Global spectral analysis of two fully discrete discontinuous Galerkin methods on unfitted meshes

Hu Yu [1], Ziqiang Cheng [2], Yan Jiang [3 4]

Abstract

In this paper, we analyze the dispersion and dissipation properties of two fully discrete discontinuous Galerkin (DG) methods, namely the Runge-Kutta DG (RKDG) method and the Lax-Wendroff DG (LWDG) method, when solving the linear convection equation by employing the global spectral analysis (GSA). Under the condition that the convection velocity of the numerical solution is non-constant, we derive the GSA type numerical phase velocity and numerical group velocity through the analysis of the all Fourier waves. Considering that the boundary may be unfitted, we develop an inverse Lax-Wendroff (ILW) boundary treatment for LWDG scheme based on the previous work for RKDG scheme (Yang L, et al. Inverse Lax-Wendroff boundary treatment of discontinuous Galerkin method for 1D conservation laws. *Commun. Appl. Math. Comput.*, 7, 796–826 (2025)) and make conservation corrections. Finally, we analyze the dispersion properties of the two DG schemes employing the ILW boundary treatment near the boundary, comparing the impact of the boundary treatment. A series of numerical examples verify our theoretical results.

Keywords Global spectral analysis, fully discrete discontinuous Galerkin methods, inverse Lax-Wendroff method, boundary analysis

1 Introduction

The error dynamics and dispersion/dissipation properties of numerical solutions have always been an important research topic [23, 42, 49, 50]. In this paper, we will analyze the global dispersion and dissipation errors of fully discrete high order discontinuous Galerkin (DG) methods when solving the

[1]School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: yuhumath@mail.ustc.edu.cn.

[2]The School of Mathematics, Hefei University of Technology, Hefei, Anhui 230009, China. Email: czq10491@hfut.edu.cn.

[3]School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, China. E-mail: jiangy@ustc.edu.cn.

[4]Laoshan Laboratory, Qingdao 266237, China.

linear convection equation, coupling with the so-called inverse Lax-Wendroff (ILW) boundary treatment on unfitted meshes.

So far, most previous works on the dispersion/dissipation analysis are based on von Neumann analysis, which brought some limitations. Firstly, von Neumann analysis is only applicable to periodic problems, not boundary value problems. Secondly, von Neumann analysis considers the dispersion properties of a single Fourier mode, without considering the interaction between the various Fourier modes. The convection velocity is usually considered to be constant in von Neumann analysis. However, this assumption does not match the exact situation of numerical solutions [35]. Recently, a new analytical method called global spectral analysis (GSA) was proposed [35] and has been applied to a variety of equations and numerical schemes, including convection equations [35, 33, 36, 37], diffusion equations [33], convection-diffusion equations [30, 31], etc. Compared with the Fourier analysis, GSA analyzes discrete equations with general initial conditions on the spectral plane in the non-periodic region instead of a single Fourier wave, and derives non-constant convection velocities in the numerical solution. Thus, it can be used on initial-boundary value problems. For a review of GSA, please refer to [28].

In particular, for the linear convection equation $u_t + cu_x = 0$, the numerical dispersion relation is given as $\omega = c \cdot k_N$ where k_N is the numerical wavenumber in von Neumann analysis. The numerical group velocity is given as $v_{gN} = cdk_N/dk$. But these dispersion relations has been proven to be wrong[37, 36, 35, 28]. Under the assumption of GSA, the numerical convection velocity c_N is not a constant. Through the numerical dispersion relation $\omega_N = k \cdot c_N$, we can obtain the expression for the numerical group velocity as

$$v_{gN} = \frac{\mathrm{d}\omega_N}{\mathrm{d}k} = c_N + k\frac{\mathrm{d}c_N}{\mathrm{d}k}.$$

This expression differs from that obtained through von Neumann analysis.

In this paper, we concern on the dispersion/dissipation analysis of DG method. DG method was first used by Reed and Hill [27] in 1973 to solve the neutron transport equation. Later, Cockburn, Shu, and others [10, 9, 8, 11, 12] combined spatial DG discretization with explicit Runge-Kutta time discretization to develop the more practical Runge-Kutta DG (RKDG) method, which was applied to solve nonlinear hyperbolic conservation laws. The DG method is based on piecewise polynomial discontinuous approximations, where inschemeion is exchanged only through numerical fluxes between neighboring cells. Due to the local nature of the DG scheme, it is highly compact, easily handles various boundary conditions. An alternative approach could be using a Lax-Wendroff type time discretization procedure, which re-

lies on a Taylor expansion in time and converting all the time derivatives into spatial derivatives by repeatedly using the PDE and its differentiated versions. The Lax-Wendroff DG (LWDG) scheme was proposed by Qiu, Dumbser, and Shu [26, 25]. As one-step one-stage high-order numerical methods, LWDG methods demonstrate cost efficiency in some applications, such as the two-dimensional Euler equations in gas dynamics, when compared with one-step multi-stage RKDG methods. There has been a lot of work analyzing the dispersion and dissipative properties of semi-discrete DG scheme [38, 19, 1, 48, 16, 7] and fully-discrete DG scheme [46, 2, 3, 4]. However, all these works were concerned on the linear problems with periodic boundary condition, employing the Fourier analysis. The global spectral analysis for the DG scheme and the dispersion properties at the boundaries are still not given. In this paper, we will utilize the GSA to analyze the dissipation-dispersion errors of the RKDG scheme and the LWDG scheme. In particular, the boundary influence will be taken into account as well.

Although DG methods can use unstructured meshes for complex geometries, the small cell sizes can result in very small time steps due to stability restrictions. Moreover, in many cases (such as moving boundaries), generating a high-quality mesh can be expensive, making the use of a regular mesh a viable alternative. Since the mesh boundary and the computational domain boundary may not coincide, the so-called "cut-cells" may appear, imposing additional limitations on the allowable time steps. To address this issue, several efficient strategies have been developed for handling the "cut-cell" problem, including implicit time-stepping [5], cell merging or agglomeration [22, 24, 29], stabilization with ghost penalties [6, 13, 17, 18, 40, 41], state redistribution [15], and the shifted boundary method [39]. In this paper, we focus on the Inverse Lax-Wendroff (ILW) method, developed by Tan and Shu [43, 44], which was originally designed to match the finite difference method on Cartesian grids for solving hyperbolic conservation law equations. This method leverages the equations to convert normal derivatives into time derivatives and tangential derivatives, thereby constructing ghost points near inflow boundaries. The algorithm achieves arbitrary-order accuracy. To simplify the algebraic operations involved in the ILW process, Tan et al. [45] proposed a simplified ILW (SILW) method, which uses extrapolation to construct high-order normal derivatives. More recently, Yang et al. [47] incorporated ILW boundary treatment into RKDG methods, which proves beneficial for DG methods on unfitted meshes and complex boundary problems. This approach completely mitigates the issue of small time steps, allowing the method to maintain the same explicit time step as the standard DG method. However, they also found that the boundary treatment disrupts the local conservative property in ghost cells, leading to a significant

impact on error magnitudes. To address this, an additional post-processing technique is proposed to restore local conservation. In this paper, we extend the ILW boundary treatment to the LWDG method and provide a stability analysis to ensure the effectiveness of the scheme. Additionally, the boundary influences for RKDG methods and LWDG methods will be compared via GSA.

The remaining of this paper is organized as follows. In Sec. 2, the general forms of the RKDG method and LWDG method are introduced. In Sec. 3, the dispersion-dissipation properties of the two fully-discrete DG schemes for the one-dimensional linear advection equation under periodic boundary conditions are analyzed, and the dispersion-dissipation errors of the RKDG scheme and LWDG scheme are compared across different wavenumber ranges. Sec. 4 presents the ILW boundary treatment method for the matching LWDG scheme, and linear stability is analyzed through the visualizing the eigenspectrum method. In Sec. 5, the impact of the ILW boundary treatment method on the dispersion properties of the RKDG scheme and LWDG scheme are analyzed. Several numerical examples are provided to validate our theoretical analysis and findings will be given in Sec. 6. Finally, Sec. 7 concludes the paper and discusses future directions.

2 Fully Discrete DG Methods

In this section, we give a review on the DG methods to solve 1D scalar hyperbolic conservation law,

$$u_t + f(u)_x = 0, \quad x \in [a, b],\, t > 0. \tag{1}$$

Following the standard DG methods procedure [10], we first divide the spatial domain and define the corresponding DG solution space. Here, we will use a uniform grid. The computational domain is divided into N equally spaced cells,

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N+\frac{1}{2}} = b,$$

and each cell is denoted as $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ with mesh size $h = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ and the midpoint $x_j = \frac{1}{2}(x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}})$.

Let q be a non-negative integer, and $P^q(I_j)$ be the space of polynomials of degree at most q on I_j. We define the piecewise polynomials space

$$V_h^q = \{v : v|_{I_j} \in P^q(I_j),\, j = 1, 2, \cdots, N\}. \tag{2}$$

In this paper, we will consistently use the basis functions $\{\varphi_l^j(x) = \varphi_l(\frac{x-x_j}{h/2}),\, l = 0, \cdots, q,\, j = 1, \cdots, N\}$, where φ_l is the Legendre polynomial of degree

l defined on the reference interval $[-1,1]$. For any $v \in V_h^q$, we denote $v^-_{j+\frac{1}{2}}$ (resp. $v^+_{j+\frac{1}{2}}$) as the left (resp. right) limit of the discontinuous function v at the cell interface $x_{j+\frac{1}{2}}$.

Next we will introduce RKDG method and LWDG method, and both of them can be of arbitrary order of accuracy.

2.1 RKDG Schemes

The RKDG schemes is constructed in the method of lines framework, meaning that the spatial variable is first discretized, then the numerical solution is updated in time by coupling a suitable Runge-Kutta time discretization.

The semi-discrete DG method for (1) is: find the unique approximate solution $u_h(\cdot,t) \in V_h^q$, such that for any $v \in V_h^q$ and all $1 \le j \le N$ the following holds,

$$\int_{I_j} (u_h)_t v \,\mathrm{d}x = \int_{I_j} f(u_h) v_x \,\mathrm{d}x - \hat{f}_{j+\frac{1}{2}} v^-_{j+\frac{1}{2}} + \hat{f}_{j-\frac{1}{2}} v^+_{j-\frac{1}{2}}, \tag{3}$$

where $\hat{f}_{j+\frac{1}{2}} = \hat{f}((u_h)^-_{j+\frac{1}{2}}, (u_h)^+_{j+\frac{1}{2}})$ is a numerical flux at $x_{j+\frac{1}{2}}$. One of the most commonly used monotone fluxes is Lax-Friedrichs flux,

$$\hat{f}^{LF}(u^-,u^+) = \frac{1}{2}\left(f(u^-) + f(u^+) - \alpha(u^+ - u^-)\right), \quad \alpha = \max_u |f'(u)|.$$

What needs special attention is that the values of $\hat{f}_{\frac{1}{2}}$ and $\hat{f}_{N+\frac{1}{2}}$ depend on the specific boundary conditions settings.

Let $u_h(x,t)|_{I_j} = \sum_{l=0}^q u_l^j(t)\varphi_l^j(x)$, then the semi-discrete DG scheme (3) can be rewritten as the first-order ODE system

$$\boldsymbol{u}_t = L(\boldsymbol{u}),$$

where $\boldsymbol{u}(t) = (u_0^1(t), \cdots, u_q^1(t), \cdots, u_0^N(t), \cdots, u_q^N(t))^T$ is the coefficient vector to be determined, and the operator $L(\boldsymbol{u})$ is arised from the spatial discretization.

The third-order TVD Runge-Kutta method [11] is usually used for the time discretization. Let Δt be the time step, then we can update the coefficient vector from time t^n to $t^{n+1} = t^n + \Delta t$ as

$$\begin{aligned}
\boldsymbol{u}^{(1)} &= \boldsymbol{u}^n + \Delta t L(\boldsymbol{u}^n), \\
\boldsymbol{u}^{(2)} &= \frac{3}{4}\boldsymbol{u}^n + \frac{1}{4}(\boldsymbol{u}^{(1)} + \Delta t L(\boldsymbol{u}^{(1)})), \\
\boldsymbol{u}^{n+1} &= \frac{1}{3}\boldsymbol{u}^n + \frac{2}{3}(\boldsymbol{u}^{(2)} + \Delta t L(\boldsymbol{u}^{(2)})).
\end{aligned} \tag{4}$$

The scheme can give $(q+1)$-th order in space and third order in time. In particular, [10] has proved that the fully discrete scheme is stable under the CFL condition $\frac{\alpha\Delta t}{h} \leq \frac{1}{2q+1}$.

2.2 LWDG Schemes

The LWDG method in [26] starts with a Taylor expansion of order $q+1$ of the solution u in time,

$$u(x,t+\Delta t) \approx u(x,t) + \sum_{s=1}^{q+1} \frac{\Delta t^s}{s!} \partial_t^{(s)} u(x,t).$$

From $u_t + f(u)_x = 0$, it follows that $\partial_t^{(s)} u = -\partial_x \partial_t^{(s-1)} f(u)$, $s = 1, 2, \cdots, q+1$. So we get

$$u(x,t+\Delta t) \approx u(x,t) - \Delta t F_x$$

with

$$F(u) = \sum_{s=0}^{q} \frac{\Delta t^s}{(s+1)!} \partial_t^{(s)} f(u). \tag{5}$$

Furthermore, we can calculate the time derivatives in (5) with the space derivatives based on the original PDE (1). For example, for $q = 2$ we have

$$\begin{aligned} u_t =& -f(u)_x = -f' u_x, \\ f(u)_t =& f' u_t = -(f')^2 u_x, \\ u_{tt} =& -(f(u)_t)_x = \left((f')^2 u_x\right)_x, \\ f(u)_{tt} =& f'' u_t^2 + f' u_{tt} = 3f''(f')^2 (u_x)^2 + (f')^3 u_{xx}. \end{aligned}$$

Then we can obtain

$$F(u) = f(u) - \frac{\Delta t}{2}(f')^2 u_x + \frac{\Delta t^2}{6}\left(3f''(f')^2 (u_x)^2 + (f')^3 u_{xx}\right).$$

The LWDG method for (1) is: find the unique approximate solution $u_h(\cdot,t) \in V_h^q$, such that $\forall v \in V_h^q$ and $1 \leq j \leq N$ the following equation holds,

$$\int_{I_j} \frac{u_h(x,t+\Delta t) - u_h(x,t)}{\Delta t} v \, \mathrm{d}x = \int_{I_j} F(u_h) v_x \, \mathrm{d}x - \hat{F}_{j+\frac{1}{2}} v^-_{j+\frac{1}{2}} + \hat{F}_{j-\frac{1}{2}} v^+_{j-\frac{1}{2}}, \tag{6}$$

where $\hat{F}_{j+\frac{1}{2}}(u_h) = \hat{F}((u_h)^-_{j+\frac{1}{2}}, (u_h)^+_{j+\frac{1}{2}})$ is a numerical flux at $x_{j+\frac{1}{2}}$. The numerical flux given in [25] is based on the decompositions $F = f + (F - f)$, that

$$\hat{F}(u^-, u^+) = \hat{f}(u^-, u^+) + \frac{1}{2}\left(f^*(u^-) + f^*(u^+)\right), \quad f^* = F - f, \tag{7}$$

where $\hat{f}$ can be any commonly used numerical flux. The scheme can achive $(q+1)$-th order in space and time at the same time. In addition, for the case of $f(u) = cu$ $(c > 0)$, another form of numerical flux in convex combination is given in [46], where

$$\hat{F}(u^-, u^+) = \beta f(u^-) + (1-\beta) f(u^+) + \gamma f^*(u^-) + (1-\gamma) f^*(u^+) \tag{8}$$

with $\beta, \gamma \in [0, 1]$. For the linear convection equation, (8) is a more general numerical flux, and (7) is a special case of (8) when $\beta = 1$ and $\gamma = \frac{1}{2}$.

3 Dispersion Analysis by GSA

We start our discussion with the 1D linear advection equation:

$$\begin{cases} u_t + cu_x = 0, \quad c > 0, \quad x \in \mathbb{R}, \\ u(x, 0) = u_0(x). \end{cases} \tag{9}$$

The solution of the fully discrete DG method to solve (9) is given by

$$u_h^n(x)|_{I_j} = \sum_{l=0}^{q} u_l^{j,n} \varphi_l^j(x). \tag{10}$$

The initial values for the DG scheme are obtained from the L^2 projection, which is represented by the operator P_h, into the approximation space of the initial function. That is to say, $u_h(x, 0) = P_h(u_0(x))$, satisfying

$$\int_{I_j} u_h(x, 0) v(x) dx = \int_{I_j} u_0(x) v(x) dx, \quad \forall v \in V_h^q.$$

Assume that the initial function $u_0(x)$ can be expressed in the form of a Fourier transform $u_0(x) = \int_{\mathbb{R}} \hat{u}_0(k) e^{ikx} \, \mathrm{d}k$, then we can obtain

$$u_l^{j,0} = \frac{2l+1}{h} \int_{I_j} u_0(x) \varphi_l^j(x) \, \mathrm{d}x = \int_{\mathbb{R}} U_l^0 \hat{u}_0(k) e^{ikx_j} \, \mathrm{d}k, \tag{11}$$

where $U_l^0 = \frac{2l+1}{2} \int_{-1}^{1} e^{\frac{ikh\xi}{2}} \varphi_l(\xi) \, \mathrm{d}\xi$. Furthermore, we have

$$u_h^0(x)|_{I_j} = \int_{\mathbb{R}} \boldsymbol{U}^0 \cdot \boldsymbol{\varphi}^j(x) \, \hat{u}_0(k) e^{ikx_j} \, \mathrm{d}k. \tag{12}$$

The operation $(\cdot)$ refers to the inner product of vector $\boldsymbol{U}^0 = [U_0^0, \cdots, U_q^0]^T$ and $\boldsymbol{\varphi}^j(x) = [\varphi_0^j(x), \cdots, \varphi_q^j(x)]^T$.

We can assume that the numerical solution (10) at any time has a form similar to (12), that is

$$u_h^n(x)|_{I_j} = \int_{\mathbb{R}} \boldsymbol{U}^n \cdot \boldsymbol{\varphi}^j(x)\, \hat{u}_0(k) e^{ikx_j}\, \mathrm{d}k \tag{13}$$

with $\boldsymbol{U}^n = [U_0^n, \cdots, U_q^n]^T$, $n = 0, 1, \cdots$. Or equivalently, we assume that

$$u_l^{j,n} = \int_{\mathbb{R}} U_l^n \hat{u}_0(k) e^{ikx_j}\, \mathrm{d}k \tag{14}$$

Next, we will solve $\boldsymbol{U}^n$ by recursive relation $\boldsymbol{U}^{n+1} = \boldsymbol{G}\boldsymbol{U}^n$, and use numerical amplification factor $\boldsymbol{G}$ to analyze the phase velocity and group velocity of the scheme by GSA method.

3.1 Numerical Amplification Factor

For the DG methods to solve equation (9), we can use $\hat{f}_{j+\frac{1}{2}} = f((u_h)^-_{j+\frac{1}{2}})$ for RKDG method in (3). As for the LWDG method (6), we use the more general numerical flux (8) for linear scalar problems.

3.1.1 RKDG Schemes

Utilizing the forward Euler time discretization and substituting $u_h^n(x)|_{I_j} = \sum_{l=0}^q u_l^{j,n} \varphi_l^j(x)$ into (3), the scheme can be simplified to

$$u_l^{j,n+1} = u_l^{j,n} + \frac{c\Delta t}{h} \cdot (2l+1) \sum_{m=0}^{q} \left(\int_{-1}^{1} (\varphi_m \varphi_l'\, \mathrm{d}\xi - 1) u_m^{j,n} + (-1)^l u_m^{j-1,n} \right). \tag{15}$$

Set $N_c = \frac{c\Delta t}{h}$, taking (14) into (15) gives us

$$U_l^{n+1} = U_l^n + N_c \cdot (2l+1) \sum_{m=0}^{q} \left(\int_{-1}^{1} \varphi_m \varphi_l'\, \mathrm{d}\xi - 1 + (-1)^l e^{-ikh} \right) U_m^n. \tag{16}$$

Let

$$A_{lm} = (2l+1) \left(\int_{-1}^{1} \varphi_m \varphi_l'\, \mathrm{d}\xi + (-1)^l e^{-ikh} - 1 \right)$$

and take it to form a matrix $\boldsymbol{A}$. $\boldsymbol{I}$ is the identity matrix of size $q+1$, we can express (16) in matrix form,

$$\boldsymbol{U}^{n+1} = (\boldsymbol{I} + N_c \boldsymbol{A}) \boldsymbol{U}^n.$$

Furthermore, when the third-order TVD Runge-Kutta method (4) is used for the time discretization, we have

$$\boldsymbol{U}^{n+1} = \left(\boldsymbol{I} + N_c\boldsymbol{A} + \frac{N_c^2}{2}\boldsymbol{A}^2 + \frac{N_c^3}{6}\boldsymbol{A}^3\right)\boldsymbol{U}^n. \tag{17}$$

In order to match the accuracy of time and space, when $q = 1$,we use the second-order TVD Runge-Kutta method [11], so there is

$$\boldsymbol{U}^{n+1} = \left(\boldsymbol{I} + N_c\boldsymbol{A} + \frac{N_c^2}{2}\boldsymbol{A}^2\right)\boldsymbol{U}^n. \tag{18}$$

3.1.2 LWDG Schemes

Since $f(u) = cu$, the F in (5) can be easily obtained,

$$F(u) = c\sum_{s=0}^{q}\frac{(-c\Delta t)^s}{(s+1)!}\frac{\partial^s u}{\partial x^s}. \tag{19}$$

Similarly, for the LWDG scheme (6), we have

$$\begin{aligned} u_l^{j,n+1} =& u_l^{j,n} + N_c\cdot(2l+1)\sum_{m=0}^{q}\Big((b_{lm} + (-1)^l\tilde{b}_m - b_m)u_m^{j,n} - \tilde{b}_m u_m^{j+1,n} \\ &+ (-1)^l b_m u_m^{j-1,n}\Big). \end{aligned}$$

where

$$\begin{aligned} b_{lm} =& \sum_{s=0}^{q}\frac{(-2N_c)^s}{(s+1)!}\int_{-1}^{1}\frac{\mathrm{d}^s\varphi_m}{\mathrm{d}\xi^s}\frac{\mathrm{d}\varphi_l}{\mathrm{d}\xi}\mathrm{d}\xi, \\ \tilde{b}_m =& (1-\gamma)\sum_{s=0}^{q}\frac{(-2N_c)^s}{(s+1)!}\frac{\mathrm{d}^s\varphi_m}{\mathrm{d}\xi^s}|_{\xi=-1} + (\gamma-\beta)(-1)^m, \\ b_m =& \gamma\sum_{s=0}^{q}\frac{(-2N_c)^s}{(s+1)!}\frac{\mathrm{d}^s\varphi_m}{\mathrm{d}\xi^s}|_{\xi=1} - (\gamma-\beta). \end{aligned}$$

Define $B_{lm} = (2l+1)\left(b_{lm} + ((-1)^l - e^{ikh})(\tilde{b}_m + e^{-ikh}b_m)\right)$, we can get

$$\boldsymbol{U}^{n+1} = (\boldsymbol{I} + N_c\boldsymbol{B})\boldsymbol{U}^n. \tag{20}$$

3.1.3 Physical Mode

At this point, for the fully discrete DG schemes, we have obtained $\boldsymbol{U}^{n+1} = \boldsymbol{G}\boldsymbol{U}^n$, and

$$\boldsymbol{G} = \begin{cases} \boldsymbol{I} + N_c\boldsymbol{A} + \frac{N_c^2}{2}\boldsymbol{A}^2, & \text{for RKDG scheme with } q = 1, \\ \boldsymbol{I} + N_c\boldsymbol{A} + \frac{N_c^2}{2}\boldsymbol{A}^2 + \frac{N_c^3}{6}\boldsymbol{A}^3, & \text{for RKDG scheme with } q \geq 2, \\ \boldsymbol{I} + N_c\boldsymbol{B}, & \text{for LWDG scheme.} \end{cases} \tag{21}$$

Based on these results, firstly we can discuss the linear stability of the schemes. We seek the maximum value of N_c (CFL number) such that the spectral radius $\rho(\boldsymbol{G}) \leq 1$ holds for all $kh \in [0, \pi]$. The results of the second-order and third-order DG schemes are listed in Tab. 1. Here, $\beta = \gamma = 1$ are chosen in the LWDG scheme, due to the fact this combination provides the maximum CFL number. And in the following, we always use $\beta = \gamma = 1$

Tab. 1: CFL condition of fully discrete DG schemes

Scheme	$q = 1$		$q = 2$	
	RKDG2	LWDG2	RKDG3	LWDG3
CFL number	0.333	0.333	0.209	0.170

Apart from that, let the eigenvalues and corresponding eigenvectors of $\boldsymbol{G}$ be $\{(\lambda_l, \boldsymbol{r}_l),\ l = 0, 1, \cdots, q\}$, we can obtain the l-th eigen-solution of $\boldsymbol{U}^n$ $(n \geq 1)$ as

$$\boldsymbol{U}_l^n = \lambda_l^n \boldsymbol{r}_l. \tag{22}$$

Set $\boldsymbol{R} = [\boldsymbol{r}_0, \cdots, \boldsymbol{r}_q]$, and define $\boldsymbol{\mu} = \boldsymbol{R}^{-1}\boldsymbol{U}^0 = [\mu_0, \cdots, \mu_q]^T$. By combining (22), we get

$$\boldsymbol{U}^n = \boldsymbol{G}^n\boldsymbol{U}^0 = \sum_{l=0}^{q} \mu_l \boldsymbol{U}_l^n. \tag{23}$$

Substituting (23) into (13) gives us

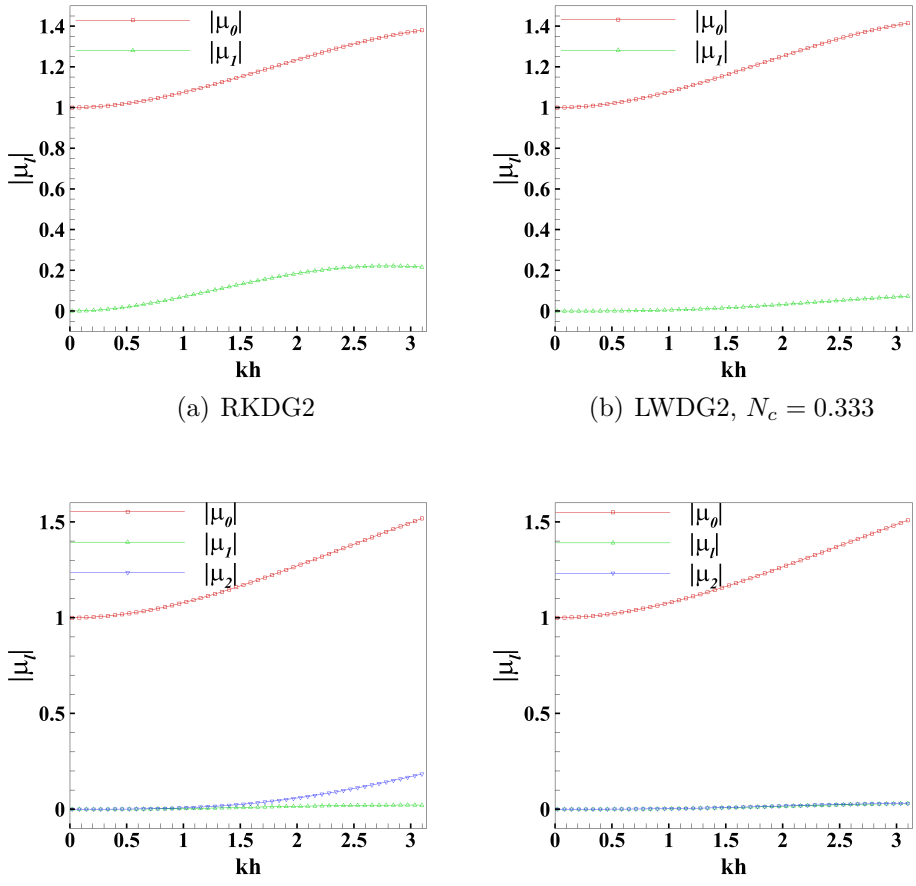
$$u_h^n(x)|_{I_j} = \sum_{l=0}^{q} \int_{\mathbb{R}} \mu_l\, \boldsymbol{U}_l^n \cdot \boldsymbol{\varphi}^j \hat{u}_0(k) e^{ikx_j}\, \mathrm{d}k. \tag{24}$$

That is, the fully discrete DG solution can be written as a combination of $q + 1$ the eigen-solutions, with the coefficients $\boldsymbol{\mu}$.

In general, $\boldsymbol{\mu}$ is a function with respect to kh and N_c. But for RKDG scheme, from the definitions of $\boldsymbol{A}$, $\boldsymbol{G}$ and $\boldsymbol{\mu}$, it can be seen that $\boldsymbol{\mu}$ is only a function of kh. This feature allows us to easily visualize the relative size of

the magnitude of each component of $\boldsymbol{\mu}$ for RKDG scheme. At this time, for LWDG scheme, we are forced to fix N_c as the CFL number.

Consider the magnitude of all components of $\boldsymbol{\mu}$. One of the values is close to 1, while the others are close to 0 (see Fig. 1). In most references, this dominant eigen-solution is referred to as the physical mode, while the others are called spurious modes. The former is what we are more concerned about.



(a) RKDG2

(b) LWDG2, $N_c = 0.333$

(c) RKDG3

(d) LWDG3, $N_c = 0.170$

Fig. 1: The magnitude of each component of $\boldsymbol{\mu}$

3.2 Phase Velocity and Group Velocity of Physical mode

The exact solution of (9) can be written as

$$u(x,t) = u_0(x-ct) = \int_{\mathbb{R}} \hat{u}_0(k) e^{i(kx-\omega t)}\,\mathrm{d}k \tag{25}$$

with $\omega = kc$. Similar to (12), the L^2 projection of the exact solution (25) into the approximation space is

$$P_h(u(x,t))|_{I_j} = \int_{\mathbb{R}} e^{-i\omega t}\boldsymbol{U}^0 \cdot \boldsymbol{\varphi}^j\, \hat{u}_0(k) e^{ikx_j}\,\mathrm{d}k. \tag{26}$$

We already know that when $h \to 0$, u_h approximates $P_h u$, and both approximate u.

Without loss of generality, let the 0-th eigen-solution be the physical mode. From $\boldsymbol{U}_0^n = \lambda_0^n \boldsymbol{r}_0 \sim e^{-i\omega t^n}\boldsymbol{U}^0$ as $kh \to 0$, we obtain $\lambda_0 \sim e^{-i\omega\Delta t}$ (this means $\lambda_0 - e^{-i\omega\Delta t} \to 0$ as $kh \to 0$). Let $\lambda_0 = e^{-i\omega_N \Delta t}$ and $\omega_N = \omega_{Nr} + i\omega_{Ni}$, we have $\omega_{Nr} \sim \omega$ and $\omega_{Ni} \sim 0$ as $kh \to 0$. In [46], the asymptotic formulations are derived analytically for the discrete dispersion relation in the limit of $kh \to 0$ as a function of N_c.

If we define $\lambda_0 = |\lambda_0| e^{-i\phi}$ again, then combining $\lambda_0 = e^{-i\omega_N \Delta t}$, we have

$$\begin{cases} |\lambda_0| = e^{\omega_{Ni}\Delta t}, \\ \phi = \omega_{Nr}\Delta t, \end{cases}$$

Following the idea of [35, 28], we define the numerical dispersion relation as $\omega_{Nr} = kc_N$. Thus, the phase velocity is $c_N = \frac{\omega_{Nr}}{k}$, and group velocity is

$$v_{gN} = \frac{\mathrm{d}\omega_{Nr}}{\mathrm{d}k} = c_N + k\frac{dc_N}{dk}. \tag{27}$$

Substituting $\omega_{Nr} = \frac{\phi}{\Delta t}$ ($\phi = \phi(N_c, kh)$) gives

$$\frac{c_N}{c} = \frac{\phi}{N_c \cdot kh}, \quad \frac{v_{gN}}{c} = \frac{1}{N_c}\frac{\mathrm{d}\phi}{\mathrm{d}kh}, \tag{28}$$

then low dissipation means $|\lambda_0| \sim 1$ and low dispersion means $c_N/c \sim 1$ and $v_{gN}/c \sim 1$.

Next, we will formally derive the error propagation equation, which is closely related to the dispersion and dissipation properties. The solution (24) can be written as

$$\begin{aligned} u_h(x,t^n)|_{I_j} = &\int_{\mathbb{R}} \mu_0 \,|\lambda_0|^{t^n/\Delta t} \boldsymbol{r}_0 \cdot \boldsymbol{\varphi}^j \hat{u}_0(k) e^{ik(x_j - c_N t^n)}\,\mathrm{d}k \\ &+ \sum_{l=1}^{q} \int_{\mathbb{R}} \mu_l \,|\lambda_l|^{t^n/\Delta t} \boldsymbol{r}_l \cdot \boldsymbol{\varphi}^j \hat{u}_0(k) e^{ik(x_j - c_N^l t^n)}\,\mathrm{d}k, \end{aligned}$$

where we have also defined a non-existent phase velocity c_N^l ($l \geq 1$) for the solutions of non-physical modes. Note that $\boldsymbol{\varphi}^j$ takes the same value at the cell midpoint x_j, and this constant vector is denoted as $\boldsymbol{\varphi}_0$. We get

$$\begin{aligned} u_h(x_j, t^n)|_{I_j} = &\int_{\mathbb{R}} \mu_0 \left|\lambda_0\right|^{t^n/\Delta t} \boldsymbol{r}_0 \cdot \boldsymbol{\varphi}_0 \hat{u}_0(k) e^{ik(x_j - c_N t^n)} \,\mathrm{d}k \\ &+ \sum_{l=1}^{q} \int_{\mathbb{R}} \mu_l \left|\lambda_l\right|^{t^n/\Delta t} \boldsymbol{r}_l \cdot \boldsymbol{\varphi}_0 \hat{u}_0(k) e^{ik(x_j - c_N^l t^n)} \,\mathrm{d}k. \end{aligned}$$

Map these discrete point values to a continuous function globally,

$$\begin{aligned} \tilde{u}(x,t) = &\int_{\mathbb{R}} \mu_0 \left|\lambda_0\right|^{t/\Delta t} \boldsymbol{r}_0 \cdot \boldsymbol{\varphi}_0 \hat{u}_0(k) e^{ik(x - c_N t)} \,\mathrm{d}k \\ &+ \sum_{l=1}^{q} \int_{\mathbb{R}} \mu_l \left|\lambda_l\right|^{t/\Delta t} \boldsymbol{r}_l \cdot \boldsymbol{\varphi}_0 \hat{u}_0(k) e^{ik(x - c_N^l t)} \,\mathrm{d}k \end{aligned} \tag{29}$$

then define the error as $e = \tilde{u} - u$, which is closely related to the dispersion error and dissipation errors. Combining the governing equation $u_t + cu_x = 0$ and (27), we can obtain the error propagation equation about specific wave number k', that is

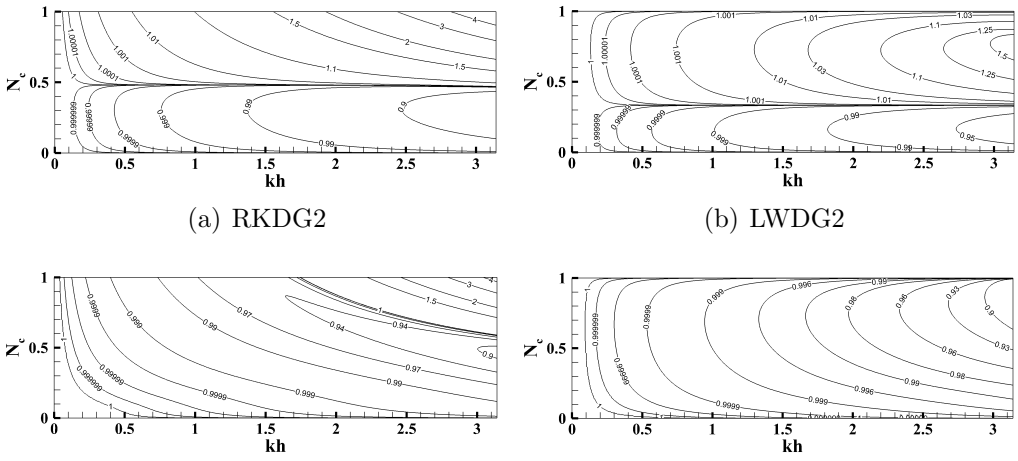
$$\begin{aligned} &e_t + ce_x \\ =&\tilde{u}_t + c_N(k')\tilde{u}_x + (c - c_N(k'))\tilde{u}_x \\ =&c(1 - \frac{c_N(k')}{c})\tilde{u}_x \\ &+ \int_{\mathbb{R}} \left(\tfrac{\ln|\lambda_0|}{\Delta t} + ik \int_k^{k'} \tfrac{v_{gN}(\tilde{k}) - c_N(\tilde{k})}{\tilde{k}} \,\mathrm{d}\tilde{k} \right) \mu_0 \left|\lambda_0\right|^{t/\Delta t} \boldsymbol{r}_0 \cdot \boldsymbol{\varphi}_0 \hat{u}_0(k) e^{ik(x - c_N(k)t)} \,\mathrm{d}k \\ &+ \sum_{l=1}^{q} \int_{\mathbb{R}} \left(\tfrac{\ln|\lambda_l|}{\Delta t} + ik(c_N(k') - c_N^l(k)) \right) \mu_l \left|\lambda_l\right|^{t/\Delta t} \boldsymbol{r}_l \cdot \boldsymbol{\varphi}_0 \hat{u}_0(k) e^{ik(x - c_N^l(k)t)} \,\mathrm{d}k. \end{aligned} \tag{30}$$

Eq. (30) is called as the *midpoint error propagation equation*. Noting the error to dramatically increase when the numerical solution displays sharp spatial variation. This can now be explained as the effect of the first term on the right-hand side of (30) [35]. In addition, we can see from Eq. (30) that the group velocity and phase velocity have a decisive influence on dispersion error propagation.

To more clearly describe the effect of the right-hand side of (30), we plot the contours of $|\lambda_0|$, c_N/c and v_{gN}/c with respect to N_c and kh. See Fig. 2, 3, and 4. In the region where kh approaches 0, we can obtain the following conclusions:

- Higher-order schemes have smaller dispersion and dissipation errors.
- For RKDG2 scheme and LWDG2 scheme, the dispersive error dominates, while for RKDG3 scheme and LWDG3 scheme, the dissipative error dominates.
- The RKDG2 scheme has larger dissipative errors than the LWDG2 scheme, but its dispersive errors are smaller.
- The RKDG3 scheme has smaller dispersive and dissipative errors compared to the LWDG3 scheme.

Some of the them are consistent with those in [46]. Besides, in the region where kh approaches π, we observe some different results, which are also supported by subsequent numerical experiments in Sec. 6.

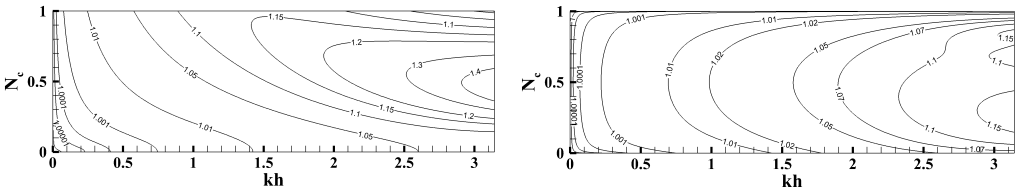


(a) RKDG2 (b) LWDG2

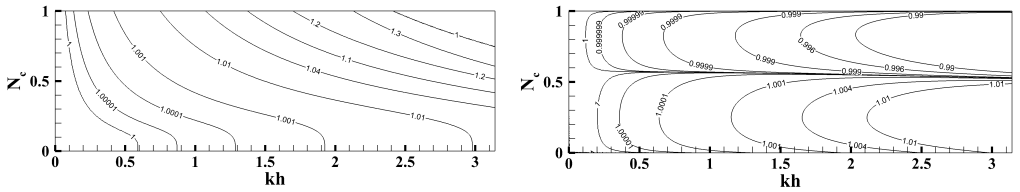
(c) RKDG3 (d) LWDG3

Fig. 2: The contour map of $|\lambda_0|$ with respect to N_c and kh.

In order to compare the properties of the two DG schemes more accurately, we fix a certain N_c and make the relevant one-dimensional graphs, see Figures 5, 6, and 7. It is observed in Fig. 5 (a) that, when $kh > 2.71$, the dissipation error of LWDG3 scheme will be smaller than that of RKDG3 scheme. This is contrary to the previous conclusion obtained from [46]. We have provided numerical experiments in Example 6.2 to support this, although we have specified a particular value of kh. From Fig. 7 (b), we can see that when $kh > 2.53$, the group velocity error of the RKDG2 scheme decreases, and when $kh > 2.94$, it becomes smaller than the group velocity

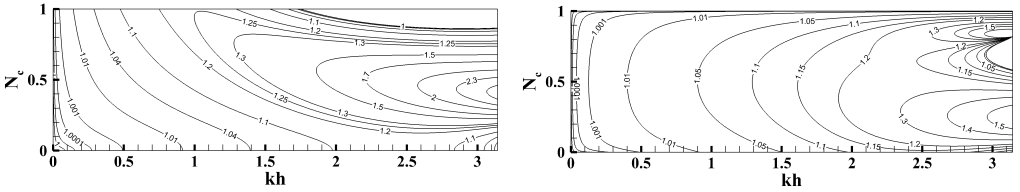


(a) RKDG2 (b) LWDG2

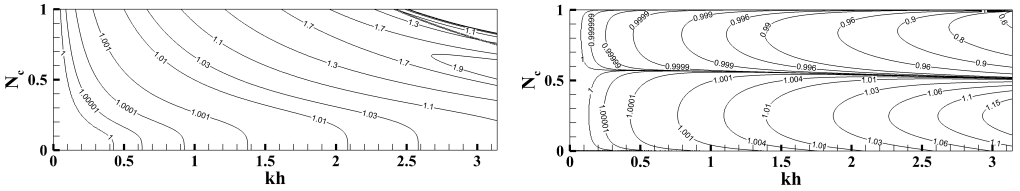


(c) RKDG3 (d) LWDG3

Fig. 3: The contour map of c_N/c with respect to N_c and kh.



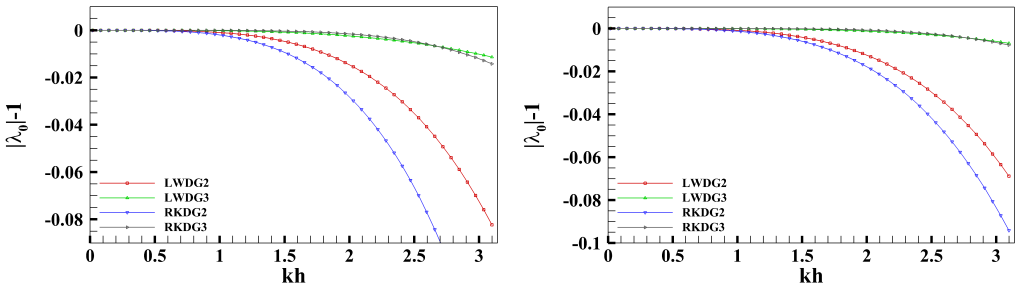
(a) RKDG2 (b) LWDG2



(c) RKDG3 (d) LWDG3

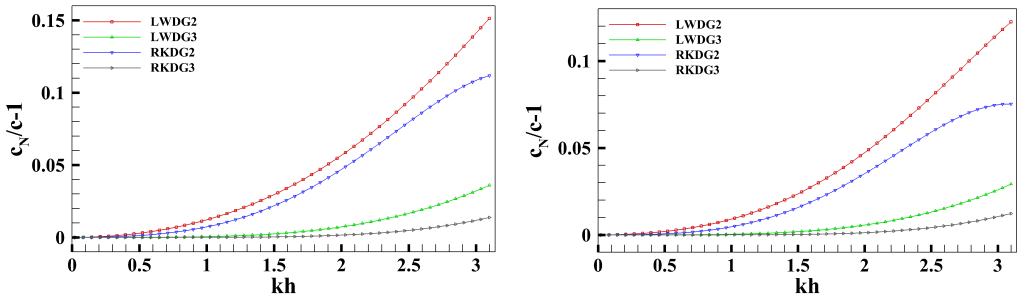
Fig. 4: The contour map of v_{gN}/c with respect to N_c and kh.

error of the LWDG3 scheme. This phenomenon can be observed in Example 6.3. It is worth noting that although these conclusions are drawn from specific N_c, they are valid for the entire stability domain.



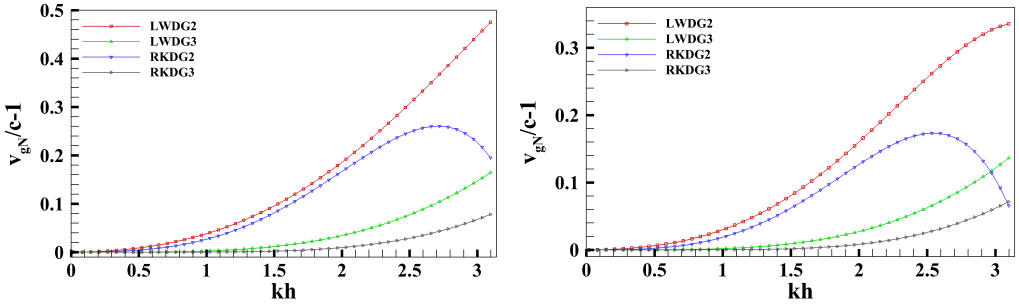
(a) $N_c = 0.16$ (b) $N_c = 0.1$

Fig. 5: A comparison of dissipation error with respect to kh within the stability region.



(a) $N_c = 0.16$ (b) $N_c = 0.1$

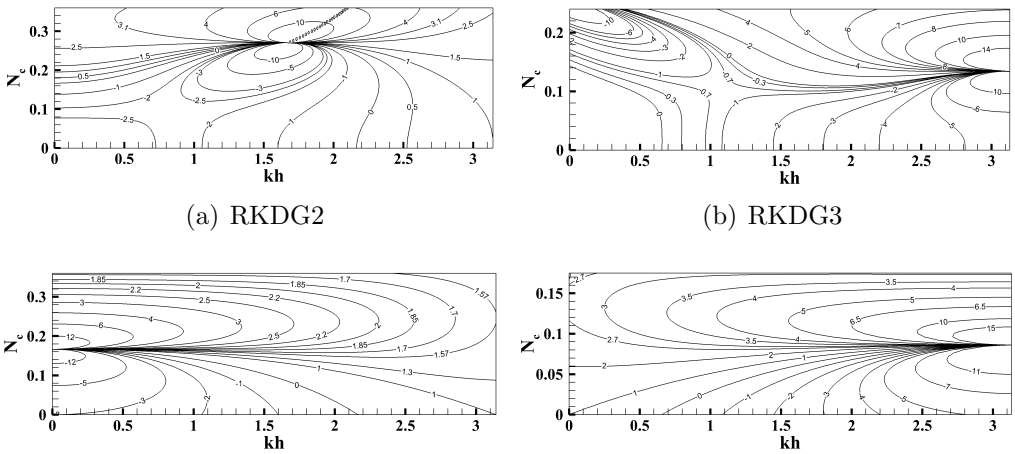
Fig. 6: A comparison of phase velocity error with respect to kh within the stability region.



(a) $N_c = 0.16$ (b) $N_c = 0.1$

Fig. 7: A comparison of group velocity error with respect to kh within the stability region.

3.3 q wave



(a) RKDG2 (b) RKDG3

(c) LWDG2 (d) LWDG3

Fig. 8: q wave: v_{gN}/c

4 ILW Methods for LWDG Schemes

In this section, we introduce the ILW method for LWDG, and the ILW method for RKDG has been studied in the previous work [47]. Here, we consider the following one-dimensional scalar conservation law:

$$\begin{cases} u_t + f(u)_x = 0, & x \in [a,b],\ t > 0, \\ u(x,0) = u_0(x), & x \in [a,b], \\ u(a,t) = g(t), & t > 0, \end{cases} \tag{31}$$

where $f'(u(a,t)) \geq \sigma > 0$ and $f'(u(b,t)) > 0$ for all $t > 0$.

To introduce our methods, we assume that the physical boundary is allowed to not coincide with grid points, that there are two cut cells, $\tilde{I}_0 = [a, a+\delta_1]$ and $\tilde{I}_{N+1} = [b-\delta_2, b]$, between the physical domain and the computational domain. Thus, our uniform grid subdivision is given as

$$a + \delta_1 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \cdots < x_{N+\frac{1}{2}} = b - \delta_2.$$

Here $h = (b-a)/(N + \delta_1/h + \delta_2/h)$ and $\delta_{1,2}/h \in [0,1)$ are predetermined.

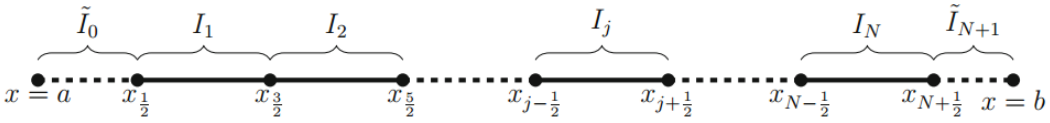


Fig. 9: Schematic diagram of unfitted mesh discretization.

When solving the problem (31), the DG method in applied on I_j, $j = 1, \cdots, N$. In particular, when using the LWDG method (6), the numerical fluxes $\hat{F}_{\frac{1}{2}}$ and $\hat{F}_{N+\frac{1}{2}}$ require special handling. Notice that the right boundary $x = b$ is an outflow boundary where no boundary condition is needed, thus we can take $\hat{F}_{N+\frac{1}{2}} = F((u_h)^-_{N+\frac{1}{2}})$ straightforwardly.

The left boundary $x = a$ is an inflow boundary. We will construct $\hat{F}_{\frac{1}{2}}$ using the ILW methods to ensure that the scheme is high-order, stable, and consist with the given boundary condition. Moreover, we want the scheme can employ the time step as the standard DG method with periodic boundary conditions, that is, the Δt is independent of $\delta_{1,2}$.

As we related in the introduction section. In [47], the ILW boundary treatment method and its simplified version were developed for the RKDG methods. Here, we will follow their ideas and develop the corresponding boundary treatment for the fully discrete LWDG schemes. The slight difficulty is that the RKDG methods have a semi-discrete scheme to use, and we have to work directly on the fully discrete scheme.

4.1 (S)ILW Methods

In the original ILW boundary treatment method, we will construct a polynomial $p(x,t)$ of degree q on $\tilde{I}_0$, such that it satisfies

$$\partial_x^{(m)} p(a,t) = \partial_x^{(m)} u(a,t), \quad m = 0, \cdots, q, \tag{32}$$

and then take $\hat{F}_{\frac{1}{2}} = \hat{F}(p(x_{\frac{1}{2}},t), (u_h)^+_{\frac{1}{2}})$. In particular, we can obtain the spatial derivatives of u at the boundary $x = a$ via the ILW procedure, by repeatedly using the PDE and boundary conditions to convert spatial derivatives into temporal derivatives. For example, for $q = 2$, we have

$$\begin{aligned} u|_{x=a} =& g(t), \\ u_x|_{x=a} =& -\frac{g'(t)}{f'(g(t))}, \\ u_{xx}|_{x=a} =& \frac{f'(g(t))g''(t) - 2f''(g(t))g'(t)^2}{f'(g(t))^3}. \end{aligned} \tag{33}$$

Denote the first $q+1$ terms of the Taylor expansion of u at $x=a$ as

$$\Pi_a^q[u](x,t)=\sum_{m=0}^{q}\frac{(x-a)^m}{m!}\partial_x^{(m)}u(a,t).$$

Thus, we can take

$$p(x,t)=\Pi_a^q[u](x,t). \tag{34}$$

It is observed that the algebraic calculations of the higher-order spatial derivatives in (33) are quite tedious. To simplify the algorithm, we can replace restriction on the the high-order derivatives at boundary in (32) by the integral average of lower-order derivatives of u_h over neighbor cell I_1. Denote

$$\bar{u}_1(t)=\frac{1}{h}\int_{I_1}u_h(x,t)\,\mathrm{d}x,\quad (\bar{u}_x)_1(t)=\frac{1}{h}\int_{I_1}(u_h)_x(x,t)\,\mathrm{d}x.$$

If only $\bar{u}_1(t)$ is used, we refer to it as the SILW1 method ($q\geq 1$). That is, $p(x,t)$ satisfies

$$\begin{cases}\partial_x^{(m)}p(a,t)=\partial_x^{(m)}u(a,t), & m=0,\cdots,q-1,\\ \frac{1}{h}\int_{I_1}p(x,t)\,\mathrm{d}x=\bar{u}_1(t).\end{cases}$$

Then, we can get

$$p(x,t)=\Pi_a^{q-1}[u(x,t)](x)+R(t)(x-a)^q, \tag{35}$$

where

$$R(t)=\frac{h\bar{u}_1(t)-\int_{I_1}\Pi_a^{q-1}[u(x,t)](x)\,\mathrm{d}x}{((\delta_1+h)^{q+1}-\delta_1^{q+1})/(q+1)}.$$

If $\bar{u}_1(t)$ and $(\bar{u}_x)_1(t)$ are both used, we refer to it as the SILW2 method ($q\geq 2$). At this point, $p(x,t)$ will satisfy

$$\begin{cases}\partial_x^{(m)}p(a,t)=\partial_x^{(m)}u(a,t), & m=0,\cdots,q-2,\\ \frac{1}{h}\int_{I_1}p(x,t)\,\mathrm{d}x=\bar{u}_1(t),\\ \frac{1}{h}\int_{I_1}p_x(x,t)\,\mathrm{d}x=(\bar{u}_x)_1(t).\end{cases}$$

In this case, we have

$$p(x,t)=\Pi_a^{q-2}[u(x,t)](x)+R(t)(x-a)^{q-1}+S(t)(x-a)^q, \tag{36}$$

where $C(q) = \frac{(\delta_1+h)^q-\delta_1^q}{q}$ and

$$\begin{bmatrix} R(t) \\ S(t) \end{bmatrix} = \frac{1}{qC(q)^2 - (q-1)C(q-1)C(q+1)} \begin{bmatrix} qC(q) & -C(q+1) \\ -(q-1)C(q-1) & C(q) \end{bmatrix} \begin{bmatrix} h\bar{u}_1(t) - \int_{I_1} \Pi_a^{q-2}[u(x,t)](x)\,\mathrm{d}x \\ h(\bar{u}_x)_1(t) - \Pi_a^{q-2}[u(x,t)](x)|_{I_1} \end{bmatrix}.$$

By analogy, methods utilizing more integral averages can be developed. However, it should be noted that excessive use of integral averages may lead to instability. In such cases, the use of boundary information should be increased.

Numerical experiments show that the schemes can always achieve the $(q+1)$-th order accuracy. However, the error heavily depends on the parameter δ_1/h. To describe the dependency of δ_1/h on the error more intuitively, we have plotted in Fig. 10 with the suffix "no cv". Here, we set $N = 80$ with other settings consistent with Example 6.4. To overcome this disadvantage, we propose a post-processing based on the local conservation in the following subsection. From Tab. 2, Tab. 3 and Fig. 10, we can see that this post-processing is highly effective.

4.2 Conservation Correction

In this subsection, we present the specific details for post-processing based on the local conservation property. The LWDG scheme is a conservative scheme, meaning for all $1 \le j \le N$, it satisfies

$$\int_{I_j} \frac{u_h(x,t+\Delta t) - u_h(x,t)}{\Delta t}\,\mathrm{d}x = \hat{F}_{j-\frac{1}{2}} - \hat{F}_{j+\frac{1}{2}}. \tag{37}$$

This can be obtained by setting $v = 1$ in Eq. (6). However, for the pre-constructed polynomial $p(x,t)$ in Sec. 4.1, it generally does not satisfy local conservation on $\tilde{I}_0$. That is,

$$\int_{\tilde{I}_0} \frac{p(x,t+\Delta t) - p(x,t)}{\Delta t}\,\mathrm{d}x \neq F|_{x=a} - \hat{F}_{\frac{1}{2}} \tag{38}$$

where $\hat{F}_{\frac{1}{2}} = \hat{F}(p(x_{\frac{1}{2}},t),(u_h)^+_{\frac{1}{2}})$ and $F|_{x=a} = \sum_{s=0}^{q} \frac{\Delta t^s}{(s+1)!}\frac{\mathrm{d}^s}{\mathrm{d}t^s} f(g(t))$.

In order to ensure $p(x,t)$ satisfying the local conservation property, we modify $\hat{F}_{\frac{1}{2}}$ to make the equality in (38) hold, specifically setting

$$\hat{F}^c_{\frac{1}{2}} = F|_{x=a} - \int_{\tilde{I}_0} \frac{p(x,t+\Delta t) - p(x,t)}{\Delta t}\,\mathrm{d}x. \tag{39}$$

For the ILW method, directly substituting (34) into (39) yields

$$\hat{F}^c_{\frac{1}{2}} = F|_{x=a} - \sum_{m=0}^{q} \frac{\delta_1^{m+1}}{(m+1)!} \frac{\partial_x^{(m)} u(a,t+\Delta t) - \partial_x^{(m)} u(a,t)}{\Delta t}. \tag{40}$$

For the SILW1 method and SILW2 method, the situation is slightly more complex. The computational details are provided in the appendix A. Here, we only list the final results. For the SILW1 method, let $\tilde{C}(q) = \frac{\delta_1^{q+1}}{(\delta_1+h)^{q+1}-\delta_1^{q+1}}$, then

$$\begin{aligned}\hat{F}^c_{\frac{1}{2}} = &\frac{1}{1+\tilde{C}(q)} \Bigg(F|_{x=a} + \tilde{C}(q)\hat{F}_{\frac{3}{2}} \\ &+ \sum_{m=0}^{q-1} \frac{\tilde{C}(q)((\delta_1+h)^{m+1} - \delta_1^{m+1}) - \delta_1^{m+1}}{(m+1)!} \frac{\partial_x^{(m)} u(a,t+\Delta t) - \partial_x^{(m)} u(a,t)}{\Delta t} \Bigg).\end{aligned} \tag{41}$$

For the SILW2 method, let

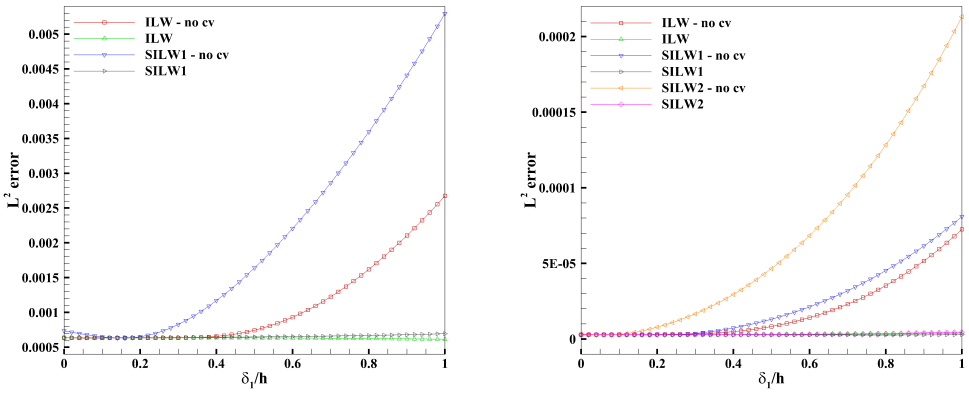
$$\begin{aligned}\begin{bmatrix} C_1(q) & C_2(q) \end{bmatrix} = &\frac{1}{qC(q)^2 - (q-1)C(q-1)C(q+1)} \\ &\begin{bmatrix} \frac{\delta_1^q}{q} & \frac{\delta_1^{q+1}}{q+1} \end{bmatrix} \begin{bmatrix} qC(q) & -C(q+1) \\ -(q-1)C(q-1) & C(q) \end{bmatrix}.\end{aligned}$$

Then for $q = 2$,

$$\begin{aligned}\hat{F}^c_{\frac{1}{2}} = &\frac{1}{1 + C_1(2) - \frac{6C_2(2)}{h}} \Bigg(F|_{x=a} + (C_1(2) + \frac{6C_2(2)}{h})\hat{F}_{\frac{3}{2}} - \frac{6C_2(2)}{h} \int_{-1}^{1} F(u_h)\varphi_1'(\xi)\,\mathrm{d}\xi \\ &+ (C_1(2)h - \delta_1) \frac{u(a,t+\Delta t) - u(a,t)}{\Delta t} \Bigg),\end{aligned} \tag{42}$$

and for $q = 3, 4$,

$$\begin{aligned}\hat{F}^c_{\frac{1}{2}} = &\frac{1}{1 + C_1(q) - \frac{20C_2(q)}{h}} \Bigg(F|_{x=a} + (C_1(q) + \frac{20C_2(q)}{h})\hat{F}_{\frac{3}{2}} \\ &- \frac{C_2(q)}{h} \left(6\int_{-1}^{1} F(u_h)\varphi_1'(\xi)\,\mathrm{d}\xi + 14\int_{-1}^{1} F(u_h)\varphi_3'(\xi)\,\mathrm{d}\xi \right) \\ &+ \sum_{m=0}^{q-2} \frac{C_1(q)((\delta_1+h)^{m+1} - \delta_1^{m+1}) + C_2(q)((\delta_1+h)^m - \delta_1^m)(m+1) - \delta_1^{m+1}}{(m+1)!} \\ &\frac{\partial_x^{(m)} u(a,t+\Delta t) - \partial_x^{(m)} u(a,t)}{\Delta t} \Bigg).\end{aligned} \tag{43}$$



(a) LWDG2, $N_c = 0.333$ (b) LWDG3, $N_c = 0.170$

Fig. 10: L^2 error for $\delta_1/h \in [0, 1)$

4.3 Stability Analysis

We use the eigenvalue spectrum visualization to study the linear stability of the schemes. As before, we take $f(u) = cu$ (with $c > 0$) and $g(t) = 0$ in (31), and use the upwind flux. Therefore, we can continue using the notations from Sec. 3.1.2.

For $j \geq 2$, we have

$$u_l^{j,n+1} = u_l^{j,n} + N_c \cdot (2l+1) \sum_{m=0}^{q} \left((b_{lm} - b_m) u_m^{j,n} + (-1)^l b_m u_m^{j-1,n} \right). \tag{44}$$

On the cell I_1, the flux $\hat{F}_{\frac{1}{2}}$ is related to the boundary treatment method. Therefore, for $j = 1$, we obtain

$$u_l^{j,n+1} = u_l^{j,n} + N_c \cdot (2l+1) \sum_{m=0}^{q} \left((b_{lm} - b_m + (-1)^l \hat{b}_m) u_m^{j,n} \right), \tag{45}$$

where,

$$\hat{b}_m = \begin{cases} 0, & \text{for ILW,} \\ b_m (1 - \frac{1}{1+\frac{\delta_1}{h}})^{q+1}, & \text{for SILW1,} \\ \frac{1}{1+C_1(2)-\frac{6C_2(2)}{h}} \left(C_1(2) - \frac{6C_2(2)}{h} (b_{1m} - b_m) \right), & \text{for SILW2.} \end{cases} \tag{46}$$

Let $\boldsymbol{u}^{j,n} = [u_0^{j,n}, \cdots, u_q^{j,n}]^T$, and $\boldsymbol{u}^n = [\boldsymbol{u}^{1,n}, \cdots, \boldsymbol{u}^{N,n}]^T$. Denote

$$(D_1)_{lm} = (2l+1)(b_{lm} - b_m), \quad (D_2)_{lm} = (2l+1)(-1)^l b_m$$
$$(\hat{D}_2)_{lm} = (2l+1)(-1)^l \hat{b}_m.$$

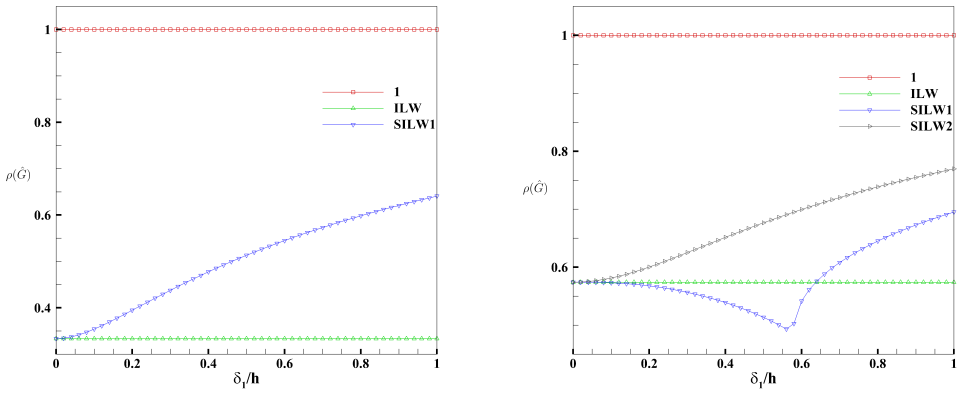
Then, (44) and (45) can be rewritten in matrix-vector form,

$$\begin{aligned} \boldsymbol{u}^{j,n+1} &= (\boldsymbol{I} + N_c \boldsymbol{D_1})\boldsymbol{u}^{j,n} + N_c \boldsymbol{D}_2 \boldsymbol{u}^{j-1,n}, \quad j \geq 2 \\ \boldsymbol{u}^{1,n+1} &= (\boldsymbol{I} + N_c(\boldsymbol{D}_1 + \hat{\boldsymbol{D}}_2))\boldsymbol{u}^{1,n}. \end{aligned} \tag{47}$$

Thus, we can obtain $\boldsymbol{u}^{n+1} = \hat{\boldsymbol{G}}\boldsymbol{u}^n$, where

$$\hat{\boldsymbol{G}} = \begin{pmatrix} \boldsymbol{I} + N_c(\boldsymbol{D}_1 + \hat{\boldsymbol{D}}_2) & & & & \\ N_c \boldsymbol{D}_2 & \boldsymbol{I} + N_c \boldsymbol{D_1} & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & N_c \boldsymbol{D}_2 & \boldsymbol{I} + N_c \boldsymbol{D_1} \end{pmatrix}.$$

The eigenvalues of $\hat{\boldsymbol{G}}$ are composed of the eigenvalues of $\boldsymbol{I} + N_c(\boldsymbol{D}_1 + \hat{\boldsymbol{D}}_2)$ and $\boldsymbol{I} + N_c \boldsymbol{D_1}$, and the latter are independent of δ_1/h. Therefore, we only concern on the spectral radius $\rho(\boldsymbol{I} + N_c(\boldsymbol{D}_1 + \hat{\boldsymbol{D}}_2)) \leq 1$ for all $\delta_1/h \in [0, 1)$, see Fig. 11. For all $\delta_1/h \in [0, 1)$, the spectral radius $\rho(\hat{\boldsymbol{G}})$ does not exceed 1, so the scheme is stable.



(a) LWDG2, $N_c = 0.333$ (b) LWDG3, $N_c = 0.170$

Fig. 11: Linear stability of the LWDG schemes combined with the ILW boundary treatment method.

5 Boundary Analysis by GSA

In this section, we analyze the impact of the boundary treatment methods presented in Sec. 4 on the dispersion-dissipation properties of the scheme, using the methods introduced in Sec. 3. To simplify the analysis, let the boundary value $g(t) = 0$. This requires the initial value $u_0(x)$ to have a compact support on the interval $[a, b]$. In summary, we will analyze the following IBVP,

$$\begin{cases} u_t + cu_x = 0, \quad c > 0,\ x \in [a, b],\ t > 0, \\ u(x, 0) = u_0(x), \quad x \in [a, b], \\ u(a, t) = 0, \quad t > 0. \end{cases} \tag{48}$$

We use the RKDG3 method and the LWDG3 method to solve this problem, with the boundary treatment methods SILW1 and SILW2. We still assume that the solution in the form of (13), but due to the influence of the boundary treatment, the value of $\boldsymbol{U}^n$ on each cell will be different. Thus, we assume

$$u_h^n(x)|_{I_j} = \int_{\mathbb{R}} \boldsymbol{U}^{j,n} \cdot \boldsymbol{\varphi}^j \hat{u}_0(k) e^{ikx_j} \, \mathrm{d}k. \tag{49}$$

5.1 LWDG Schemes

We first discuss the LWDG3 method. Let $(B_1)_{lm} = (2l + 1)(b_{lm} - b_m)$ and $(B_2)_{lm} = (2l + 1)(-1)^l b_m e^{-ikh}$ (b_{lm} and b_m are given in Sec. 3.1). For any $j \geq 2$, we have

$$\boldsymbol{U}^{j,n+1} = \boldsymbol{G}_1 \boldsymbol{U}^{j,n} + \boldsymbol{G}_2 \boldsymbol{U}^{j-1,n} \tag{50}$$

where $\boldsymbol{G}_1 = \boldsymbol{I} + N_c \boldsymbol{B}_1$ and $\boldsymbol{G}_2 = N_c \boldsymbol{B}_2$.

For the homogenous boundary condition $u(a, t) = 0$, the numerical flux $\hat{F}_{\frac{1}{2}}$ obtained by the SILW1 method with conservative correction is

$$\hat{F}^c_{\frac{1}{2}} = (1 - \frac{1}{1 + \frac{\delta_1}{h}})^3 \hat{F}_{\frac{3}{2}},$$

while that obtained by the SILW2 method is

$$\hat{F}^c_{\frac{1}{2}} = \frac{1}{1 + \tilde{C}_1 + \tilde{C}_2} \left((\tilde{C}_1 - \tilde{C}_2) \hat{F}_{\frac{3}{2}} + \tilde{C}_2 \int_{-1}^{1} F(u_h) \varphi_1'(\xi) \, \mathrm{d}\xi \right),$$

where

$$\begin{bmatrix} \tilde{C}_1 & \tilde{C}_2 \end{bmatrix} = \frac{(\frac{\delta_1}{h})^2}{(\frac{\delta_1}{h})^2 + \frac{\delta_1}{h} + \frac{1}{6}} \begin{bmatrix} \frac{2}{3}\frac{\delta_1}{h} + \frac{1}{2} & (\frac{\delta_1}{h} + 1)^2 \end{bmatrix}.$$

Let $\hat{\boldsymbol{G}}_2 = N_c\hat{\boldsymbol{B}}_2$ and $\tilde{\boldsymbol{G}} = \boldsymbol{G}_1 + \hat{\boldsymbol{G}}_2$, with

$$(\hat{B}_2)_{lm} = \begin{cases} (2l+1)(-1)^l b_m(1-\frac{1}{1+\frac{\delta_1}{h}})^3, & \text{for SILW1,} \\ \frac{(2l+1)(-1)^l}{1+\tilde{C}_1+\tilde{C}_2}\left(\tilde{C}_1+\tilde{C}_2(b_{1m}-b_m)\right), & \text{for SILW2.} \end{cases}$$

Then we get

$$\boldsymbol{U}^{1,n+1} = \tilde{\boldsymbol{G}}\boldsymbol{U}^{1,n}. \tag{51}$$

Since $\boldsymbol{U}^{j,0} = \boldsymbol{U}^0$, $\forall j \geq 1$, the recurrence relations (50) and (51) lead to

$$\begin{aligned} \boldsymbol{U}^{1,n} =& \tilde{\boldsymbol{G}}^n\boldsymbol{U}^0, \\ \boldsymbol{U}^{j,n} =& \left(\sum_{l=0}^{j-2}\binom{n}{l}\boldsymbol{G}_1^{n-l}\boldsymbol{G}_2^l + \boldsymbol{G}_2^{j-1}\sum_{l=0}^{n-(j-1)}\binom{l+j-2}{j-2}\boldsymbol{G}_1^l\tilde{\boldsymbol{G}}^{n-(j-1)-l}\right)\boldsymbol{U}^0, \ \forall j \geq 2, \end{aligned} \tag{52}$$

where $\binom{n}{l}$ is the binomial coefficient.

5.2 RKDG Schemes

A similar discussion can be applied to the RKDG3 scheme as well. Specifically, let $(A_1)_{lm} = (2l+1)\left(\int_{-1}^{1}\varphi_m\varphi_l'\,\mathrm{d}\xi - 1\right)$ and $(A_2)_{lm} = (2l+1)(-1)^l e^{-ikh}$. Furthermore, define $\boldsymbol{G}_1 = \boldsymbol{I} + N_c\boldsymbol{A}_1$ and $\boldsymbol{G}_2 = N_c\boldsymbol{A}_2$. Then, for any $j \geq 2$, we can obtain that

$$\begin{aligned} \boldsymbol{U}^{j,(1)} &= \boldsymbol{G}_1\boldsymbol{U}^{j,n} + \boldsymbol{G}_2\boldsymbol{U}^{j-1,n}, \\ \boldsymbol{U}^{j,(2)} &= \frac{3}{4}\boldsymbol{U}^{j,n} + \frac{1}{4}\left(\boldsymbol{G}_1\boldsymbol{U}^{j,(1)} + \boldsymbol{G}_2\boldsymbol{U}^{j-1,(1)}\right), \\ \boldsymbol{U}^{j,n+1} &= \frac{1}{3}\boldsymbol{U}^{j,n} + \frac{2}{3}\left(\boldsymbol{G}_1\boldsymbol{U}^{j,(2)} + \boldsymbol{G}_2\boldsymbol{U}^{j-1,(2)}\right). \end{aligned} \tag{53}$$

Especially, it can be obtained that, for all $j \geq 4$,

$$\begin{aligned} \boldsymbol{U}^{j,n+1} =& (\frac{1}{3}\boldsymbol{I} + \frac{1}{2}\boldsymbol{G}_1 + \frac{1}{6}\boldsymbol{G}_1^3)\boldsymbol{U}^{j,n} + (\frac{1}{2}\boldsymbol{G}_1^2\boldsymbol{G}_2 + \frac{1}{2}\boldsymbol{G}_2)\boldsymbol{U}^{j-1,n} \\ &+ \frac{1}{2}\boldsymbol{G}_1\boldsymbol{G}_2^2\boldsymbol{U}^{j-2,n} + \frac{1}{6}\boldsymbol{G}_2^3\boldsymbol{U}^{j-3,n}. \end{aligned} \tag{54}$$

We also can derive $\boldsymbol{U}^{2,n}$ and $\boldsymbol{U}^{3,n}$ from (53), but $\boldsymbol{U}^{1,n}$ needs to be computed first.

For the RKDG scheme, the numerical flux $\hat{f}^c_{\frac{1}{2}}$ obtained by the SILW1 method is

$$\hat{f}^c_{\frac{1}{2}} = (1 - \frac{1}{1+\frac{\delta_1}{h}})^3\hat{f}_{\frac{3}{2}},$$

while that obtained by the SILW2 method is

$$\hat{f}^c_{\frac{1}{2}} = \frac{1}{1+\tilde{C}_1+\tilde{C}_2}\left((\tilde{C}_1-\tilde{C}_2)\hat{f}_{\frac{3}{2}} + \tilde{C}_2\int_{-1}^{1} f(u_h)\varphi_1'(\xi)\,\mathrm{d}\xi\right).$$

Hence, let $\hat{\boldsymbol{G}}_2 = N_c\hat{\boldsymbol{A}}_2$, we can get

$$\boldsymbol{U}^{1,n+1} = (\frac{1}{3}\boldsymbol{I} + \frac{1}{2}(\boldsymbol{G}_1+\hat{\boldsymbol{G}}_2) + \frac{1}{6}(\boldsymbol{G}_1+\hat{\boldsymbol{G}}_2)^3)\boldsymbol{U}^{1,n}. \tag{55}$$

where

$$(\hat{A}_2)_{lm} = \begin{cases} (2l+1)(-1)^l(1-\frac{1}{1+\frac{\delta_1}{h}})^3, & \text{for SILW1} \\ \frac{(2l+1)(-1)^l}{1+\tilde{C}_1+\tilde{C}_2}\left(\tilde{C}_1+\tilde{C}_2(\int_{-1}^{1}\varphi_m\varphi_1'\,\mathrm{d}\xi-1)\right), & \text{for SILW2.} \end{cases}$$

To continue our discussion, we introduce the following notations,

$$\begin{array}{llll} \boldsymbol{G}_{00} = \frac{1}{3}\boldsymbol{I}+\frac{1}{2}\boldsymbol{G}_1+\frac{1}{6}\boldsymbol{G}_1^3 & & & \\ \boldsymbol{G}_{10} = \frac{1}{2}(\boldsymbol{I}+\boldsymbol{G}_1^2)\boldsymbol{G}_2 & \boldsymbol{G}_{01} = \frac{1}{2}(\boldsymbol{I}+\boldsymbol{G}_1^2)\hat{\boldsymbol{G}}_2 & & \\ \boldsymbol{G}_{20} = \frac{1}{2}\boldsymbol{G}_1\boldsymbol{G}_2^2 & \boldsymbol{G}_{11} = \frac{1}{2}\boldsymbol{G}_1\boldsymbol{G}_2\hat{\boldsymbol{G}}_2 & \boldsymbol{G}_{02} = \frac{1}{2}\boldsymbol{G}_1\hat{\boldsymbol{G}}_2^2 & \\ \boldsymbol{G}_{30} = \frac{1}{6}\boldsymbol{G}_2^3 & \boldsymbol{G}_{21} = \frac{1}{6}\boldsymbol{G}_2^2\hat{\boldsymbol{G}}_2 & \boldsymbol{G}_{12} = \frac{1}{6}\boldsymbol{G}_2\hat{\boldsymbol{G}}_2^2 & \boldsymbol{G}_{03} = \frac{1}{6}\hat{\boldsymbol{G}}_2^3 \end{array}$$

and

$$\tilde{\boldsymbol{G}}_1 = \boldsymbol{G}_{00}+\boldsymbol{G}_{01}+\boldsymbol{G}_{02}+\boldsymbol{G}_{03},\ \tilde{\boldsymbol{G}}_2 = \boldsymbol{G}_{10}+\boldsymbol{G}_{11}+\boldsymbol{G}_{12},\ \tilde{\boldsymbol{G}}_3 = \boldsymbol{G}_{20}+\boldsymbol{G}_{21}$$

then we have

$$\begin{aligned} \boldsymbol{U}^{j,n+1} =& \boldsymbol{G}_{00}\boldsymbol{U}^{j,n}+\boldsymbol{G}_{10}\boldsymbol{U}^{j-1,n}+\boldsymbol{G}_{20}\boldsymbol{U}^{j-2,n}+\boldsymbol{G}_{30}\boldsymbol{U}^{j-3,n},\ j\geq 4, \\ \boldsymbol{U}^{3,n+1} =& \boldsymbol{G}_{00}\boldsymbol{U}^{3,n}+\boldsymbol{G}_{10}\boldsymbol{U}^{2,n}+\tilde{\boldsymbol{G}}_3\boldsymbol{U}^{1,n}, \\ \boldsymbol{U}^{2,n+1} =& \boldsymbol{G}_{00}\boldsymbol{U}^{2,n}+\tilde{\boldsymbol{G}}_2\boldsymbol{U}^{1,n}, \\ \boldsymbol{U}^{1,n+1} =& \tilde{\boldsymbol{G}}_1\boldsymbol{U}^{1,n}. \end{aligned} \tag{56}$$

Since $\boldsymbol{U}^{j,0} = \boldsymbol{U}^0$, $\forall j \geq 1$, the recurrence relations (56) lead to

$$\begin{aligned} \boldsymbol{U}^{1,n} =& \tilde{\boldsymbol{G}}_1^n\boldsymbol{U}^0, \\ \boldsymbol{U}^{2,n} =& \left(\boldsymbol{G}_{00}^n+\tilde{\boldsymbol{G}}_2\sum_{l=0}^{n-1}\boldsymbol{G}_{00}^l\tilde{\boldsymbol{G}}_1^{n-1-l}\right)\boldsymbol{U}^0, \\ \boldsymbol{U}^{3,n} =& \left(\boldsymbol{G}_{00}^n+\boldsymbol{G}_{00}^{n-1}(n\boldsymbol{G}_{10}+\tilde{\boldsymbol{G}}_3)+\sum_{l=0}^{n-2}\left((l+1)\boldsymbol{G}_{10}\tilde{\boldsymbol{G}}_2+\tilde{\boldsymbol{G}}_1\tilde{\boldsymbol{G}}_3\right)\boldsymbol{G}_{00}^l\tilde{\boldsymbol{G}}_1^{n-2-l}\right)\boldsymbol{U}^0. \end{aligned} \tag{57}$$

Through mathematical induction, the general formula for $\boldsymbol{U}^{j,n}$ for any $j \geq 1$ can be obtained.

5.3 GSA Analysis

At this point, for both the LWDG3 scheme and the RKDG3 scheme, we have obtained $\boldsymbol{U}^{j,n} = \boldsymbol{G}^{j,n}\boldsymbol{U}^0$, with $\boldsymbol{G}^{j,n}$ given by (52) and (57) respectively .

Let the eigenvalues and corresponding eigenvectors of $\boldsymbol{G}^{j,n}$ $(j \geq 2)$ be $\{(\tilde{\lambda}_l, \tilde{\boldsymbol{r}}_l),\ l = 0, 1, \cdots, q\}$. Again define $\tilde{\boldsymbol{\Lambda}} = diag(\tilde{\lambda}_0, \cdots, \tilde{\lambda}_q)$ and $\tilde{\boldsymbol{R}} = [\tilde{\boldsymbol{r}}_0, \cdots, \tilde{\boldsymbol{r}}_q]$, then $\boldsymbol{G}^{j,n} = \tilde{\boldsymbol{R}}\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{R}}^{-1}$. With reference to the derivation process of the eigen-solution in Sec. 3.1, setting $\tilde{\boldsymbol{\mu}} = \tilde{\boldsymbol{R}}^{-1}\boldsymbol{U}^0 = [\tilde{\mu}_0, \cdots, \tilde{\mu}_q]^T$ and $\boldsymbol{U}_l^{j,n} = \tilde{\lambda}_l\tilde{\boldsymbol{r}}_l$, we can get

$$\boldsymbol{U}^{j,n} = \boldsymbol{G}^{j,n}\boldsymbol{U}^0 = \sum_{l=0}^{q} \tilde{\mu}_l \boldsymbol{U}_l^{j,n}. \tag{58}$$

It should be noted that, since we have not solved the eigenvalue problem in the strict sense, the eigen-solution defined here is only analogical.

As before, let the 0-th eigen-solution be the physical mode, which should satisfy $|\tilde{\lambda}_0| \sim 1$ as $kh \to 0$. By

$$\begin{cases} \tilde{\lambda}_0 = |\tilde{\lambda}_0| e^{-i\tilde{\phi}} \\ \tilde{\lambda}_0 = e^{-i\tilde{\omega}_N t^n} = e^{n\cdot\tilde{\omega}_{Ni}\Delta t} e^{-in\cdot\tilde{\omega}_{Nr}\Delta t} \end{cases}$$

we can obtain

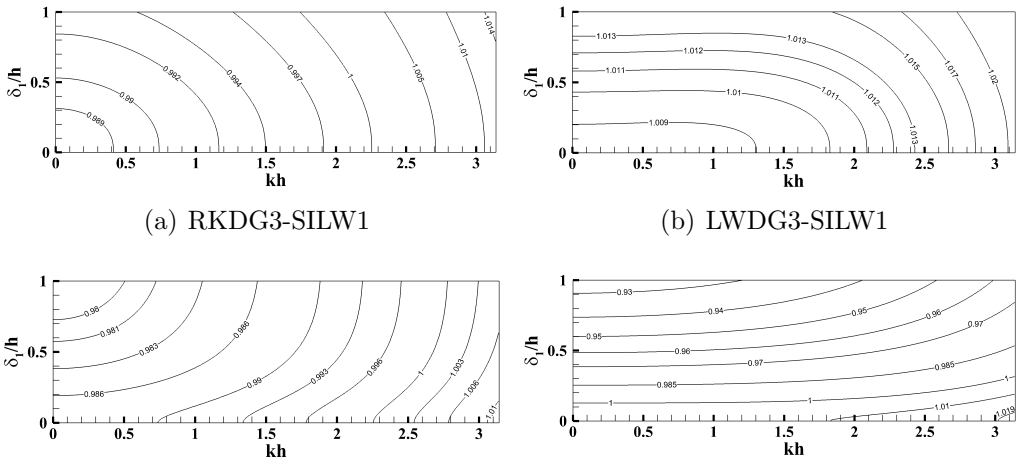
$$\tilde{\phi} = n \cdot \tilde{\omega}_{Nr}\Delta t. \tag{59}$$

Define $\tilde{\omega}_{Nr} = k\tilde{c}_N$. Thus, the phase velocity is $\tilde{c}_N = \frac{\tilde{\omega}_{Nr}}{k}$, and group velocity is $\tilde{v}_{gN} = \frac{\mathrm{d}\tilde{\omega}_{Nr}}{\mathrm{d}k}$. So we have

$$\frac{\tilde{c}_N}{c} = \frac{\tilde{\phi}}{nN_c \cdot kh}, \quad \frac{\tilde{v}_{gN}}{c} = \frac{1}{nN_c}\frac{\mathrm{d}\tilde{\phi}}{\mathrm{d}kh}$$

Next, we will compare the phase velocity and group velocity within the internal cells applying different boundary treatments, which involves different cells and different time instances.

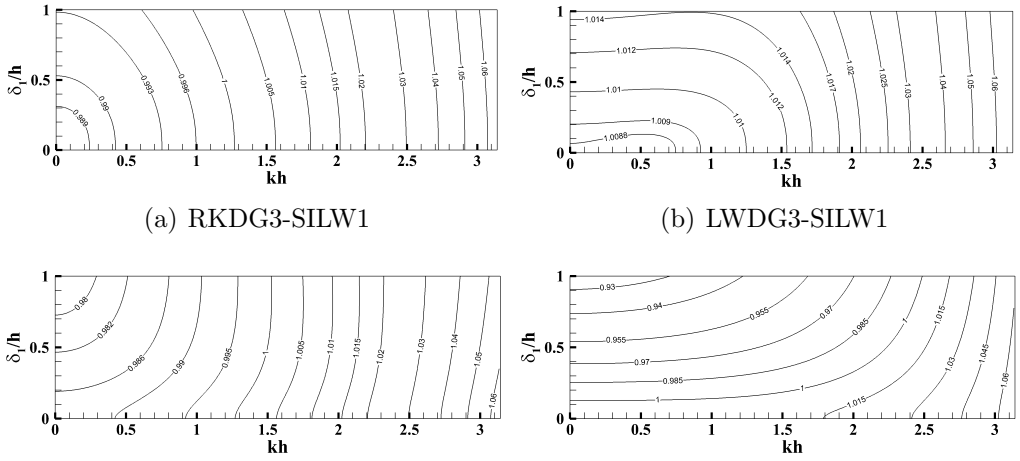
We first present here the results for the second element ($j = 2$ and $n = 5$), wherein for the RKDG3 scheme, N_c is taken as 0.075, and for LWDG3 scheme, $N_c = 0.085$. From Fig .12 and Fig .13, it can be seen that the RKDG3 scheme is less affected by δ_1/h, especially in the region where $kh > 1.55$. On the other hand, the LWDG3 scheme is consistently more affected by δ_1/h, and in the case where the SILW1 boundary treatment method is used, the effect significantly decreases in the region where $kh > 1.85$. Using SILW2 method is always more influenced by δ_1/h than using SILW1 method.



(a) RKDG3-SILW1 (b) LWDG3-SILW1

(c) RKDG3-SILW2 (d) LWDG3-SILW2

Fig. 12: The contour map of c_N/c in the second element with respect to δ_1/h and kh.



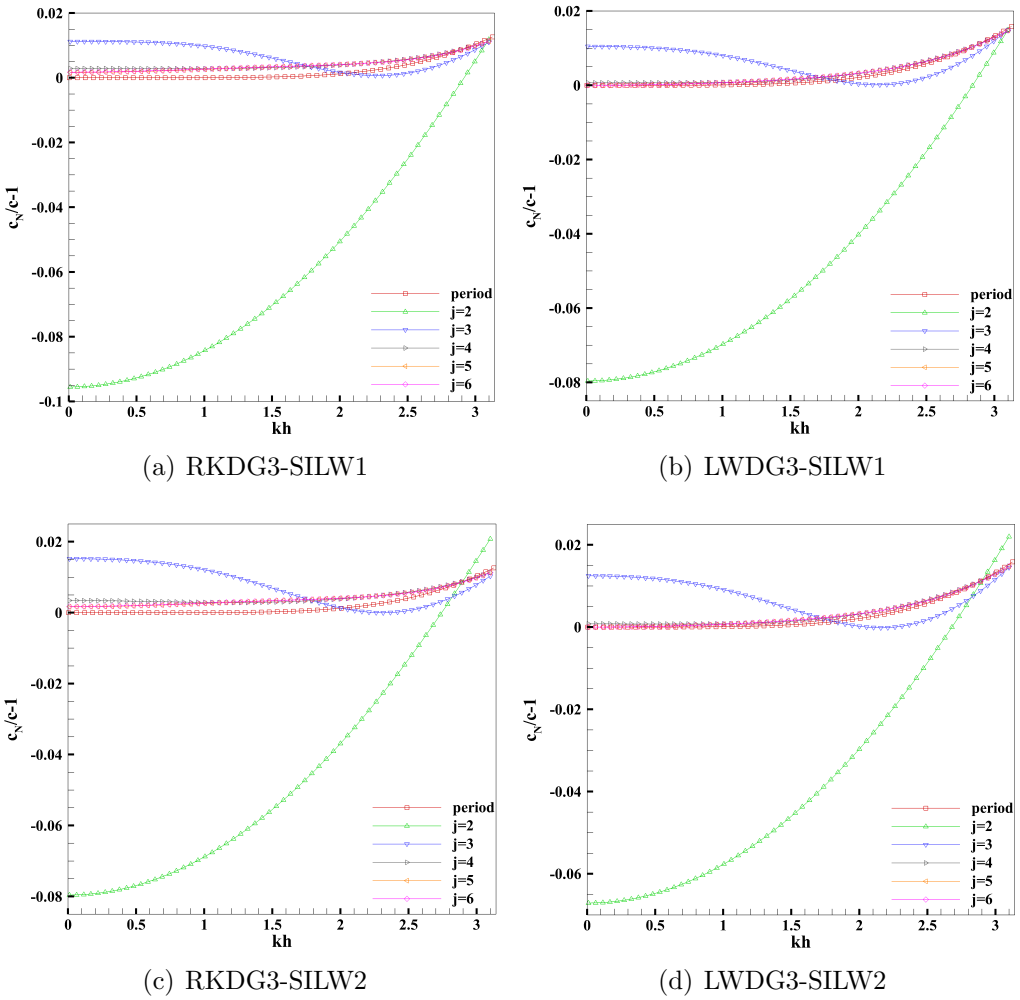
(a) RKDG3-SILW1 (b) LWDG3-SILW1

(c) RKDG3-SILW2 (d) LWDG3-SILW2

Fig. 13: The contour map of v_{gN}/c in the second element with respect to δ_1/h and kh.

To more clearly compare the impact of boundary treatment methods on dispersion error, we have plotted the trends of phase velocity error in Fig. 14 and group velocity error in Fig. 15 with respect to kh for several cells near the boundary, and compared these with the results under periodic boundary conditions. Here we set $\delta_1/h = 0.5$, for RKDG3 we take $N_c = 0.02$, and for

LWDG3 we take $N_c = 0.016$. In Fig. 14, we can observe that the LWDG3 scheme approaches the phase velocity under periodic boundary conditions faster than the RKDG3 scheme. The same applies to Fig. 15. We speculate that this is determined by the width of the stencil. The LWDG3 scheme has a narrower computational stencil.



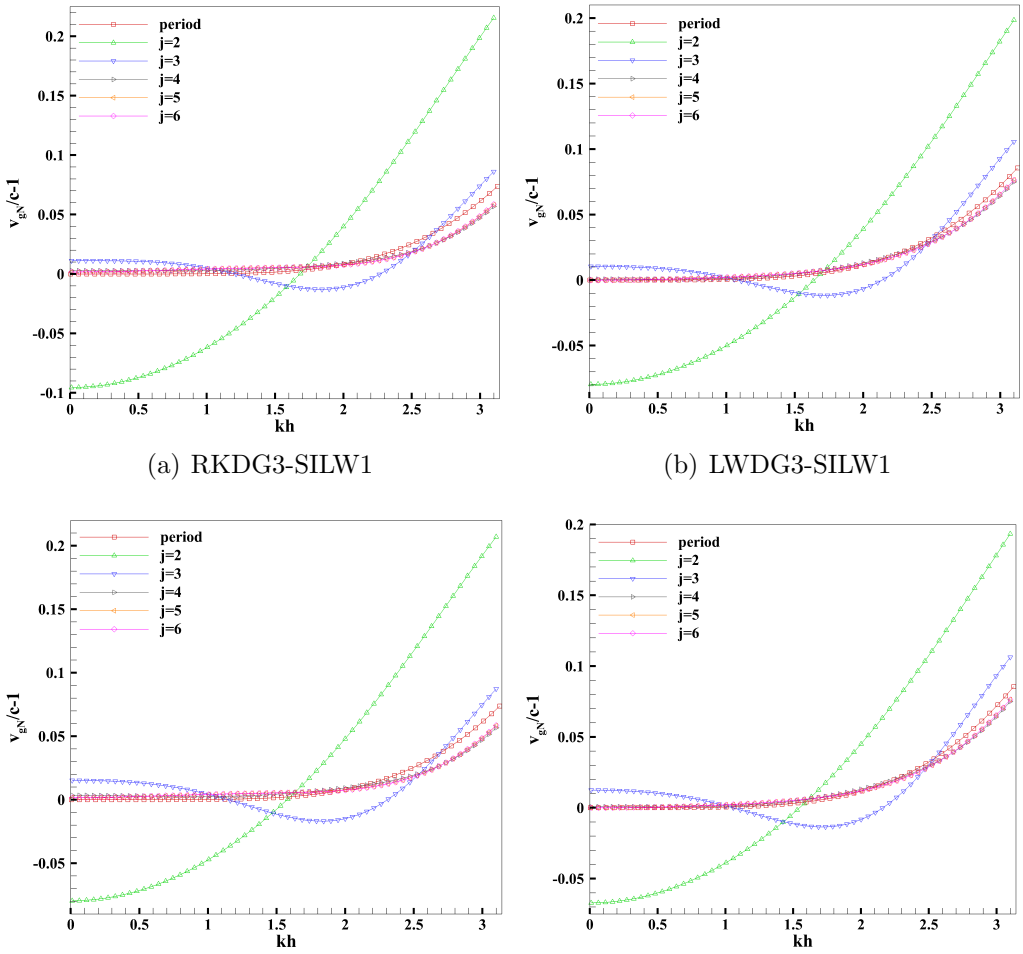
(a) RKDG3-SILW1

(b) LWDG3-SILW1

(c) RKDG3-SILW2

(d) LWDG3-SILW2

Fig. 14: Compare the phase velocity errors of the elements near the boundary with those under periodic boundary conditions.

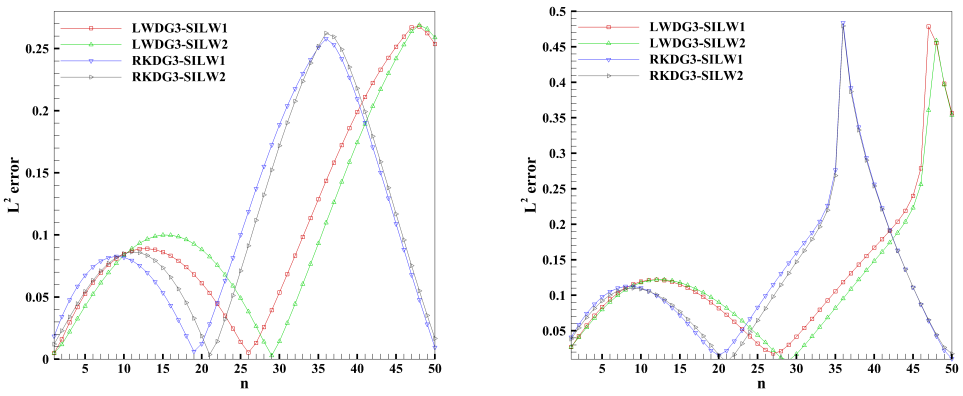


(a) RKDG3-SILW1 (b) LWDG3-SILW1

(c) RKDG3-SILW2 (d) LWDG3-SILW2

Fig. 15: Compare the group velocity errors of the elements near the boundary with those under periodic boundary conditions.

Next, we fix $j = 2$, that is, select the second element, to investigate the variation of dispersion error over time layers. Here we set $\delta_1/h = 0.5$, for RKDG3 we take $N_c = 0.02$, and for LWDG3 we take $N_c = 0.015$. As can be seen from Fig. 16, initially, the dispersion error of the RKDG3 scheme is greater than that of the LWDG3 scheme. However, as time evolves, the dispersion error of the RKDG3 scheme becomes significantly smaller than that of the LWDG3 scheme, and this phenomenon will recur.



(a) $c_N/c-1$ (b) $v_{gN}/c-1$

Fig. 16: The trend of dispersion error (here represented by L^2 error) over time.

6 Numerical Experiments

In the following numerical experiments, we will use different initial conditions and boundary conditions for the equation $u_t + u_x = 0$ to verify the results of our previous analysis.

6.1 Periodic boundary conditions

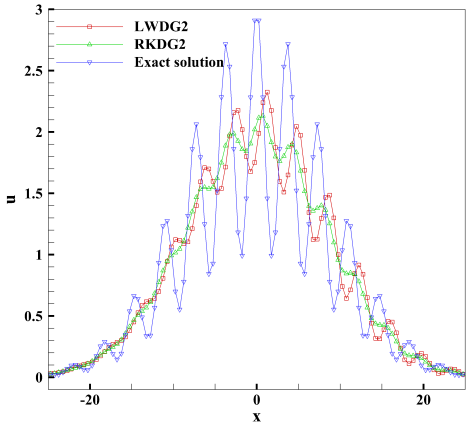
Example 6.1 In this example, we use the test case from [36] to compare the numerical dissipation and numerical dispersion between the LWDG scheme and the RKDG scheme. we have

$$\begin{cases} u_t + u_x = 0, & -75 < x < 75,\ t > 0, \\ u(x,0) = (2+\cos 1.7x)e^{-\ln 2\,(x/10)^2}, \end{cases} \tag{60}$$

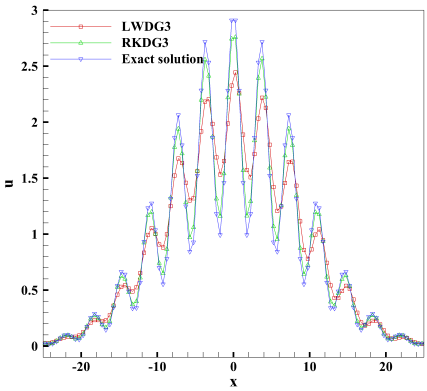
with periodic boundary condition. We divided the computational domain into 300 equal cells, and set $N_c = 0.15$. To better demonstrate the differences, we extended the computation time: the second-order scheme's results show the solution after one period, while the third-order scheme was computed for five periods. The results are shown in the Fig. 17. It is observed that the numerical results are in complete agreement with the analytical results from Fig. 2 – 7,

- Higher-order schemes have smaller dispersion and dissipation errors.

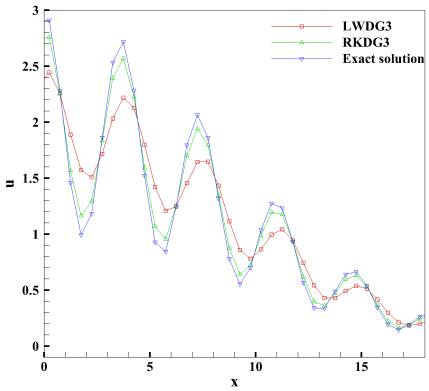
- The RKDG2 scheme has larger dissipative errors than the LWDG2 scheme, but its dispersive errors are smaller.
- The RKDG3 scheme has smaller dispersive and dissipative errors compared to the LWDG3 scheme.



(a) $q = 1$, $t = 150$



(b) $q = 2$, $t = 750$



(c) Zoomed-in section of (b)

Fig. 17: Example 6.1: A comparison between the RKDG scheme and the LWDG scheme under periodic boundary conditions.

Example 6.2 In Fig. 5(a), we have obtained that when $kh > 2.71$, the dissipation error of LWDG3 scheme will be smaller than that of RKDG3

scheme. The following example serves to verify this point. We consider a high-frequency wave test,

$$\begin{cases} u_t + u_x = 0, & -\pi < x < \pi,\ t > 0, \\ u(x,0) = \cos 20x, \end{cases} \tag{61}$$

with periodic boundary condition. In the numerical tests, we take $N_c = 0.16$.

In Fig. 18, we show the results of two grids with different sizes. Fig. 18(a) corresponds to $kh \approx 1.26$. We can see that the dissipation error of LWDG3 scheme is larger than that of RKDG3 scheme. In fact, from Fig. 2, we have $|\lambda_0| = 0.9999$ for RKDG3 scheme and $|\lambda_0| = 0.9995$ for LWDG3 scheme, so this result is natural.

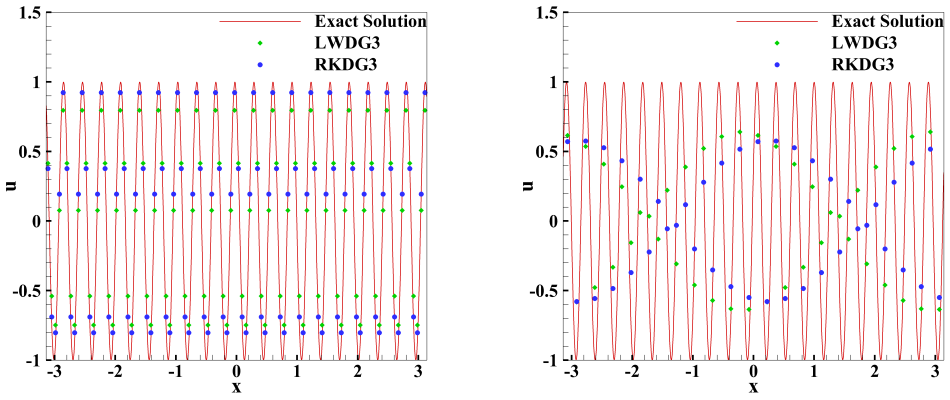
We use a coarser grid in Fig. 18(b), which makes $kh \approx 2.99$. Because the grid is particularly sparse, it is not always possible to get the extreme point, we can see that the amplitude of the numerical solution is changing. Moreover, when $kh \approx 2.99$, from Fig. 7 (a), we observe that the LWDG3 scheme yields $v_{gN}/c = 1.1478$, while the RKDG3 scheme gives $v_{gN}/c = 1.068$. This indicates that the LWDG3 scheme exhibits larger dispersion errors, which is also clearly visible in the figure. Moreover, the amplitude magnitude indicates that the LWDG3 scheme exhibits smaller dissipation errors at this stage. This observation is consistent with the results shown in Fig. 2, where $|\lambda_0| = 0.9898$ for the LWDG3 scheme and $|\lambda_0| = 0.9878$ for the RKDG3 scheme.

Example 6.3 We have shown that the group velocity error of the RKDG2 scheme is a little smaller than that of the LWDG3 scheme when $kh > 2.94$ in Fig. 7(b), although its dissipation error is considerably worse. This phenomenon can be observed in the following meticulously designed example,

$$\begin{cases} u_t + u_x = 0, & -60 < x < 60,\ t > 0, \\ u(x,0) = e^{-0.01x^2} \cos 20x, \end{cases} \tag{62}$$

with periodic boundary condition. In our computations, we set $N_c = 0.1$ and $N = 800$. The end time is set at $t = 5$.

As can be seen from Fig. 19, the group velocity error of RKDG2 scheme is comparable to that of the LWDG3 scheme starting with the wave packet on the left. Furthermore, the relative size of the group velocity error at $kh \approx 3.00$ can be judged according to the wave packets in the blue boxes: RKDG2 < LWDG3 < LWDG2, which is consistent with the conclusion in Fig. 7(b).



(a) $N = 100$, $t = 5$. (b) $N = 42$, $t = 1$.

Fig. 18: Example 6.2: Compare the dissipative errors between LWDG3 scheme and RKDG3 scheme for different kh. The grid used for the exact solution satisfies $N = 1000$.

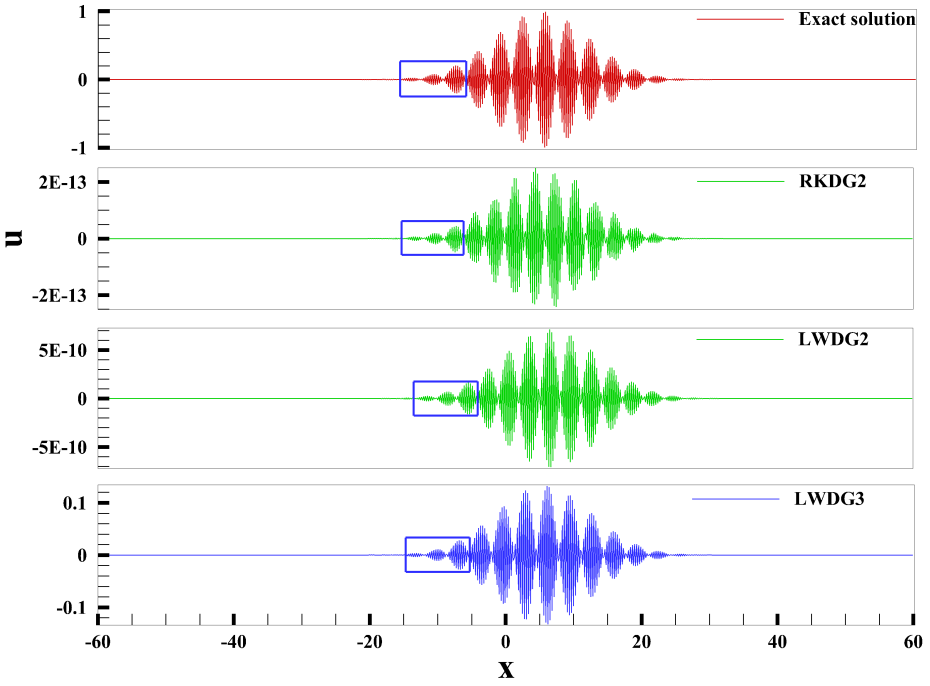


Fig. 19: Example 6.3: Compare the group velocities of RKDG2, LWDG2 and LWDG3 at $kh \approx 3.00$. The relative group velocities of RKDG2, LWDG3 and LWDG2 are 1.1031, 1.1220, 1.3293 in Fig. 7(b), respectively.

6.2 Non-periodic boundary conditions

Example 6.4 Similar to [47], it is necessary to conduct an accuracy test on the boundary treatment method corresponding to the LWDG scheme developed in Sec. 4. Specifically, we employ the following example,

$$\begin{cases} u_t + u_x = 0, \quad 0 < x < 2\pi,\ t > 0, \\ u(x,0) = -\sin x, \\ u(0,t) = \sin t. \end{cases} \tag{63}$$

This example has the exact solution $u(x,t) = \sin(t-x)$. To test the effectiveness of the algorithm, we intentionally misalign the computational grid with the physical boundary in our calculations. For the LWDG2 scheme, we take $N_c = 0.333$, and for the LWDG3 scheme, we take $N_c = 0.170$. Tab. 2 - 3 show that the schemes can always achieve the $(q+1)$-th order accuracy. Moreover, the errors of the scheme satisfying local conservation do not vary with δ_1/h significantly, as show in Fig. 10.

Tab. 2: Example 6.4: L^2 errors and orders of accuracy for advection equation with LWDG2 at $t = 3$.

		ILW		SILW1	
δ_1/h	N	error	order	error	order
0.01	20	9.06E-03	—	9.06E-03	—
	40	2.45E-03	1.88	2.45E-03	1.88
	80	6.35E-04	1.95	6.35E-04	1.95
	160	1.62E-04	1.97	1.62E-04	1.97
	320	4.09E-05	1.99	4.09E-05	1.99
	640	1.03E-05	1.99	1.03E-05	1.99
0.99	20	8.67E-03	—	1.35E-02	—
	40	2.34E-03	1.89	2.97E-03	2.18
	80	6.16E-04	1.92	6.93E-04	2.10
	160	1.59E-04	1.95	1.69E-04	2.04
	320	4.05E-05	1.98	4.17E-05	2.02
	640	1.02E-05	1.99	1.04E-05	2.01

Example 6.5 In this example, we will employ the LWDG3 scheme to solve the Burgers' equation,

$$\begin{cases} u_t + (\frac{u^2}{2})_x = 0,\ -\pi < x < \pi,\ t > 0, \\ u(x,0) = 1 + 2\sin x, \\ u(-\pi,t) = g(t) \end{cases} \tag{64}$$

Tab. 3: Example 6.4: L^2 errors and orders of accuracy for advection equation with LWDG3 at $t = 3$.

		ILW		SILW1		SILW2	
δ_1/h	N	error	order	error	order	error	order
0.01	20	1.65E-04	—	1.65E-04	—	1.65E-04	—
	40	2.22E-05	2.90	2.22E-05	2.90	2.22E-05	2.90
	80	2.89E-06	2.94	2.89E-06	2.94	2.89E-06	2.94
	160	3.68E-07	2.97	3.68E-07	2.97	3.68E-07	2.97
	320	4.65E-08	2.98	4.65E-08	2.98	4.65E-08	2.98
	640	5.84E-09	2.99	5.84E-09	2.99	5.84E-09	2.99
0.99	20	4.88E-04	—	4.24E-04	—	6.48E-04	—
	40	4.12E-05	3.57	3.16E-05	3.75	5.23E-05	3.63
	80	3.91E-06	3.40	2.91E-06	3.44	4.55E-06	3.52
	160	4.22E-07	3.21	3.45E-07	3.07	4.56E-07	3.32
	320	4.95E-08	3.09	4.43E-08	2.96	5.12E-08	3.15
	640	6.01E-09	3.04	5.68E-09	2.96	6.11E-09	3.07

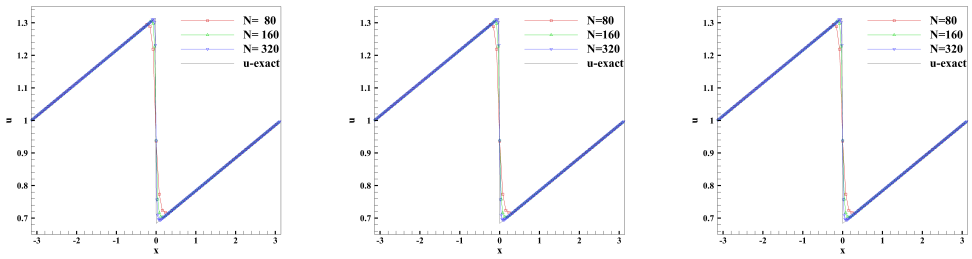
Here, $g(t)$ is given as the exact solution with periodic boundary conditions at $x = -\pi$. For all time, the left boundary $x = -\pi$ is an inflow boundary and the right boundary $x = \pi$ is an outflow boundary. In the computation, we take the CFL number $N_c = 0.12$.

Tab. 4: Example 6.5: L^2 errors and orders of accuracy for the Burgers' equation with LWDG3 at $t = 0.3$.

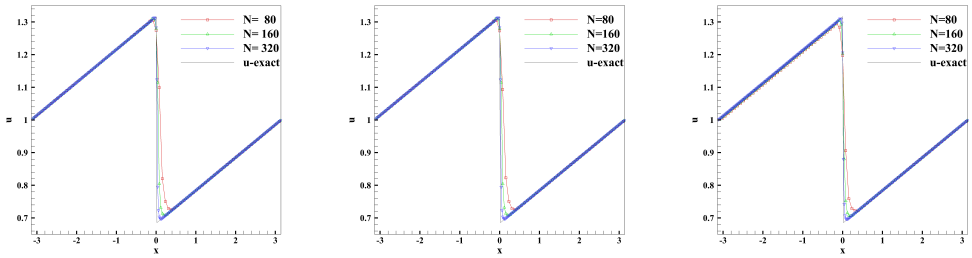
		ILW		SILW1		SILW2	
δ_1/h	N	error	order	error	order	error	order
0.01	40	4.15E-04	—	4.15E-04	—	4.15E-04	—
	80	7.42E-05	2.48	7.42E-05	2.48	7.42E-05	2.48
	160	1.22E-05	2.60	1.22E-05	2.60	1.22E-05	2.60
	320	1.87E-06	2.71	1.87E-06	2.71	1.87E-06	2.71
	640	2.70E-07	2.79	2.70E-07	2.79	2.70E-07	2.79
	1280	3.73E-08	2.86	3.73E-08	2.86	3.73E-08	2.86
0.99	40	1.97E-03	—	1.59E-03	—	1.37E-03	4.21
	80	1.39E-04	3.83	1.35E-04	3.56	1.25E-04	3.45
	160	1.42E-05	3.29	1.43E-05	3.24	1.50E-05	3.05
	320	1.92E-06	2.88	1.93E-06	2.89	1.97E-06	2.93
	640	2.71E-07	2.83	2.71E-07	2.83	2.73E-07	2.85
	1280	3.73E-08	2.86	3.73E-08	2.86	3.74E-08	2.87

It's known that the exact solution will develop a discontinuity for $t > 0.5$, while the solution remains smooth before that time. We utilized the solutions at $t = 0.3$ to test the L^2 errors and orders of the scheme. As can be seen in Tab. 4, the scheme achieves the expected order accuracy, and the error is not affected by δ_1/h.

A shock enters the left boundary at $t = 2\pi$ and moves to $x = 0$ at $t = 3\pi$. Internally, we employ the TVB limiter [10] to prevent numerical oscillations, while on the boundary element I_1, we use the nonlinear limiter from [47] to ensure the stability of the scheme. The Fig. 20 demonstrates the scheme's capability to capture shock.



(a) ILW (b) SILW1 (c) SILW2



(d) ILW (e) SILW1 (f) SILW2

Fig. 20: Example 6.5: Numerical solutions for Burgers' equation with LWDG3 at $t = 3\pi$. Top: $\delta_1/h = 0.01$; bottom: $\delta_1/h = 0.99$.

Example 6.6 Next, we consider the linearized one-dimensional Euler equations on a uniform mean flow:

$$\begin{cases} \rho_t + M\rho_x + u_x = 0, \\ u_t + Mu_x + p_x = 0, \\ p_t + Mp_x + u_x = 0. \end{cases} \tag{65}$$

where M is constant mean flow Mach number. The exact solution of this system of equations satisfies:

$$\begin{aligned}
\rho(x,t) &= \rho(x-Mt,\,0) + \Big(p(x,t) - p(x-Mt,\,0)\Big),\\
u(x,t) &= u(x-(M-1)t,\,0) + \Big(p(x,t) - p(x-(M-1)t,\,0)\Big),\\
p(x,t) &= \frac{1}{2}\Big(u(x-(M+1)t,\,0) - u(x-(M-1)t,\,0)\Big)\\
&\quad + \frac{1}{2}\Big(p(x-(M+1)t,\,0) + p(x-(M-1)t,\,0)\Big).
\end{aligned}$$

The eigenvalues of coefficient matrix of system (65) are $M-1$, M, $M+1$. Here we take $M = 1.5$, in which case all characteristic components are inflow at the left boundary and outflow at the right boundary. The proposed (S)ILW boundary treatment for scalar problems will be applied on each component directly. We take the following initial values for the accuracy test,

$$u(x,0) = 0, \quad \rho(x,0) = p(x,0) = e^{-x^2}, \quad x \in [0,10].$$

In the computation, the three boundary conditions at $x = 0$ are provided by the exact solution, and $N_c = 0.170$ is used. Tab. 5 presents the L^2 error of the density ρ at $t = 3$, and the results are consistent with expectations.

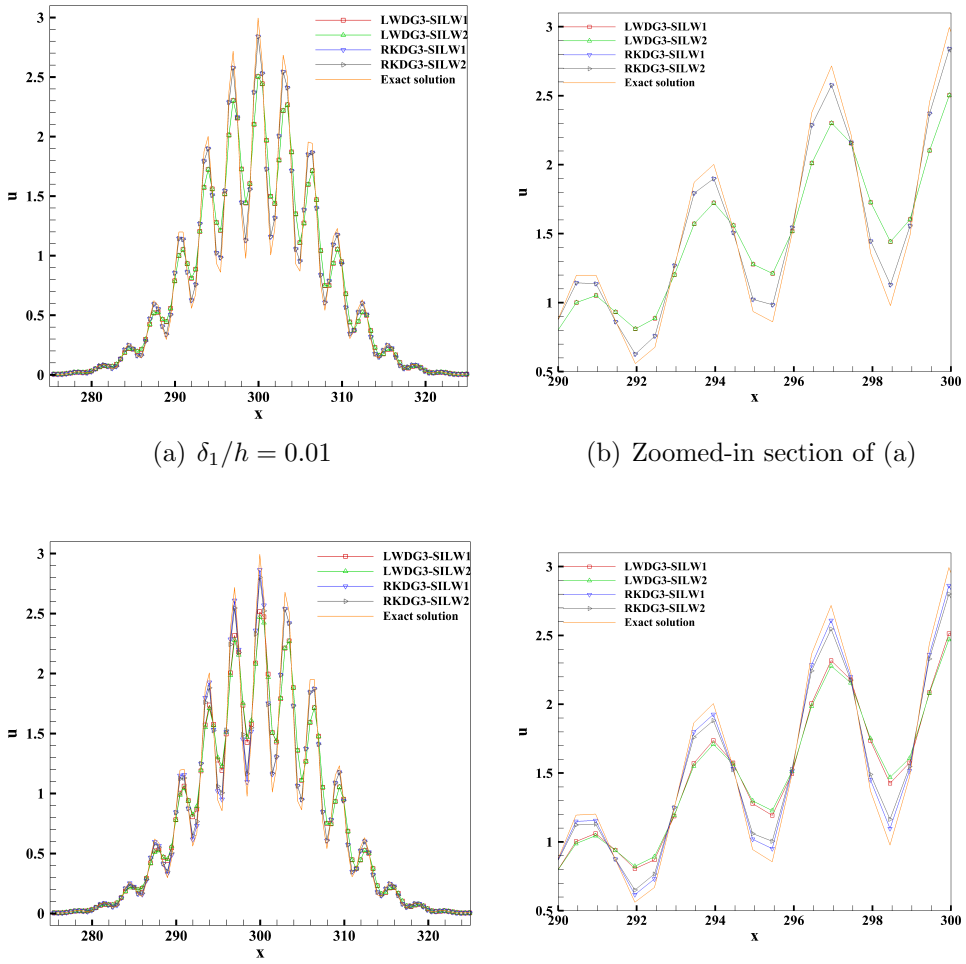
Tab. 5: Example 6.6: L^2 errors and orders of accuracy for ρ in the linearized Euler equations with LWDG3 at $t = 3$.

		ILW		SILW1		SILW2	
δ_1/h	N	error	order	error	order	error	order
0.01	40	7.30E-04	—	7.30E-04	—	7.30E-04	—
	80	9.35E-05	2.97	9.35E-05	2.97	9.35E-05	2.97
	160	1.19E-05	2.98	1.19E-05	2.98	1.19E-05	2.98
	320	1.50E-06	2.99	1.50E-06	2.99	1.50E-06	2.99
	640	1.88E-07	2.99	1.88E-07	2.99	1.88E-07	2.99
0.99	40	1.31E-03	—	7.75E-04	—	1.74E-03	—
	80	1.30E-04	3.33	7.58E-05	3.35	1.55E-04	3.49
	160	1.40E-05	3.22	1.02E-05	2.89	1.52E-05	3.34
	320	1.62E-06	3.11	1.38E-06	2.89	1.68E-06	3.18
	640	1.95E-07	3.05	1.80E-07	2.94	1.99E-07	3.08

Example 6.7 Finally, we consider the following advection equation:

$$\begin{cases} u_t + u_x = 0, \quad 0 < x < 500,\ t > 0,\\ u(x,0) = (2+\cos 2x)e^{-0.01x^2},\\ u(0,t) = (2+\cos 2t)e^{-0.01t^2}. \end{cases} \tag{66}$$

The computational interval is divided into 1000 equal parts, with the computation terminated at time $t = 100$, and N_c is taken as 0.15. We can observe from Fig. 21 that, using the same internal numerical scheme, different boundary treatments have little impact on the dispersion properties of the scheme when $\delta_1/h = 0.01$, but for $\delta_1/h = 0.99$, the effect is obvious. This is consistent with the conclusions we obtained from Fig. 12 and Fig. 13.



(a) $\delta_1/h = 0.01$ (b) Zoomed-in section of (a)

(c) $\delta_1/h = 0.99$ (d) Zoomed-in section of (c)

Fig. 21: Example 6.7: Compare the performance of different boundary treatment methods with varying δ_1/h.

7 Conclusion

In this paper, we analyze the dispersion and dissipation properties of two fully discrete DG methods by employing the GSA method. Firstly, we derived the midpoint error propagation equation for the general form of DG solution (initial value is not a Fourier single wave), and found that the influencing factors of DG error propagation include $|\lambda_0|$, c_N/c and v_{gN}. For the second-order and third-order RKDG and LWDG, we drew the graphs and discussed those properties respectively. Next, when the calculation area is inconsistent with the grid, we designed a simplified ILW method for the LWDG method and improved the accuracy of the scheme through conservation correction. Finally, we analyzed the dispersion error at the boundary via GSA method as well. A number of numerical examples verify our theoretical results.

In the future, we can consider optimizing the algorithm based on the research methods in this paper, such as using the DG method based on upwind-biased fluxes [21], using the new SILW method with adjustable parameters at the boundary [20], or using a higher resolution time discretization scheme.

A Construction of Numerical Flux $\hat{F}^c_{1/2}$

For the SILW1 method, substituting (35) into (39) we have

$$\begin{aligned}\hat{F}^c_{\frac{1}{2}} = F|_{x=a} &- \tilde{C}(q)h\frac{\bar{u}_1(t+\Delta t)-\bar{u}_1(t)}{\Delta t}\\ &+\sum_{m=0}^{q-1}\frac{\tilde{C}(q)((\delta_1+h)^{m+1}-\delta_1^{m+1})-\delta_1^{m+1}}{(m+1)!}\frac{\partial_x^{(m)}u(a,t+\Delta t)-\partial_x^{(m)}u(a,t)}{\Delta t}.\end{aligned} \tag{67}$$

Here, $\frac{\bar{u}_1(t+\Delta t)-\bar{u}_1(t)}{\Delta t}$ is unknown, but we can immediately obtain it using Eq. (37), we get

$$\frac{\bar{u}_1(t+\Delta t)-\bar{u}_1(t)}{\Delta t} = \frac{1}{h}(\hat{F}^c_{\frac{1}{2}}-\hat{F}_{\frac{3}{2}}). \tag{68}$$

Solve the equations by combining (67) and (68), we can obtain (41).

For the SILW2 method, substituting (36) into (39) we have

$$\begin{aligned}\hat{F}^c_{\frac{1}{2}} =&F|_{x=a} - C_1(q)h\frac{\bar{u}_1(t+\Delta t)-\bar{u}_1(t)}{\Delta t} - C_2(q)h\frac{(\bar{u}_x)_1(t+\Delta t)-(\bar{u}_x)_1(t)}{\Delta t}\\ &+\sum_{m=0}^{q-2}\frac{C_1(q)((\delta_1+h)^{m+1}-\delta_1^{m+1})+C_2(q)((\delta_1+h)^m-\delta_1^m)(m+1)-\delta_1^{m+1}}{(m+1)!}\cdot\\ &\frac{\partial_x^{(m)}u(a,t+\Delta t)-\partial_x^{(m)}u(a,t)}{\Delta t}.\end{aligned} \tag{69}$$

The $\frac{\bar{u}_1(t+\Delta t)-\bar{u}_1(t)}{\Delta t}$ can be given by (68), but the expression for the $\frac{(\bar{u}_x)_1(t+\Delta t)-(\bar{u}_x)_1(t)}{\Delta t}$ depends on the polynomial degree q ($q \geq 2$). For example, for $q = 4$ and $u_h(x,t)|_{I_1} = \sum_{l=0}^{4} u_l^1(t)\varphi_l^1(x)$, we have

$$(\bar{u}_x)_1(t) = \frac{2}{h}(u_1^1(t) + u_3^1(t)).$$

By setting $j = 1$, $v = \varphi_1^1(x)$ and $v = \varphi_3^1(x)$ in (6), we can obtain

$$\begin{aligned}
\frac{u_1^1(t+\Delta t) - u_1^1(t)}{\Delta t} =& \frac{3}{h}\left(\int_{-1}^{1} F(u_h)\varphi_1'(\xi)\,\mathrm{d}\xi - \hat{F}_{\frac{3}{2}} - \hat{F}_{\frac{1}{2}}^c\right), \\
\frac{u_3^1(t+\Delta t) - u_3^1(t)}{\Delta t} =& \frac{7}{h}\left(\int_{-1}^{1} F(u_h)\varphi_3'(\xi)\,\mathrm{d}\xi - \hat{F}_{\frac{3}{2}} - \hat{F}_{\frac{1}{2}}^c\right).
\end{aligned}$$

So we can get

$$\frac{(\bar{u}_x)_1(t+\Delta t) - (\bar{u}_x)_1(t)}{\Delta t} = \begin{cases} \frac{6}{h^2}\left(\int_{-1}^{1} F(u_h)\varphi_1'(\xi)\,\mathrm{d}\xi - (\hat{F}_{\frac{1}{2}}^c + \hat{F}_{\frac{3}{2}})\right), & q = 2 \\ \frac{6}{h^2}\int_{-1}^{1} F(u_h)\varphi_1'(\xi)\,\mathrm{d}\xi + \frac{14}{h^2}\int_{-1}^{1} F(u_h)\varphi_3'(\xi)\,\mathrm{d}\xi & \\ -\frac{20}{h^2}(\hat{F}_{\frac{1}{2}}^c + \hat{F}_{\frac{3}{2}}), & q = 3, 4. \end{cases} \tag{70}$$

Substituting (68) and (70) into (69) to solve the equation, we can obtain (42) and (43).

Funding

Z. Cheng was partially supported by the National Natural Science Foundation of China (Nos. 12201169). Y. Jiang was also partially supported by the National Natural Science Foundation of China (Nos. 12271499).

Data Availability

Data will be made available from the corresponding author upon reasonable request.

Declarations

Conflict of interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

1. Ainsworth, M.: Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods. J. Comput. Phys. **198**, 106–130 (2004).
2. Alhawwary, M. and Wang, Z.: Comparative Fourier analysis of DG, FD and compact difference schemes. 2018 fluid dynamics conference, 4267 (2018).
3. Alhawwary, M. and Wang, Z.: Fourier analysis and evaluation of DG, FD and compact difference methods for conservation laws. J. Comput. Phys. **373**, 835–862 (2018).
4. Alhawwary, M. and Wang, Z.: A combined-mode Fourier analysis of DG methods for linear parabolic problems. SIAM J. Sci. Comput. **42**, A3825–A3858 (2020).
5. Bastian, P., Engwer, C., Fahlke, J., and Ippisch, O.: An unfitted discontinuous Galerkin method for pore-scale simulations of solute transport. Math. Comput. Simulat. **81**, 2051–2061 (2011).
6. Burman, E. and Hansbo, P.: Fictitious domain finite element methods using cut elements II: A stabilized Nitsche method. Appl. Numer. Math. **62**, 328–341 (2012).
7. Cheng, Z., Fang, J., Shu, C.-W., and Zhang, M.: Assessment of aeroacoustic resolution properties of DG schemes and comparison with DRP schemes. J. Comput. Phys. **399**, 108960 (2019).
8. Cockburn, B., Hou, S., and Shu, C.-W.: The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case. Math. Comput. **54**, 545–581 (1990).
9. Cockburn, B., Lin, S.-Y., and Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One-dimensional systems. J. Comput. Phys. **84**, 90–113 (1989).
10. Cockburn, B. and Shu, C.-W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework. Math. Comput. **52**, 411–435 (1989).
11. Cockburn, B. and Shu, C.-W.: The Runge-Kutta local projection discontinuous Galerkin finite element method for scalar conservation laws. ESAIM: M2AN. **25**, 337–361 (1991).
12. Cockburn, B. and Shu, C.-W.: The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. J. Comput. Phys. **141**, 199–224 (1998).

13. Fu, P., Frachon, T., Kreiss, G., and Zahedi, S.: High order discontinuous cut finite element methods for linear hyperbolic conservation laws with an interface. J. Sci. Comput. **90**, 84 (2022).

14. Fu, P. and Kreiss, G.: High order cut discontinuous Galerkin methods for hyperbolic conservation laws in one space dimension. SIAM J. Sci. Comput. **43**, A2404–A2424 (2021).

15. Giuliani, A.: A two-dimensional stabilized discontinuous Galerkin method on curvilinear embedded boundary grids. SIAM J. Sci. Comput. **44**, A389–A415 (2022).

16. Guo, W., Zhong, X., and Qiu, J.: Superconvergence of discontinuous Galerkin and local discontinuous Galerkin methods: Eigen-structure analysis based on Fourier approach. J. Comput. Phys. **235**, 458–485 (2013).

17. Gurkan, C., Sticko, S., and Massing, A.: Stabilized cut discontinuous Galerkin methods for advection-reaction problems. SIAM J. Sci. Comput. **42**, A2620–A2654 (2020).

18. Hansbo, P., Larson, M. G., and Zahedi, S.: A cut finite element method for a Stokes interface problem. Appl. Numer. Math. **85**, 90–114 (2014).

19. Hu, F. Q., Hussaini, M. Y., and Rasetarinera, P.: An analysis of the discontinuous Galerkin method for wave propagation problems. J. Comput. Phys. **151**, 921–946 (1999).

20. Liu, S., Li, T., Cheng, Z., Jiang, Y., Shu, C. W., and Zhang, M.: A new type of simplified inverse Lax-Wendroff boundary treatment I: Hyperbolic conservation laws. J. Comput. Phys. **514** (2024).

21. Meng, X., Shu, C.-W., and Wu, B.: Optimal error estimates for discontinuous Galerkin methods based on upwind-biased fluxes for linear hyperbolic equations. Math. Comput. **85**, 1225–1261 (2016).

22. Müller, B., Krämer-Eis, S., Kummer, F., and Oberlack, M.: A high-order discontinuous Galerkin method for compressible flows with immersed boundaries. Int. J. Numer. Meth. Eng. **110**, 3–30 (2017).

23. Neumann, J. von and Richtmyer, R.: On the numerical solutions of partial differential equations of parabolic type. John von Neumann Collected Works **5**, 18 (1947).

24. Qin, R. and Krivodonova, L.: A discontinuous Galerkin method for solutions of the Euler equations on Cartesian grids with embedded geometries. J. Comput. Sci-Neth. **4**, 24–35 (2013).

25. Qiu, J.: A numerical comparison of the Lax-Wendroff discontinuous Galerkin method based on different numerical fluxes. J. Sci. Comput. **30**, 345–367 (2007).

26. Qiu, J., Dumbser, M., and Shu, C.-W.: The discontinuous Galerkin method with Lax-Wendroff type time discretizations. Comput. Method. Appl. M. **194**, 4528–4543 (2005).

27. Reed, W. H. and Hill, T. R.: Triangular mesh methods for the neutron transport equation. Tech. rep. Los Alamos Scientific Lab., N. Mex.(USA) (1973).

28. Sagaut, P., Suman, V., Sundaram, P., Rajpoot, M., Bhumkar, Y., Sengupta, S., Sengupta, A., and Sengupta, T.: Global spectral analysis: Review of numerical methods. Comput. Fluids. **261**, 105915 (2023).

29. Schoeder, S., Sticko, S., Kreiss, G., and Kronbichler, M.: High-order cut discontinuous Galerkin methods with local time stepping for acoustics. Int. J. Numer. Meth. Eng. **121**, 2979–3003 (2020).

30. Sengupta, S., Sengupta, T. K., Puttam, J. K., and Vajjala, K. S.: Global spectral analysis for convection-diffusion-reaction equation in one and two-dimensions: Effects of numerical anti-diffusion and dispersion. J. Comput. Phys. **408**, 109310 (2020).

31. Sengupta, S., Sreejith, N., Mohanamuraly, P., Staffelbach, G., and Gicquel, L.: Global spectral analysis of the Lax-Wendroff central difference scheme applied to convection-diffusion equation. Comput. Fluids. **242**, 105508 (2022).

32. Sengupta, T.: High accuracy computing methods: fluid flows and wave phenomena. Cambridge University Press (2013).

33. Sengupta, T. K. and Bhole, A.: Error dynamics of diffusion equation: Effects of numerical diffusion and dispersive diffusion. J. Comput. Phys. **266**, 240–251 (2014).

34. Sengupta, T. K., Bhumkar, Y. G., Rajpoot, M. K., Suman, V., and Saurabh, S.: Spurious waves in discrete computation of wave phenomena and flow problems. Appl. Math. Comput. **218**, 9035–9065 (2012).

35. Sengupta, T. K., Dipankar, A., and Sagaut, P.: Error dynamics: Beyond von Neumann analysis. J. Comput. Phys. **226**, 1211–1218 (2007).

36. Sengupta, T. K., Sagaut, P., Sengupta, A., and Saurabh, K.: Global spectral analysis of three-time level integration schemes: Focusing phenomenon. Comput. Fluids. **157**, 182–195 (2017).

37. Sengupta, T. K., Sengupta, A., and Saurabh, K.: Global spectral analysis of multi-level time integration schemes: Numerical properties for error analysis. Appl. Math. Comput. **304**, 41–57 (2017).

38. Sherwin, S.: Dispersion analysis of the continuous and discontinuous Galerkin formulations. Discontinuous Galerkin Methods: Theory, Computation and Applications. Springer, 425–431 (2000).

39. Song, T., Main, A., Scovazzi, G., and Ricchiuto, M.: The shifted boundary method for hyperbolic systems: Embedded domain computations of linear waves and shallow water flows. J. Comput. Phys. **369**, 45–79 (2018).

40. Sticko, S. and Kreiss, G.: A stabilized Nitsche cut element method for the wave equation. Comput. Method. Appl. M. **309**, 364–387 (2016).

41. Sticko, S. and Kreiss, G.: Higher order cut finite elements for the wave equation. J. Sci. Comput. **80**, 1867–1887 (2019).

42. Tam, C. K. and Webb, J. C.: Dispersion-relation-preserving finite difference schemes for computational acoustics. J. Comput. Phys. **107**, 262–281 (1993).

43. Tan, S. and Shu, C.-W.: Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws. J. Comput. Phys. **229**, 8144–8166 (2010).

44. Tan, S. and Shu, C.-W.: Inverse Lax-Wendroff procedure for numerical boundary conditions of hyperbolic equations: Survey and new developments. Advances in Applied Mathematics, Modeling, and Computational Science, 41–63 (2013).

45. Tan, S., Wang, C., Shu, C.-W., and Ning, J.: Efficient implementation of high order inverse Lax-Wendroff boundary treatment for conservation laws. J. Comput. Phys. **231**, 2510–2527 (2012).

46. Yang, H., Li, F., and Qiu, J.: Dispersion and dissipation errors of two fully discrete discontinuous Galerkin methods. J. Sci. Comput. **55**, 552–574 (2013).

47. Yang, L., Li, S., Jiang, Y., Shu, C.-W., and Zhang, M.: Inverse Lax-Wendroff boundary treatment of discontinuous Galerkin method for 1D conservation laws. Commun. Appl. Math. Comput. **7**, 796–826 (2025).

48. Zhong, X. and Shu, C.-W.: Numerical resolution of discontinuous Galerkin methods for time dependent wave equations. Comput. Method. Appl. M. **200**, 2814–2827 (2011).

49. Zingg, D. W., Lomax, H., and Jurgens, H.: High-accuracy finite-difference schemes for linear wave propagation. SIAM J. Sci. Comput. **17**, 328–346 (1996).

50. Zingg, D. W.: Comparison of high-accuracy finite-difference schemes for linear wave propagation. SIAM J. Sci. Comput. **22**, 476–502 (2000).