error with varying CFL number of Example 3.4. Left: test1 using exact solution on mesh  $J = 200^2$ . Right: test2 using reference solution when  $CFL = 0.01$  on mesh  $J = 150^2$ .



**Fig. 11.** Plots of the initial profile consisting of a slotted disk, a cone and a smooth hump for Example 3.4. Mesh size is  $200 \times 200$ . Left: Initial condition. Right: Contour plot. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)



**Fig. 12.** First row consists of plots for numerical solutions with SLDG-LDG for equation (3.4) with initial data Fig. 11 and the second row consists of the corresponding contour plots. Mesh size is  $200 \times 200$ . Final integration time  $T = 1.0$ .  $\Delta t = 2.5\Delta x$ . From left to right:  $P^1$  SLDG-LDG,  $P^2$  SLDG-LDG.



**Fig. 13.** Plots of the 1D cuts of the numerical solutions for equation (3.4) at  $X = -1.0$  (left) and  $Y = 1.0$  (right) with initial data in Fig. 11 with  $CFL = 2.5$  at  $T = 1.0$ .

**Table 6**

Spatial order of accuracy for Example 3.5 with  $CFL = 1.0$ ,  $\epsilon = 1.0$  at  $T = 0.1$ .

$k = 0$ ; Quadrilateral						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
$20^2$	2.15E-03		6.50E-03		5.86E-02	
$60^2$	7.56E-04	2.05	2.42E-03	1.94	2.15E-02	1.96
$100^2$	3.60E-04	2.20	1.18E-03	2.13	1.13E-02	1.92
$140^2$	2.83E-04	0.96	9.35E-04	0.93	1.20E-02	-0.23
$180^2$	2.29E-04	1.05	7.58E-04	1.04	8.08E-03	1.95
$k = 1$ ; Quadrilateral						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
$20^2$	2.93E-04		9.13E-04		1.49E-02	
$60^2$	3.17E-05	4.36	1.04E-04	4.25	2.14E-03	3.80
$100^2$	1.10E-05	3.13	3.63E-05	3.14	7.66E-04	3.06
$140^2$	5.90E-06	2.50	1.98E-05	2.42	4.53E-04	2.09
$180^2$	3.83E-06	2.16	1.29E-05	2.14	3.04E-04	1.98
$k = 2$ ; Quadrilateral						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
$20^2$	3.42E-05		1.10E-04		2.73E-03	
$60^2$	1.35E-06	6.33	4.74E-06	6.15	1.22E-04	6.09
$100^2$	3.00E-07	4.47	1.06E-06	4.44	2.67E-05	4.51
$140^2$	1.10E-07	3.98	3.96E-07	3.94	1.00E-05	3.90
$180^2$	5.32E-08	3.63	1.91E-07	3.63	5.04E-06	3.43
$k = 2$ ; QC						
Mesh	$L^1$ error	Order	$L^2$ error	Order	$L^\infty$ error	Order
$20^2$	3.30E-05		1.06E-04		2.53E-03	
$60^2$	1.14E-06	6.58	3.75E-06	6.54	9.73E-05	6.38
$100^2$	2.48E-07	4.54	8.15E-07	4.54	2.09E-05	4.58
$140^2$	9.07E-08	4.01	3.00E-07	3.98	7.83E-06	3.90
$180^2$	4.32E-08	3.70	1.42E-07	3.71	3.86E-06	3.53

where  $r_0^b = 0.3\pi$  and  $r^b(\mathbf{x}) = \sqrt{(x - x_0^b)^2 + (y - y_0^b)^2}$  denotes the distance between  $(x, y)$  and the center of the cosine bell  $(x_0^b, y_0^b) = (0.3\pi, 0)$ . Since there is no analytical solution available, we choose the numerical solution computed on a refined mesh with  $J = 300^2$  cells as a reference solution. The results are shown in Table 6. Note that comparable performance are observed when quadrilateral and quadratic curved (QC) approximations to the upstream cells are used. Fig. 14 shows the temporal convergence study for the proposed scheme coupled with different DIRK methods. A mesh of  $J = 200^2$  cells is used. The solution computed with  $CFL = 0.01$  is chosen as the reference solution. The final time is 0.1. Again, expected high order temporal convergence rates are observed for all three cases.

Additionally, we solve Eq. (3.5) with the same initial condition as in Fig. 11. We choose  $\epsilon = 0.01$ ,  $T = 1.5$  and numerically integrate the solution up to time 1.5. Numerical solutions and the corresponding contour plots for the proposed schemes

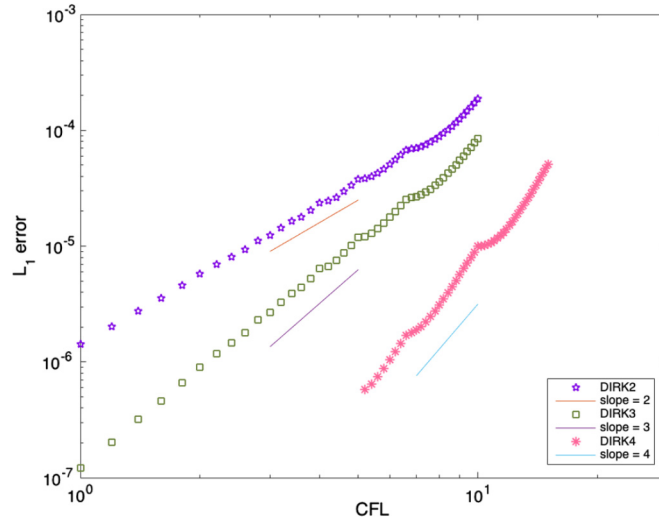


Fig. 14.  $L_1$  error with varying CFL number of Example 3.5. From top to bottom: DIRK2, DIRK3 and DIRK4.

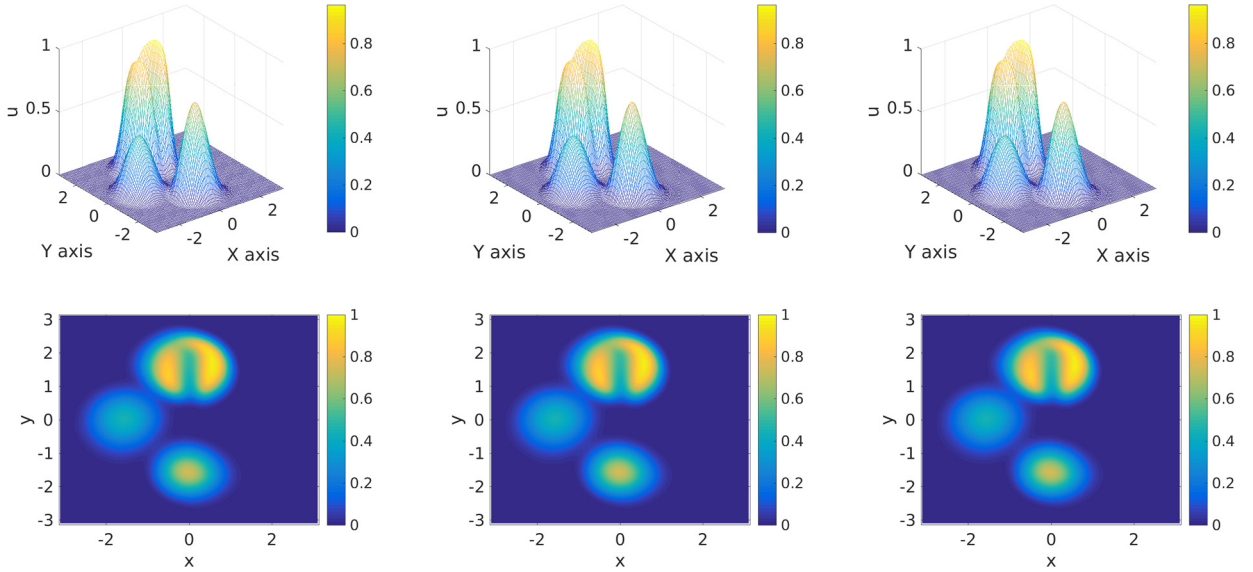


Fig. 15. First row consist of mesh plots of SLDG-LDG solutions for equation (3.5) with  $f(t) = \cos(\frac{\pi t}{T})\pi$ , initial data Fig. 11 and the second row consists of the corresponding contour plots. Mesh size is  $100 \times 100$ . Final integration time  $T = 1.5$ .  $\Delta t = 2.5\Delta x$ . From left to right:  $P^1$  SLDG-LDG,  $P^2$  SLDG-LDG,  $P^2$  SLDG-LDG+QC.

are plotted in Fig. 15. The 1D cuts of the numerical solutions of Eq. (3.5) at  $X = 0$  and  $Y = 1.54$  with mesh size  $J = 50^2$  along with the one computed on a refined mesh with  $J = 200^2$  are plotted in Fig. 16. From the 1D cut plot at  $X = 0$ , both  $P^2$  SLDG-LDG and  $P^2$  SLDG-LDG-QC schemes perform better than  $P^1$  scheme on mesh of size  $50^2$ .

#### 4. Conclusions

In this paper, we developed a semi-Lagrangian (SL) discontinuous Galerkin (DG) method for solving linear convection-diffusion equations. For the scheme formulation, the DG solution is evolved along the characteristics to treat the convection part by an efficient SLDG transport method; while the diffusion part is discretized by a local DG method in conjunction with diagonally implicit Runge-Kutta methods along characteristics. The method is high order accurate, mass conservative and is unconditionally stable. In the theoretical aspect, the unconditional  $L^2$  stability was proved for the method coupled with the backward Euler discretization. In view of the application of the SLDG method coupled with RK exponential integrators to nonlinear Vlasov dynamics in [4], extensions of our algorithm to nonlinear convection-diffusion problems will be investigated in our future research work.

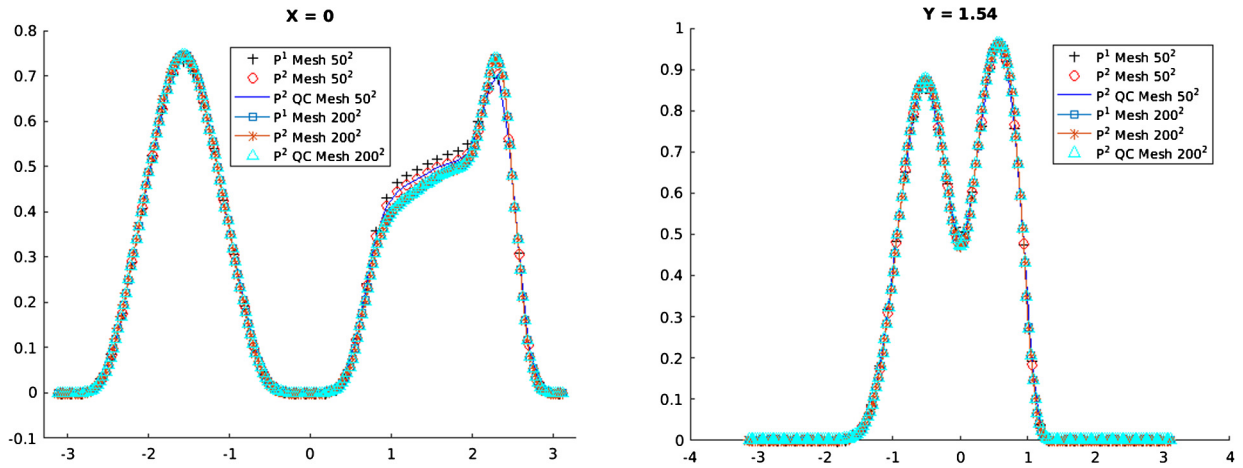


Fig. 16. Plots of the 1D cuts of the numerical solutions for equation (3.5) at  $X = 0$  (left) and  $Y = 1.54$  (right) with initial data Fig. 11 at with  $CFL = 2.5$  at  $T = 1.5$ .

Table 7

DIRK3.  $\gamma \approx 0.435866521508459$ ,  
 $\beta_1 = -\frac{3}{2}\gamma^2 + 4\gamma - \frac{1}{4}$ ,  $\beta_2 = \frac{3}{2}\gamma^2 - 5\gamma + \frac{5}{4}$ .

$\frac{\gamma}{1+\gamma}$	$\frac{\gamma}{1-\gamma}$	$\gamma$
1	$\beta_1$	$\beta_2$
	$\beta_1$	$\beta_2$

Table 8

DIRK4.

$\frac{1}{4}$	$\frac{1}{4}$			
$\frac{3}{4}$	$\frac{1}{2}$	$\frac{1}{4}$		
$\frac{11}{20}$	$\frac{17}{50}$	$-\frac{1}{25}$	$\frac{1}{4}$	
$\frac{1}{2}$	$\frac{371}{1360}$	$-\frac{137}{2720}$	$\frac{15}{544}$	$\frac{1}{4}$
1	$\frac{25}{24}$	$-\frac{49}{48}$	$\frac{125}{16}$	$-\frac{85}{12}$
	$\frac{25}{24}$	$-\frac{49}{48}$	$\frac{125}{16}$	$-\frac{85}{12}$

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Appendix A

### A.1. Butcher tableaus for time discretization methods

In the following Table 7 and Table 8, we present the third-order DIRK3 and fourth-order DIRK4 respectively. Both tableaus are stiffly accurate.

### A.2. Implementation procedures

We now briefly discuss the components of the sparse matrix  $B_1$  in (2.18). Let  $\varphi_l$  ( $l = 0, \dots, k$ ) denote the local bases of  $P^k(I_j)$ , the numerical solution  $u$  can be written as

$$u = \sum_{l=0}^k u_l \varphi_l \doteq \mathbf{u} \cdot \boldsymbol{\varphi}, \quad x \in I_j,$$

where  $\mathbf{u} = (u_0, \dots, u_k)$  and  $\boldsymbol{\varphi} = (\varphi_0, \dots, \varphi_k)^T$ . The basis functions are chosen as the scaled Legendre polynomials. For instance, for  $k = 2$ ,  $P^2(I_j) = \{1, \xi, \xi^2 - \frac{1}{12}\}$  with  $\xi = \frac{x-x_j}{\Delta x}$ .

To update  $u^{n+1}$ ,

$$(u^{n+1}, \varphi_m)_{I_j} = \left( \sum_{l=0}^k u_l^{n+1} \varphi_l, \varphi_m \right)_{I_j} = \sum_{l=0}^k u_l^{n+1} (\varphi_l, \varphi_m)_{I_j}. \quad (\text{A.1})$$

Then, as  $\varphi_m$  in (A.1) going through bases in  $P^k(I_j)$ ,

$$(u^{n+1}, \varphi)_{I_j} = M(\mathbf{u}^{n+1})_{I_j}^T$$

where  $M$  is the mass matrix of size  $(k+1)^2$  with  $M_{m,l} = (\varphi_l, \varphi_m)_{I_j}$  ( $m, l = 0, 1, \dots, k$ ).

Following the procedures in Step 1.2a, the LDG spatial approximation to  $\Delta \mathbf{u}^{n+1}$  can be represented by a matrix vector form as  $D_\Delta \mathbf{u}^{n+1}$ , where  $D_\Delta$  is the matrix approximating diffusion operator via the LDG formulation.

- For (2.14b), suppose  $\hat{u} = u^-$ , then

$$\hat{u}_{j+\frac{1}{2}}^{n+1} \mathbf{w}_{j+\frac{1}{2}}^- = C(\mathbf{u}^{n+1})_{I_j}^T, \quad \hat{u}_{j-\frac{1}{2}}^{n+1} \mathbf{w}_{j-\frac{1}{2}}^+ = D(\mathbf{u}^{n+1})_{I_{j-1}}^T, \quad (u^{n+1}, \mathbf{w}_x)_{I_j} = N(\mathbf{u}^{n+1})_{I_j}^T$$

with  $(m = 0, 1, \dots, k)$

$$C_{m,l} = \varphi_l(x_{j+\frac{1}{2}}) \varphi_m(x_{j+\frac{1}{2}}), \quad D_{m,l} = \varphi_l(x_{j-\frac{1}{2}}) \varphi_m(x_{j-\frac{1}{2}}), \quad N_{m,l} = (\varphi_l, (\varphi_m)_x)_{I_j}$$

as  $w$  going through  $\{\varphi_l\}_{l=0}^k$ .

- For (2.14a), choose  $\hat{q} = q^+$ , then

$$\hat{q}_{j+\frac{1}{2}}^{n+1} \mathbf{v}_{j+\frac{1}{2}}^- = E(\mathbf{q}^{n+1})_{I_{j+1}}^T, \quad \hat{q}_{j-\frac{1}{2}}^{n+1} \mathbf{v}_{j-\frac{1}{2}}^+ = F(\mathbf{q}^{n+1})_{I_j}^T, \quad (q^{n+1}, \mathbf{v}_x)_{I_j} = N(\mathbf{q}^{n+1})_{I_j}^T$$

with  $(m = 0, 1, \dots, k)$

$$E_{m,l} = \varphi_l(x_{j+\frac{1}{2}}) \varphi_m(x_{j+\frac{1}{2}}), \quad F_{m,l} = \varphi_l(x_{j-\frac{1}{2}}) \varphi_m(x_{j-\frac{1}{2}}),$$

as  $v$  going through  $\{\varphi_l\}_{l=0}^k$ .

Then,  $(u_{xx}^{n+1}, \varphi)_{I_j} (\forall j)$  can be represented with a linear combination of the above matrices  $C, D, E, F, N$  and  $\mathbf{u}^{n+1}$ .

To sum up,  $B_1$  in (2.18) can be obtained with pre-calculated elements  $M, C, D, E, F$  and  $N$ .

## References

- [1] U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, *Appl. Numer. Math.* 25 (1997) 151–167.
- [2] L. Bonaventura, R. Ferretti, Flux form semi-Lagrangian methods for parabolic problems, *Commun. Appl. Ind. Math.* 7 (2016) 56–73.
- [3] P.A. Bosler, A.M. Bradley, M.A. Taylor, Conservative multimoment transport along characteristics for discontinuous Galerkin methods, *SIAM J. Sci. Comput.* 41 (2019) B870–B902.
- [4] X. Cai, S. Boscarino, J.-M. Qiu, High order semi-Lagrangian discontinuous Galerkin method coupled with Runge-Kutta exponential integrators for non-linear Vlasov dynamics, preprint arXiv:1911.12229, 2019.
- [5] X. Cai, W. Guo, J.-M. Qiu, A high order conservative semi-Lagrangian discontinuous Galerkin method for two-dimensional transport simulations, *J. Sci. Comput.* 73 (2017) 514–542.
- [6] M. Calvo, J. De Frutos, J. Novo, Linearly implicit Runge-Kutta methods for advection–reaction–diffusion equations, *Appl. Numer. Math.* 37 (2001) 535–549.
- [7] M.A. Celia, T.F. Russell, I. Herrera, R.E. Ewing, An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation, *Adv. Water Resour.* 13 (1990) 187–206.
- [8] B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, *SIAM J. Numer. Anal.* 35 (1998) 2440–2463.
- [9] M. Diamantakis, The semi-Lagrangian technique in atmospheric modelling: current status and future challenges, in: *ECMWF Seminar in Numerical Methods for Atmosphere and Ocean Modelling*, 2013, pp. 183–200.
- [10] M. Falcone, R. Ferretti, *Semi-Lagrangian Approximation Schemes for Linear and Hamilton-Jacobi Equations*, vol. 133, SIAM, 2013.
- [11] F. Giraldo, J. Perot, P. Fischer, A spectral element semi-Lagrangian (SESL) method for the spherical shallow water equations, *J. Comput. Phys.* 190 (2003) 623–650.
- [12] M. Groppi, G. Russo, G. Stracquadanio, High order semi-Lagrangian methods for the BGK equation, *Commun. Math. Sci.* 14 (2016) 389–414.
- [13] W. Guo, R.D. Nair, J.-M. Qiu, A conservative semi-Lagrangian discontinuous Galerkin scheme on the cubed sphere, *Mon. Weather Rev.* 142 (2014) 457–475.
- [14] P.H. Lauritzen, R.D. Nair, P.A. Ullrich, A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid, *J. Comput. Phys.* 229 (2010) 1401–1424.
- [15] S.-J. Lin, R.B. Rood, Multidimensional flux-form semi-Lagrangian transport schemes, *Mon. Weather Rev.* 124 (1996) 2046–2070.
- [16] T.F. Russell, M.A. Celia, An overview of research on Eulerian–Lagrangian localized adjoint methods (ELLAM), *Adv. Water Resour.* 25 (2002) 1215–1231.
- [17] E. Sonnendrücker, J. Roche, P. Bertrand, A. Ghizzo, The semi-Lagrangian method for the numerical resolution of the Vlasov equation, *J. Comput. Phys.* 149 (1999) 201–220.

- [18] A. Staniforth, J. Côté, Semi-Lagrangian integration schemes for atmospheric models-a review, *Mon. Weather Rev.* 119 (1991) 2206–2223.
- [19] H. Wang, C.-W. Shu, Q. Zhang, Stability and error estimates of local discontinuous Galerkin methods with implicit-explicit time-marching for advection-diffusion problems, *SIAM J. Numer. Anal.* 53 (2015) 206–227.
- [20] G. Wanner, E. Hairer, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*, 1991.
- [21] D. Xiu, G.E. Karniadakis, A semi-Lagrangian high-order method for Navier–Stokes equations, *J. Comput. Phys.* 172 (2001) 658–684.