

Non-splitting Eulerian-Lagrangian WENO schemes for two-dimensional nonlinear convection-diffusion equations

Xiaofeng Cai

Beijing Normal University

Joint work with
Nanyi Zheng (Udel), Jing-Mei Qiu (Udel), Jianxian Qiu (XMU)

Applied Mathematics and Data Sciences
Shanghai Tech University
July 12, 2024

Outline

Background

Design of the EL-RK-FV-WENO scheme

Numerical tests

Concluding Remarks

Convection-diffusion flow problems

Consider a scalar convection-diffusion equation:

$$u_t + \nabla_{\mathbf{x}} \cdot (\mathbf{F}(u, \mathbf{x}, t)) = \epsilon \Delta u,$$

where $\epsilon \geq 0$.

Applications:

- ▶ fluid dynamics
- ▶ materials science
- ▶ meteorology

Motivation example: Incompressible flow

Incompressible Navier-Stokes equation in vorticity stream function formulation

$$\omega_t + \nabla \cdot (\mathbf{u}\omega) = \frac{1}{Re} \Delta \omega$$
$$\Delta \Phi = \omega, \quad \mathbf{u} = \nabla^\perp \Phi = (-\Phi_y, \Phi_x).$$

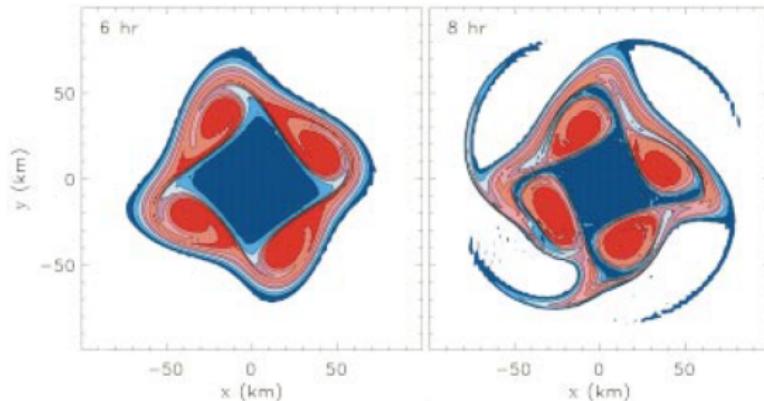


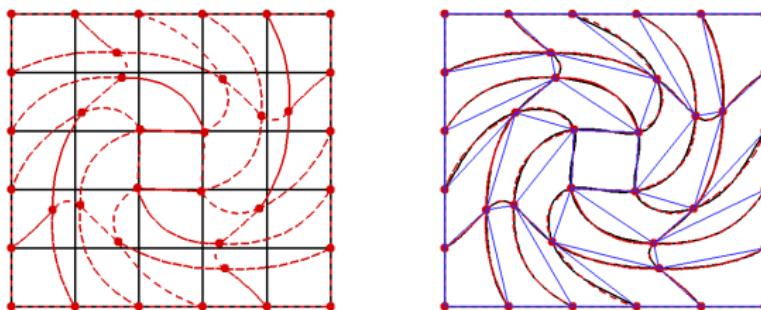
Figure 1: Hurricane eye study. Vorticity. [W. H. Schubert, et al., Journal of the atmospheric sciences, 1999.](#)

Existing Approaches

- ▶ Eulerian Approach
 - ▶ Fixed spatial-temporal mesh
 - ▶ Robust and relatively easy to implement
 - ▶ Time-step constraints
- ▶ Lagrangian Approach
 - ▶ Follows characteristics in time evolution
 - ▶ larger time-step size
 - ▶ The moving Lagrangian mesh can be greatly distorted
- ▶ Semi-Lagrangian Approach
 - ▶ Fixed Eulerian mesh
 - ▶ Accurately track information propagation along characteristics
 - ▶ Allowing for large time-step size

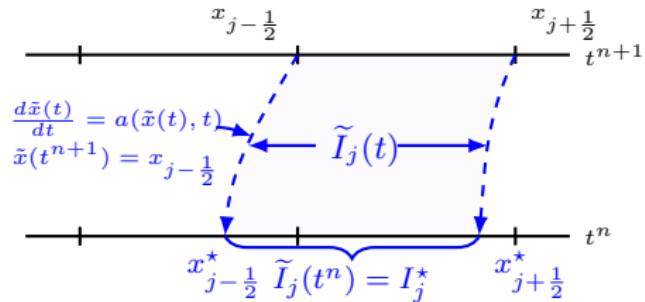
Technical difficulties of high order semi-Lagrangian schemes

- ▶ high order remapping algorithm for high-D problem: quadratic curve approximations to sides of upstream cells.



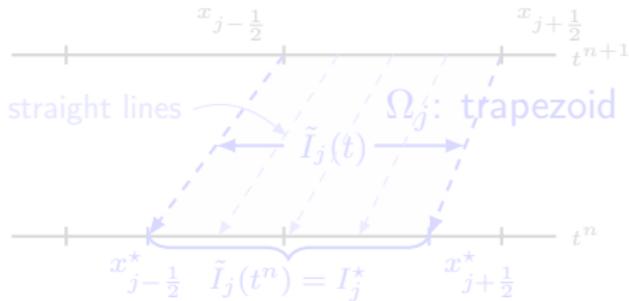
- ▶ General nonlinear problems: characteristics tracing is difficult or impossible.
- ▶ Eulerian-Lagrangian Approach: ELDG, Cai, Qiu, Yang, JCP, 2021.
 - ▶ **linearly Approximately** tracking the characteristics by redefining the space-time region.

Illustration of Eulerian-Lagrangian approach: space-time region



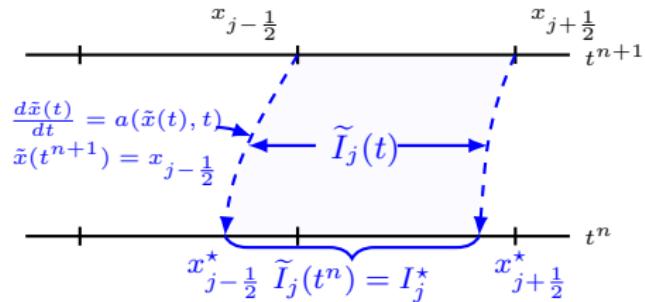
$$\frac{d}{dt} \int_{\tilde{I}_j(t)} u(x, t) dx = 0.$$

ALE formulation; still large time step!

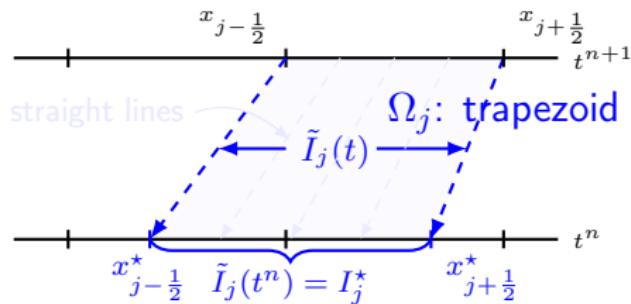


- ▶ Linear function $\alpha(x, t)$ in approximating $a(x, t)$.
- ▶ Feature I: Ω_j : trapezoid; in 2D upstream cells are quadrilaterals.
- ▶ Feature II: straight lines approximating characteristics

Illustration of Eulerian-Lagrangian approach: space-time region



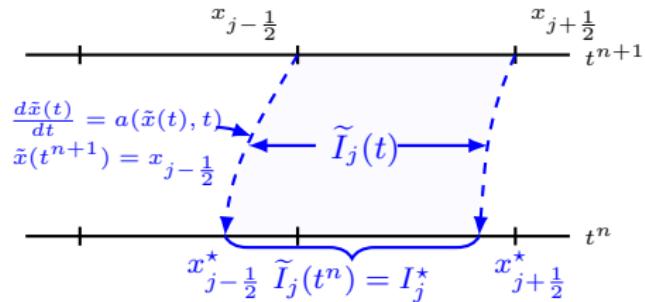
$$\frac{d}{dt} \int_{\tilde{I}_j(t)} u(x, t) dx = 0.$$



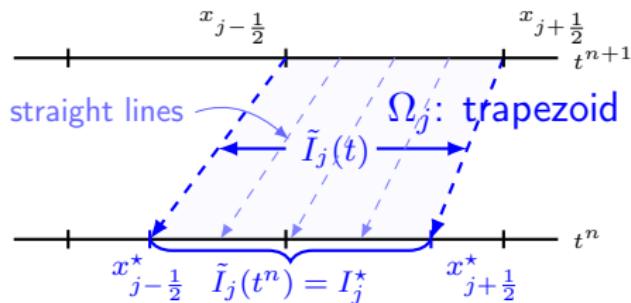
ALE formulation; still large time step!

- ▶ Linear function $\alpha(x, t)$ in approximating $a(x, t)$.
- ▶ Feature I: Ω_j : trapezoid; in 2D upstream cells are quadrilaterals.
- ▶ Feature II: straight lines approximating characteristics

Illustration of Eulerian-Lagrangian approach: space-time region



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



ALE formulation; still large time step!

- ▶ Linear function $\alpha(x, t)$ in approximating $a(x, t)$.
- ▶ Feature I: Ω_j : trapezoid; in 2D upstream cells are quadrilaterals.
- ▶ Feature II: straight lines approximating characteristics

The development of Eulerian-Lagrangian approach

- ▶ Eulerian-Lagrangian WENO method: Huang, Arbogast, J. Qiu, etc.
- ▶ Eulerian-Lagrangian scheme: ELDG, Cai, Qiu, Yang, JCP, 2021; unstructured mesh, Cai, Qiu, SISC 2022.
- ▶ Generalized Eulerian-Lagrangian schemes (splitting): linear system, (Hong, Qiu, JSC 2024)
- ▶ splitting Eulerian-Lagrangian finite volume scheme
 - ▶ WENO for convection-diffusion, Nakao, Chen, Qiu, JCP, 2022)
 - ▶ scalar hyperbolic conservation laws, Yang, Chen, Qiu, JCP, 2022)

Challenges of non-splitting high order Eulerian-Lagrangian Finite Volume Schemes

- ▶ Challenges
 - ▶ The requirement of numerical reconstruction on dynamically varying Lagrangian meshes.
 - ▶ The involvement of intermediate time stages in high-order time discretizations.
 - ▶ The nonlinearity of the scalar convection-diffusion equation.
- ▶ Novel solutions (contributions)
 - ▶ An efficient remapping method is designed to eliminate the need for direct reconstruction on the varying Lagrangian meshes.
 - ▶ A special approach is applied to manage the intermediate time stages in time discretizations.
 - ▶ Low-cost, specialized treatments are designed to address the nonlinearity.

Outline

Background

Design of the EL-RK-FV-WENO scheme

Eulerian-Lagrangian formulation

First-order EL-RK-FV scheme

High-order EL-RK-FV scheme

Numerical tests

Concluding Remarks

Eulerian-Lagrangian formulation

Consider

$$u_t + (a(x, y, t)u)_x + (b(x, y, t)u)_y = \epsilon(u_{xx} + u_{yy}). \quad (1)$$

We define a **modified velocity field** $(\alpha(x, y, t), \beta(x, y, t))$ as follows.

- 1 At $t = t^{n+1}$, $\alpha(x, y, t^{n+1})$ and $\beta(x, y, t^{n+1})$ belong to $Q^1(I_{i,j})$ satisfying

$$\begin{aligned}\alpha(x_{i \pm \frac{1}{2}}, y_{j \pm \frac{1}{2}}, t^{n+1}) &= a(x_{i \pm \frac{1}{2}}, y_{j \pm \frac{1}{2}}, t^{n+1}), \\ \beta(x_{i \pm \frac{1}{2}}, y_{j \pm \frac{1}{2}}, t^{n+1}) &= b(x_{i \pm \frac{1}{2}}, y_{j \pm \frac{1}{2}}, t^{n+1}).\end{aligned} \quad (2)$$

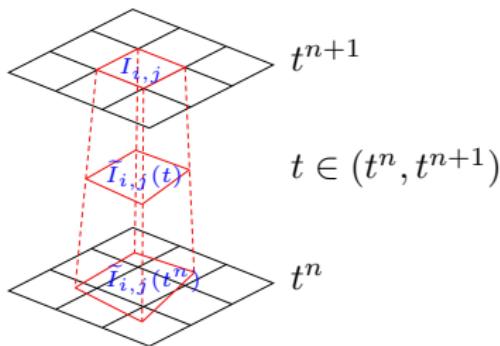
Eulerian-Lagrangian formulation

2 We define a dynamic region

$$\tilde{I}_{i,j}(t) := \{(x, y) | (x, y) = (\tilde{x}(t; (\xi, \eta, t^{n+1})), \tilde{y}(t; (\xi, \eta, t^{n+1}))), (\xi, \eta) \in I_{i,j}\}$$

where $(\tilde{x}(t; (\xi, \eta, t^{n+1})), \tilde{y}(t; (\xi, \eta, t^{n+1})))$ (**modified characteristic line**) represents the straight line going through (ξ, η, t^{n+1}) satisfying

$$\begin{cases} \tilde{x}(t; (\xi, \eta, t^{n+1})) = \xi + (t - t^{n+1})\alpha(\xi, \eta, t^{n+1}), \\ \tilde{y}(t; (\xi, \eta, t^{n+1})) = \eta + (t - t^{n+1})\beta(\xi, \eta, t^{n+1}). \end{cases}$$



Eulerian-Lagrangian formulation

3 For $t \in [t^n, t^{n+1})$ and $(\tilde{x}(t; (\xi, \eta, t^{n+1})), \tilde{y}(t; (\xi, \eta, t^{n+1}))) \in \tilde{I}_{i,j}(t)$,

$$\begin{cases} \alpha(\tilde{x}(t; (\xi, \eta, t^{n+1})), \tilde{y}(t; (\xi, \eta, t^{n+1}))) = \alpha(\xi, \eta, t^{n+1}), \\ \beta(\tilde{x}(t; (\xi, \eta, t^{n+1})), \tilde{y}(t; (\xi, \eta, t^{n+1}))) = \beta(\xi, \eta, t^{n+1}). \end{cases} \quad (3)$$

With the definition above, we have an ALE formulation

$$\boxed{\frac{d}{dt} \iint_{\tilde{I}_{i,j}(t)} u dx dy = - \int_{\partial \tilde{I}_{i,j}(t)} (\textcolor{blue}{a} - \textcolor{red}{\alpha}, \textcolor{blue}{b} - \textcolor{red}{\beta}) u \cdot \mathbf{n} ds + \epsilon \iint_{\tilde{I}_{i,j}(t)} \Delta u dx dy.}$$

Eulerian-Lagrangian formulation

We define $\mathbf{F}(u, x, y, t) := ((a - \alpha)u, (b - \beta)u)$ and provide the concise EL-FV formulation:

$$\begin{aligned} \frac{d}{dt} \iint_{\tilde{I}_{i,j}(t)} u(x, y, t) dx dy &= - \int_{\partial \tilde{I}_{i,j}(t)} \mathbf{F}(u, x, y, t) \cdot \mathbf{n} ds + \epsilon \iint_{\tilde{I}_{i,j}(t)} \Delta u dx dy \\ &:= \mathcal{F}_{i,j}(u; t) + \mathcal{G}_{i,j}(u; t). \end{aligned} \tag{4}$$

the semi-discretization of the Eulerian-Lagrangian scheme

To evaluate the right-hand side (RHS) of (4), we introduce the following notation for semi-discretization:

$$\frac{d\tilde{u}_{i,j}(t)}{dt} = \tilde{\mathcal{F}}_{i,j}(\bar{\mathbf{U}}; t) + \tilde{\mathcal{G}}_{i,j}(\bar{\mathbf{U}}; t), \quad (5)$$

where

- ▶ the notation $\tilde{\cdot}$ specifies that the integral value corresponds to the characteristic spatial region $\tilde{I}_{i,j}(t)$,
- ▶ $\tilde{u}_{i,j}(t)$ approximates $\iint_{\tilde{I}_{i,j}(t)} u(x, y, t) dx dy$,
- ▶ $\tilde{\mathcal{F}}_{i,j}(\bar{\mathbf{U}}; t)$ approximates $\mathcal{F}_{i,j}(u; t)$,
- ▶ $\tilde{\mathcal{G}}_{i,j}(\bar{\mathbf{U}}; t)$ approximates $\mathcal{G}_{i,j}(u; t)$,
- ▶ $\bar{\mathbf{U}} := (\bar{u}_{i,j}(t))_{N_x N_y}$ represents the finite volumes such that

$$\bar{u}_{i,j}(t) \approx \frac{1}{|I_{i,j}|} \iint_{I_{i,j}} u(x, y, t) dx dy. \quad (6)$$

the semi-discretization of the Eulerian-Lagrangian scheme

Similar to $\bar{\mathbf{U}}$, we can also represent (5) globally as follows:

$$\frac{d\tilde{\mathbf{U}}(t)}{dt} = \tilde{\mathcal{F}}(\bar{\mathbf{U}}; t) + \tilde{\mathcal{G}}(\bar{\mathbf{U}}; t), \quad (7)$$

where $\tilde{\mathcal{F}} := (\tilde{\mathcal{F}}_{i,j})_{N_x N_y}$ and $\tilde{\mathcal{G}} := (\tilde{\mathcal{G}}_{i,j})_{N_x N_y}$.

First-order EL-RK-FV scheme

Coupling (7) with the first-order forward-backward Euler IMEX method in¹ yields the first-order EL-RK-FV scheme:

$$\mathbf{M}\bar{\mathbf{U}}^{n+1} = \tilde{\mathbf{U}}^n + \Delta t \tilde{\mathcal{F}}(\bar{\mathbf{U}}^n, t^n) + \Delta t \left(\epsilon \mathbf{M} \bar{\mathbf{D}} \bar{\mathbf{U}}^{n+1} \right), \quad (8)$$

where

- ▶ \mathbf{M} is a diagonal matrix such that $\mathbf{M}\bar{\mathbf{U}}^{n+1} = (|I_{i,j}| \bar{u}_{i,j}^{n+1})_{N_x N_y}$.
- ▶ $\tilde{\mathbf{U}}^n$ approximates $\left(\iint_{\tilde{I}_{i,j}(t^n)} u(x, y, t^n) dx dy \right)_{N_x N_y}$,
- ▶ \mathbf{D} is a differential matrix such that

$$\mathbf{D}\bar{\mathbf{U}}^{n+1} \text{ approximates } \left(\frac{1}{|I_{i,j}|} \iint_{I_{i,j}} \Delta u dx dy \right)_{N_x N_y}.$$

¹Uri M. Ascher, Steven J. Ruuth, and Raymond J. Spiteri. "Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations". In: *Applied Numerical Mathematics* 25.2 (1997), pp. 151–167. ISSN: 01689274.

First-order EL-RK-FV scheme

As shown, the diffusion term in (8) is implicit. Consequently, $\bar{\mathbf{U}}^{n+1}$ is obtained by solving the following linear system:

$$\mathbf{M}(\mathbf{I} - \Delta t \epsilon \mathbf{D}) \bar{\mathbf{U}}^{n+1} = \tilde{\mathbf{U}}^n + \Delta t \tilde{\mathcal{F}}(\bar{\mathbf{U}}^n, t^n). \quad (9)$$

Remark 1

(*Empirical time-step constraint of the convection term for stability*)

$$\Delta t \sim \sqrt{\min\{\Delta x, \Delta y\}}. \quad (10)$$

High-order EL-RK-FV-WENO schemes

- ▶ How to approximate the $\tilde{\mathcal{F}}$ and $\tilde{\mathcal{G}}$?
 - ▶ High order WENO reconstruction and diffusion term in **Part 1** and **Part 2**
- ▶ The coupling of the semi-discretization with high-order IMEX RK methods is discussed in **Part 3**

Part 1: Flux approximation

Step 1: Construct a piecewise polynomial with respect to the Eulerian mesh, $\{I_{i,j}\}$.

We construct a piecewise reconstruction polynomial $u^{\text{WENO}}(x, y)$ such that

- ▶ $u^{\text{WENO}}(x, y)|_{I_{i,j}} = u_{i,j}^{\text{WENO}}(x, y)$ with $u_{i,j}^{\text{WENO}} \in P^2(I_{i,j})$ for all i, j ,
- ▶ each $u_{i,j}^{\text{WENO}}$ is constructed based on the information $\bar{u}_{i,j}(t)$ along with its eight neighbor finite volumes,
- ▶ $\iint_{I_{i,j}} u^{\text{WENO}}(x, y) dx dy = \Delta x_i \Delta y_j \bar{u}_{i,j}(t)$ for all i, j ,
- ▶ $u^{\text{WENO}}(x, y) = u(x, y, t) + O(\Delta x^3)$, $(x, y) \in \Omega$.

Part 1: Flux approximation

For convenience, we assume $\Delta x_i \equiv \Delta x$ and $\Delta y_j \equiv \Delta y$ for all i, j . We define $\mu_i(x) := \frac{x - x_i}{\Delta x}$, $\nu_j(y) := \frac{y - y_j}{\Delta y}$ and introduce a set of local orthogonal polynomials as $\{P_l^{(i,j)}(x, y)\}$ for a given cell $I_{i,j}$:

$$P_1^{(i,j)} := 1, \quad P_2^{(i,j)} := \mu_i(x), \quad P_3^{(i,j)} := \nu_j(y),$$

$$P_4^{(i,j)} := \mu_i^2(x) - \frac{1}{12}, \quad P_5^{(i,j)} := \mu_i(x)\nu_j(y), \quad P_6^{(i,j)} := \nu_j^2(y) - \frac{1}{12}.$$

Part 2.1: Flux approximation

We assume that $\bar{u}_5 := \bar{u}_{i,j}$ and $I_5 := I_{i,j}$, while $\{u_s\}$ and $\{I_s\}$ represent corresponding cell averages and Eulerian cells based on the serial number in Figure 2. The reconstruction procedure is performed as follows:

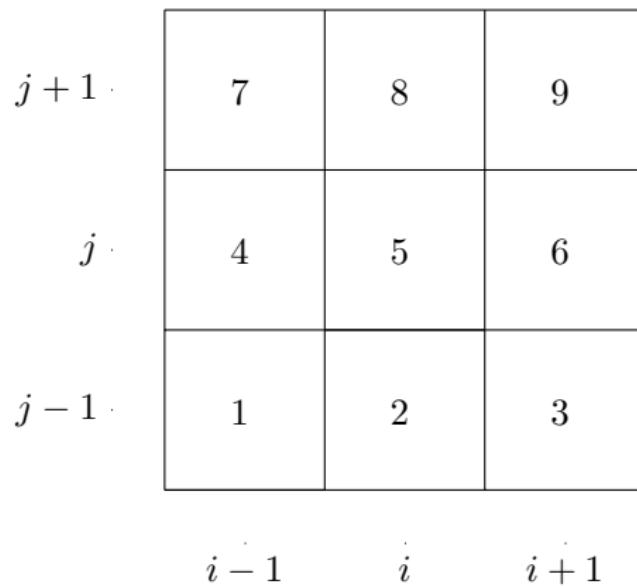


Figure 2: Stencil for the 3rd-order WENO-ZQ reconstruction.

Part 2.1: Flux approximation

Step 1.1 Construct a quadratic polynomial $q_0(x, y) = \sum_{l=1}^6 a_l^{q_0} P_l^{(i,j)}(x, y)$ using a special least-squares procedure. We define

$$V := \{p(x, y) \in P^2(I_{i,j}) \mid \frac{1}{\Delta x \Delta y} \int_{I_s} p(x, y) dx dy = \bar{u}_s, s = 2, 4, 5, 6, 8\},$$

$$E(p(x, y)) := \left[\sum_{s=1,3,7,9} \left(\frac{1}{\Delta x \Delta y} \int_{I_s} p(x, y) dx dy - \bar{u}_s \right)^2 \right]^{\frac{1}{2}}.$$

Then, we determine that $q_0(x, y)$ is the unique polynomial satisfying:

$$E(q_0(x, y)) = \min_{p \in V} E(p(x, y)). \quad (11)$$

Part 2.1: Flux approximation

Step 1.2 Construct eight linear polynomials

$$\{q_k(x, y)\}_{k=1}^8 = \left\{ \sum_{l=1}^3 a_l^{q_k} P_l^{(i,j)}(x, y) \right\} \text{ satisfying:}$$

$$\frac{1}{\Delta x \Delta y} \iint_{I_5} q_k(x, y) dx dy = \bar{u}_5 \quad \text{for } k = 1, 2, \dots, 8, \quad (12)$$

and

$$\frac{1}{\Delta x \Delta y} \iint_{I_s} q_k(x, y) dx dy = \bar{u}_s, \quad (13)$$

where

$$s = 1, 2 \quad \text{for } k = 1; \quad s = 2, 3 \quad \text{for } k = 2;$$

$$s = 3, 6 \quad \text{for } k = 3; \quad s = 6, 9 \quad \text{for } k = 4;$$

$$s = 8, 9 \quad \text{for } k = 5; \quad s = 7, 8 \quad \text{for } k = 6;$$

$$s = 4, 7 \quad \text{for } k = 7; \quad s = 1, 4 \quad \text{for } k = 8.$$

Part 2.1: Flux approximation

Step 1.3 Rewrite $q_0(x, y)$ as follows:

$$q_0(x, y) = \gamma_0 \left(\frac{1}{\gamma_0} q_0(x, y) - \sum_{k=1}^8 \frac{\gamma_k}{\gamma_0} q_k(x, y) \right) + \sum_{k=1}^8 \gamma_k q_k(x, y),$$

where $\{\gamma_k\}_{k=0}^8$ is a set of positive linear weights with their sum being 1. The linear weights control the balance between optimal reconstruction accuracy and avoiding numerical oscillation. In our numerical tests, we set $\gamma_0 = 0.6$ and $\gamma_1 = \dots = \gamma_8 = 0.05$ for such balance.

Part 2.1: Flux approximation

Step 1.4 Compute the smoothness indicators of $\{q_k(x, y)\}_{k=0}^8$:

$$\beta_0 = \frac{1}{\Delta x \Delta y} \sum_{l_1+l_2 \leq 2} \iint_{I_5} \left(\Delta x^{l_1} \Delta y^{l_2} \frac{\partial^{|l_1+l_2|}}{\partial l_1 \partial l_2} q_0(x, y) \right)^2 dx dy,$$

$$\beta_k = \frac{1}{\Delta x \Delta y} \sum_{l_1+l_2 \leq 1} \iint_{I_5} \left(\Delta x^{l_1} \Delta y^{l_2} \frac{\partial^{|l_1+l_2|}}{\partial l_1 \partial l_2} q_k(x, y) \right)^2 dx dy,$$

for $k = 1, \dots, 8$.

The explicit expressions of $\{\beta_k\}_{k=0}^8$ are given by

$$\beta_0 = (a_2^{q_0})^2 + (a_3^{q_0})^2 + \frac{13}{3} (a_4^{q_0})^2 + \frac{7}{6} (a_5^{q_0})^2 + \frac{13}{3} (a_6^{q_0})^2, \quad (14)$$

$$\beta_k = (a_2^{q_k})^2 + (a_3^{q_k})^2 \quad \text{for } k = 1, 2, \dots, 8.$$

Part 2.1: Flux approximation

Step 1.5 Compute the nonlinear weights $\{\omega_k\}_{k=0}^8$: $\omega_k = \frac{\tilde{\omega}_k}{\sum_{l=0}^8 \tilde{\omega}_l}$,

where $\tilde{\omega}_k = \gamma_k \left(1 + \frac{\tau^{\frac{5}{4}}}{\beta_k + \epsilon} \right)$ for $k = 0, 1, \dots, 8$

with $\tau = \frac{\sum_{k=1}^8 |\beta_0 - \beta_k|}{8}$.

When the exact solution is smooth over the entire large stencil

$\bigcup_{s=1}^9 I_s$, we can prove

$$\omega_k = \begin{cases} \gamma_k \left(1 + O \left(\Delta^{\frac{3}{2}} \right) \right), & \text{if } Du|_{(x_i, y_j)} \neq 0 \text{ and } D^2u|_{(x_i, y_j)} \neq 0, \\ \gamma_k (1 + O(\Delta x)), & \text{if } Du|_{(x_i, y_j)} = 0 \text{ and } D^2u|_{(x_i, y_j)} \neq 0, \end{cases}$$

by Taylor expansion.

Part 2.1: Flux approximation

Step 1.6 Construct the final reconstruction polynomial as follows:

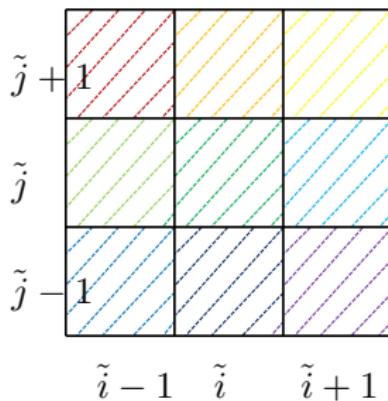
$$u_{i,j}^{\text{WENO}}(x, y) = \omega_0 \left(\frac{1}{\gamma_0} q_0(x, y) - \sum_{k=1}^8 \frac{\gamma_k}{\gamma_0} q_k(x, y) \right) + \sum_{k=1}^8 \omega_k q_k(x, y).$$

Eventually, we define that $u^{\text{WENO}}(x, y)$ is the piecewise polynomial satisfying:

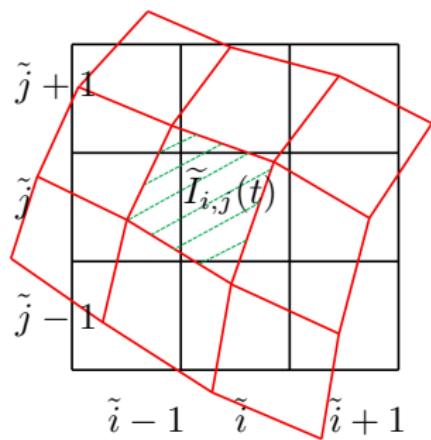
$$u^{\text{WENO}}(x, y) = u_{i,j}^{\text{WENO}}(x, y) \quad (x, y) \in I_{i,j} \quad \text{for all } i, j. \quad (15)$$

Part 2.1: Flux approximation

Step 2: Construct a piecewise polynomial on the Lagrangian mesh $\{\tilde{I}_{i,j}(t)\}$ (remapping).



(a) Piecewise polynomial
 $u^{\text{WENO}}(x, y)$



(b) $\tilde{u}_{i,j}^{\text{WENO}}(x, y)$

Figure 3: Schematic illustrations of $u^{\text{WENO}}(x, y)$ and $\tilde{u}_{i,j}^{\text{WENO}}(x, y)$.

Part 2.1: Flux approximation

The basic idea of this remapping step is that we want to conduct an efficient modification to $u^{\text{WENO}}(x, y)$ instead of involving a new reconstruction on the Lagrangian mesh. The resulting new piecewise polynomial, denoted by $\tilde{u}^{\text{WENO}}(x, y)$, satisfies the following conditions:

- ▶ $\tilde{u}^{\text{WENO}}(x, y)|_{\tilde{I}_{i,j}(t)} = \tilde{u}_{i,j}^{\text{WENO}}(x, y)$ with $\tilde{u}_{i,j}^{\text{WENO}} \in P^2(\tilde{I}_{i,j}(t))$ for all i, j ,
- ▶ $\iint_{\tilde{I}_{i,j}(t)} \tilde{u}^{\text{WENO}}(x, y) dx dy = \iint_{\tilde{I}_{i,j}(t)} u^{\text{WENO}}(x, y) dx dy$ for all i, j ,
- ▶ $\tilde{u}^{\text{WENO}}(x, y) = u(x, y, t) + O(\Delta x^3)$, $(x, y) \in \Omega$.

Part 2.1: Flux approximation

Step 2.1: Compute the exact “mass” of $u^{\text{WENO}}(x, y)$ over $\tilde{I}_{i,j}(t)$, i.e.

$$\tilde{u}_{i,j} := \iint_{\tilde{I}_{i,j}(t)} u^{\text{WENO}}(x, y) dx dy. \quad (16)$$

The integrand in Equation (16) is discontinuous over $\tilde{I}_{i,j}(t)$. A numerical integral for Equation (16) contains two basic steps. First, a clipping procedure is conducted to divide $\tilde{I}_{i,j}(t)$ into smaller polygons such that $u^{\text{WENO}}(x, y)$ is continuous in each of them. Second, a numerical integration is conducted in each polygon, and the results are summed to obtain the final integral. Following these two basic steps, there are different implementation methods^{2,3,4}.

² Juan Cheng and Chi-Wan Shu. “A high order accurate conservative remapping method on staggered meshes”. In: *Applied Numerical Mathematics. Spectral Methods in Computational Fluid Dynamics* 58.7 (2008), pp. 1042–1060. ISSN: 0168-9274.

³ Peter H. Lauritzen, Ramachandran D. Nair, and Paul A. Ullrich. “A conservative semi-Lagrangian multi-tracer transport scheme (CSLAM) on the cubed-sphere grid”. In: *Journal of Computational Physics* 229.5 (2010), pp. 1401–1424. ISSN: 00219991.

⁴ Nanyi Zheng et al. “A fourth-order conservative semi-Lagrangian finite volume WENO scheme without operator splitting for kinetic and fluid simulations”. In: *Computer Methods in Applied Mechanics and Engineering* 395 (2022), p. 114973. ISSN: 0045-7825.

Part 2.1: Flux approximation

Step 2.2: Find all Eulerian cells that intersect with $\tilde{I}_{i,j}(t)$. We define that

$$\mathcal{K} := \{(p, q) | I_{p,q} \cap \tilde{I}_{i,j}(t) \neq \emptyset\}. \text{ For instance, in Figure 3b, } \mathcal{K} = \{(i-1, j+1), (i, j+1), (\tilde{i}-1, \tilde{j}), (\tilde{i}, \tilde{j}), (\tilde{i}, \tilde{j}-1)\}.$$

Step 2.3: Compute the integrals of the candidate P^2 polynomials over $\tilde{I}_{i,j}(t)$, i.e.

$$\tilde{u}_{i,j}^{p,q} := \iint_{\tilde{I}_{i,j}(t)} u_{p,q}^{\text{WENO}}(x, y) dx dy, \quad (p, q) \in \mathcal{K}. \quad (17)$$

Step 2.4: Choose the index (\tilde{p}, \tilde{q}) such that $|\tilde{u}_{i,j}^{p,q} - \tilde{u}_{i,j}|$ reaches the minimum, i.e.

$$|\tilde{u}_{i,j}^{\tilde{p}, \tilde{q}} - \tilde{u}_{i,j}| = \min_{(p, q) \in \mathcal{K}} \{|\tilde{u}_{i,j}^{p,q} - \tilde{u}_{i,j}|\}. \quad (18)$$

Step 2.5: Define a P^2 polynomial, denoted by $\tilde{u}_{i,j}^{\text{WENO}}(x, y)$, on $\tilde{I}_{i,j}(t)$ such that

$$\tilde{u}_{i,j}^{\text{WENO}}(x, y) := u_{(\tilde{p}, \tilde{q})}^{\text{WENO}}(x, y)|_{\tilde{I}_{i,j}(t)} - \frac{1}{|\tilde{I}_{i,j}(t)|} \tilde{u}_{i,j}^{(\tilde{p}, \tilde{q})} + \frac{1}{|\tilde{I}_{i,j}(t)|} \tilde{u}_{i,j}.$$

Part 2.1: Flux approximation

Step 3: Construct the final flux approximation.

The final flux approximation is given by the following conservative formulation

$$\tilde{\mathcal{F}}_{i,j}(\bar{\mathbf{U}}; t) := - \int_{\partial \tilde{I}_{i,j}(t)} \hat{F}(\tilde{u}^{\text{WENO}}, x, y, t) ds, \quad (19)$$

where

$$\hat{F}(\tilde{u}^{\text{WENO}}, x, y, t) := W(x, y, t) \tilde{u}_{i,j}^{\text{up}}(x, y) \quad (20)$$

with

$$W(x, y, t) := (a - \alpha, b - \beta) \cdot \mathbf{n} \quad (21)$$

and

$$\tilde{u}_{i,j}^{\text{up}}(x, y) := \begin{cases} \tilde{u}_{i,j}^{\text{WENO}}(x, y), & W > 0, \\ \tilde{u}_{i,j}^{\text{WENO,ext}}(x, y), & W \leq 0. \end{cases} \quad (22)$$

Here, $\tilde{u}_{i,j}^{\text{WENO,ext}}(x, y)$ are the exterior solution with respect to corresponding edge of $\tilde{I}_{i,j}(t)$.

Part 2.2: Approximation of the diffusion term

Step 1: Recover high-order finite volume information of Δu .

We recover

$$\Delta \bar{\mathbf{U}} := \mathbf{D} \bar{\mathbf{U}}, \quad (23)$$

where \mathbf{D} is the differential matrix assembled by the following local operators

$$\bar{\Delta u}_{i,j} := \begin{bmatrix} -\frac{1}{12} & \frac{4}{3} & -\frac{5}{2} & \frac{4}{3} & -\frac{1}{12} \end{bmatrix} \left(\frac{1}{\Delta x^2} \begin{bmatrix} \bar{u}_{i-2,j} \\ \bar{u}_{i-1,j} \\ \bar{u}_{i,j} \\ \bar{u}_{i+1,j} \\ \bar{u}_{i+2,j} \end{bmatrix} + \frac{1}{\Delta y^2} \begin{bmatrix} \bar{u}_{i,j-2} \\ \bar{u}_{i,j-1} \\ \bar{u}_{i,j} \\ \bar{u}_{i,j+1} \\ \bar{u}_{i,j+2} \end{bmatrix} \right)$$

for all i, j , and corresponding boundary conditions.

Step 2: Recover a piecewise reconstruction polynomial of Δu .

We utilize the same polynomial $q_0(x, y)$ in **step 1.1** of the flux approximation as the reconstruction formula and denote the final piecewise polynomial by $\Delta u^{\text{rec}}(x, y)$.

Part 2.2: Approximation of the diffusion term

Step 3: Compute the integral of $\Delta u^{\text{rec}}(x, y)$ as the final approximation.

$$\begin{aligned}\tilde{\mathcal{G}}_{i,j}(\bar{\mathbf{U}}; t) &= \epsilon \iint_{\tilde{I}_{i,j}(t)} \Delta u dx dy \\ &\approx \epsilon \iint_{\tilde{I}_{i,j}(t)} \Delta u^{\text{rec}}(x, y) dx dy := \epsilon \widetilde{\Delta u}_{i,j}(\bar{\mathbf{U}}; t) \quad \text{for all } i, j,\end{aligned}$$

where the integral of piecewise polynomial is accomplished by the method of our previous work⁵.

⁵Nanyi Zheng et al. "A fourth-order conservative semi-Lagrangian finite volume WENO scheme without operator splitting for kinetic and fluid simulations". In: *Computer Methods in Applied Mechanics and Engineering* 395 (2022), p. 114973. ISSN: 0045-7825.

Part 2.3: High-order IMEX RK temporal discretization

An IMEX RK scheme can be represented by the following two butcher tables:

Implicit Scheme						Explicit Scheme					
0	0	0	0	...	0	0	0	0	...	0	0
c_1	0	a_{11}	0	...	0	c_1	\hat{a}_{21}	0	0	...	0
c_2	0	a_{21}	a_{22}	...	0	c_2	\hat{a}_{31}	\hat{a}_{32}	0	...	0
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	\vdots
c_s	0	a_{s1}	a_{s2}	...	a_{ss}	$c_{\sigma-1}$	$\hat{a}_{\sigma,1}$	$\hat{a}_{\sigma,2}$	$\hat{a}_{\sigma,3}$...	0
	0	b_1	b_2	...	b_s		\hat{b}_1	\hat{b}_2	\hat{b}_3	...	\hat{b}_σ

A triplet (s, σ, p) is used to demonstrate that the IMEX scheme uses an s -stage implicit scheme and a σ -stage explicit scheme achieving p th-order accuracy. Here, $\sigma = s + 1$ unless $c_s = 1$ and $\hat{a}_{\sigma,j} = \hat{b}_j$ for all $j = 1, 2, \dots, \sigma$, in which case $\sigma = s$.

Part 2.3: High-order IMEX RK temporal discretization

The butcher table of first-order IMEX scheme in (8), which is also called IMEX(1,1,1) in⁶, is

Implicit Scheme			Explicit Scheme		
0	0	0	0	0	0
1	0	1	1	1	0
	0	1		1	0

The second-order IMEX(1,2,2) scheme in⁶ is represented by the following butcher tables:

Implicit Scheme			Explicit Scheme		
0	0	0	0	0	0
$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0
	0	1		0	1

⁶Uri M. Ascher, Steven J. Ruuth, and Raymond J. Spiteri. "Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations". In: *Applied Numerical Mathematics* 25.2 (1997), pp. 151–167. ISSN: 01689274.

Part 2.3: High-order IMEX RK temporal discretization

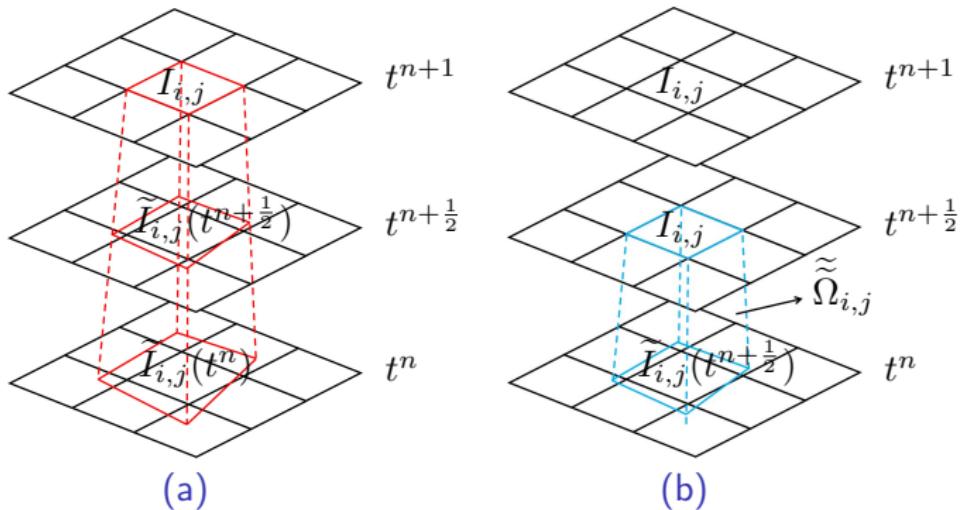


Figure 4: Schematic illustration of full discretization with IMEX(1,2,2).

Part 2.3: High-order IMEX RK temporal discretization

The second-order fully discrete scheme is provided as follows (see Figure 4):

$$\mathbf{M}\bar{\mathbf{U}}^{(1)} = \tilde{\mathbf{U}}^n + \frac{1}{2}\Delta t \tilde{\mathcal{F}}(\bar{\mathbf{U}}^n, t^n) + \frac{1}{2}\Delta t \left(\epsilon \mathbf{M} \mathbf{D} \bar{\mathbf{U}}^{(1)} \right), \quad (24)$$

$$\mathbf{M}\bar{\mathbf{U}}^{n+1} = \tilde{\mathbf{U}}^n + \Delta t \tilde{\mathcal{F}}(\bar{\mathbf{U}}^{(1)}, t^{n+\frac{1}{2}}) + \Delta t \tilde{\mathcal{G}}(\bar{\mathbf{U}}^{(1)}, t^{n+\frac{1}{2}}), \quad (25)$$

where

- ▶ the notation $\tilde{\cdot}$ is used to indicate that the corresponding values or operations refer to specific slices of the characteristic region $\tilde{\Omega}_{i,j}$,
- ▶ $\tilde{\mathbf{U}}^n := \left(\iint_{\tilde{\Omega}_{i,j}(t^{n+\frac{1}{2}})} \mathcal{W}(\bar{\mathbf{U}}^n) dx dy \right)_{N_x N_y}$ (see Figure 4b).

Part 2.4: Extension for nonlinear problems

Consider

$$u_t + (f_1(u))_x + (f_2(u))_y = \epsilon(u_{xx} + u_{yy}). \quad (26)$$

1. Redesign the modified velocity field.

The construction of the original modified velocity field requires the exact velocity field at $t = t^{n+1}$, which is unknown for nonlinear models. We redesign that $(\alpha(x, y, t), \beta(x, y, t))$ is defined by first applying interpolation at $t = t^{n+1}$ such that

$$\begin{aligned}\alpha(x_{i \pm \frac{1}{2}}, y_{j \pm \frac{1}{2}}, t^{n+1}) &= f'_1(\mathcal{W}(\bar{\mathbf{U}}^n))|_{(x_{i \pm \frac{1}{2}}, y_{j \pm \frac{1}{2}})}, \\ \beta(x_{i \pm \frac{1}{2}}, y_{j \pm \frac{1}{2}}, t^{n+1}) &= f'_2(\mathcal{W}(\bar{\mathbf{U}}^n))|_{(x_{i \pm \frac{1}{2}}, y_{j \pm \frac{1}{2}})},\end{aligned} \quad (27)$$

This modified velocity field can still give us the same time-step constraint $\Delta t \sim \sqrt{\min\{\Delta x, \Delta y\}}$ if we apply similar analysis as in Remark 1.

Part 2.4: Extension for nonlinear problems

2. Recover the velocity fields at intermediate time levels.

The velocity field for $t \in (t^n, t^{n+1}]$, which is required for evaluating numerical fluxes, is unknown for nonlinear models. For this issue, we simply use the predicted solutions at the intermediate time levels to recover corresponding velocity fields. For example, in (25), the exact velocity field $(a(x, y, t^{n+\frac{1}{2}}), b(x, y, t^{n+\frac{1}{2}}))$ is replaced with $\left(f'_1(\mathcal{W}(\bar{\mathbf{U}}^{(1)})), f'_2(\mathcal{W}(\bar{\mathbf{U}}^{(1)}))\right)$ for $\tilde{\mathcal{F}}\left(\bar{\mathbf{U}}^{(1)}, t^{n+\frac{1}{2}}\right)$.

Outline

Background

Design of the EL-RK-FV-WENO scheme

Numerical tests

Concluding Remarks

Numerical tests

- ▶ The time-steps in the following are defined by:

$$\Delta t = \frac{\text{CFL}}{\frac{\max\{|f'(u)|\}}{\Delta x} + \frac{\max\{|g'(u)|\}}{\Delta y}},$$

where $(f'(u), g'(u))$ represents the corresponding velocity field.

- ▶ For pure convection simulation, we use the third-order Runge-Kutta temporal discretization with the following butcher table:

0	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	0	0
1	-1	2	0
	$\frac{1}{6}$	$\frac{2}{3}$	$\frac{1}{6}$

Numerical tests

For convection-diffusion simulation, we apply IMEX(2,3,3) in⁷ a third-order scheme with the following butcher tables:

Implicit Scheme			
0	0	0	0
γ	0	γ	0
$1 - \gamma$	0	$1 - 2\gamma$	γ
	0	$\frac{1}{2}$	$\frac{1}{2}$

Explicit Scheme			
0	0	0	0
γ	γ	0	0
$1 - \gamma$	$\gamma - 1$	$2(1 - \gamma)$	0
	0	$\frac{1}{2}$	$\frac{1}{2}$

where $\gamma = (3 + \sqrt{3})/6$.

⁷Uri M. Ascher, Steven J. Ruuth, and Raymond J. Spiteri. "Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations". In: *Applied Numerical Mathematics* 25.2 (1997), pp. 151–167. ISSN: 01689274.

Linear models: pure advection equation

Example 1

(Swirling deformation flow) Consider the following equation:

$$u_t - \left(2\pi \cos^2\left(\frac{x}{2}\right) \sin(y) g(t) u\right)_x + \left(2\pi \sin(x) \cos^2\left(\frac{y}{2}\right) g(t) u\right)_y = 0, \quad (28)$$
$$x, y \in [-\pi, \pi],$$

where $g(t) = \cos(\pi t/T)$ with $T = 1.5$.

We first consider (28) with the following smooth initial condition:

$$u(x, y, 0) = \begin{cases} r_0^b \cos\left(\frac{r^b(\mathbf{x})\pi}{2r_0^b}\right)^6 & \text{if } r^b(\mathbf{x}) < r_0^b, \\ 0, & \text{otherwise,} \end{cases} \quad (29)$$

where $r_0^b = 0.3\pi$, $r^b(\mathbf{x}) = \sqrt{(x - x_0^b)^2 + (y - y_0^b)^2}$ and the center of the cosine bell $(x_0^b, y_0^b) = (0.3\pi, 0)$.

Spatial accuracy

Table 1: (Swirling deformation flow) L^1 , L^2 , and L^∞ errors and corresponding orders of accuracy of the EL-RK-FV-WENO scheme for (28) with initial condition (29) at $t = 1.5$ with CFL = 1.

mesh	L^1 error	order	L^2 error	order	L^∞ error	order
20×20	8.37E-03	—	4.51E-02	—	7.40E-01	—
40×40	3.85E-03	1.12	2.52E-02	0.84	4.74E-01	0.64
80×80	1.16E-03	1.72	8.06E-03	1.64	1.66E-01	1.52
160×160	2.22E-04	2.39	1.50E-03	2.43	3.24E-02	2.35
320×320	3.01E-05	2.88	2.03E-04	2.88	4.86E-03	2.74

Temporal accuracy and numerical stability

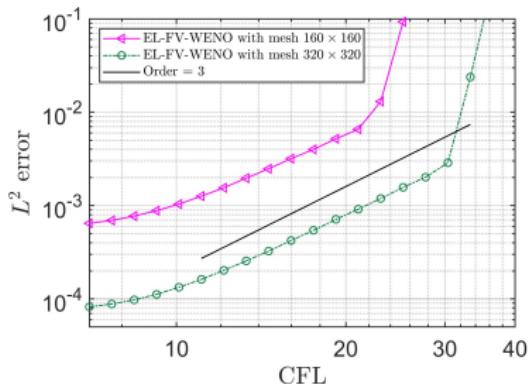


Figure 5: (Swirling deformation flow) Log-log plot of CFL numbers versus L^2 errors with fixed meshes 160×160 and 320×320 at $t = 1.5$ of the EL-RK-FV-WENO scheme.

Linear models

To validate the non-oscillatory nature of the proposed WENO reconstruction, we consider a discontinuous initial condition featuring a cylinder with a notch, a cone, and a smooth bell as shown in Figure 6.

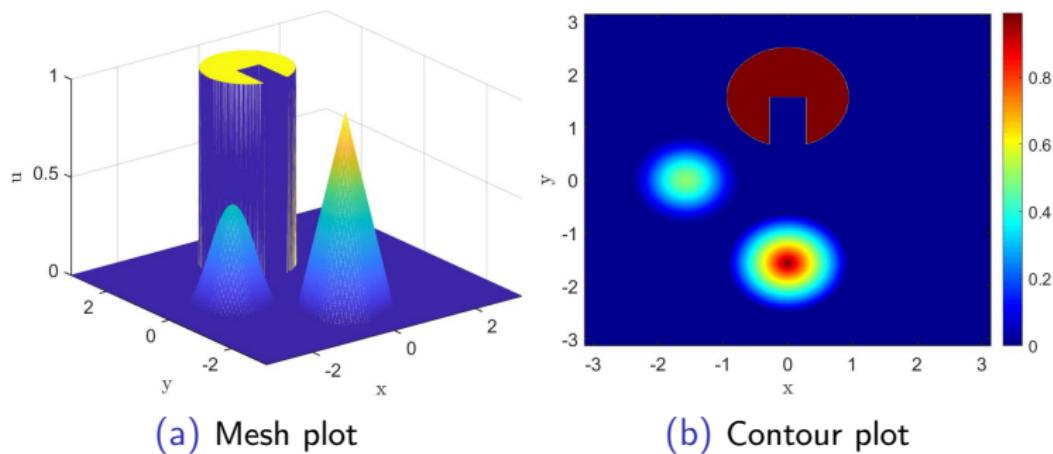


Figure 6: (Swirling deformation flow) Mesh plot and contour plot of a discontinuous initial condition.

Linear models

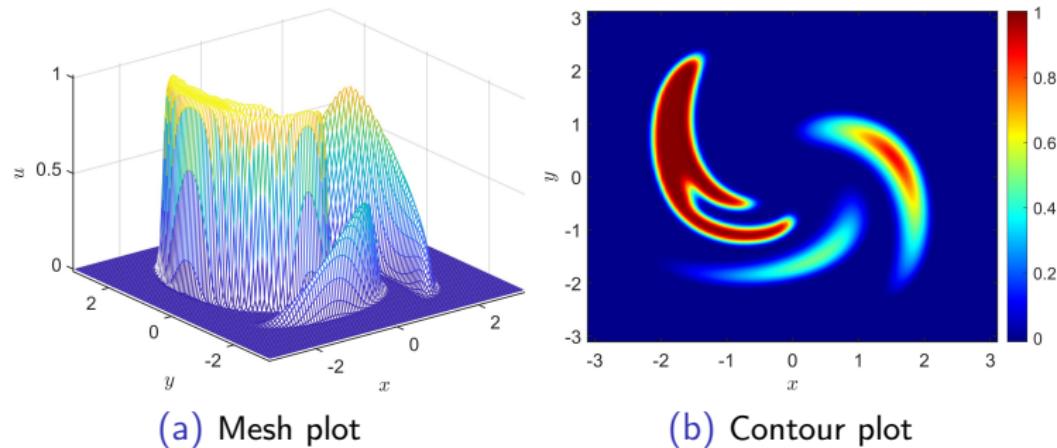


Figure 7: (Swirling deformation flow) Mesh plot and contour plot of the numerical solution of the EL-RK-FV-WENO scheme with $\text{CFL} = 10.2$ and mesh size 100×100 at $t = 0.75$.

Linear models

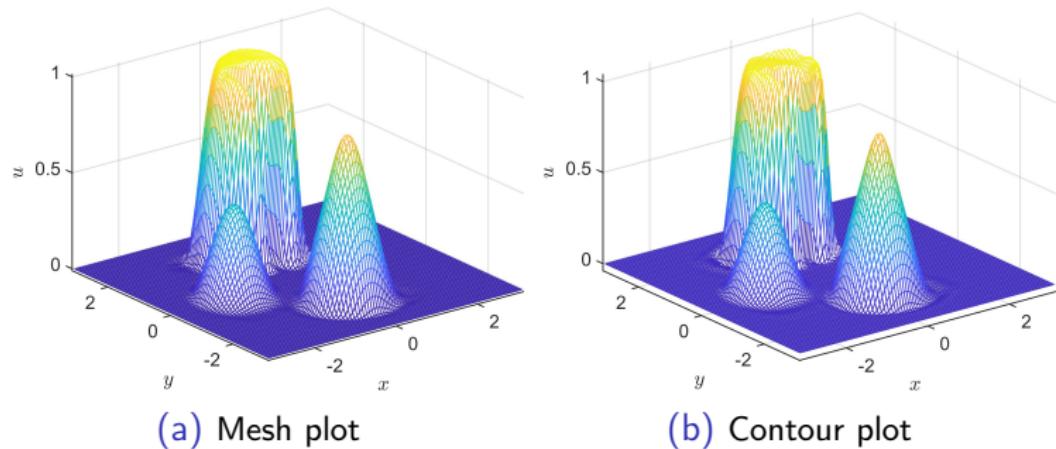


Figure 8: (Swirling deformation flow) Mesh plots of the numerical solution of the EL-RK-FV-WENO scheme (left) and WENO scheme in⁸ using the same setting (right) with CFL = 10.2 and mesh size 100×100 at $t = 1.5$.

⁸Nanyi Zheng et al. "A fourth-order conservative semi-Lagrangian finite volume WENO scheme without operator splitting for kinetic and fluid simulations". In: *Computer Methods in Applied Mechanics and Engineering* 395 (2022), p. 114973. ISSN: 0045-7825.

Linear models

Example 2

(The 0D2V Leonard-Bernstein linearized Fokker-Planck equation)

Consider the following equation:

$$f_t + \frac{1}{\epsilon} ((v_x - \bar{v}_x) f)_{v_x} - \frac{1}{\epsilon} ((v_y - \bar{v}_y) f)_{v_y} = \frac{1}{\epsilon} D (f_{v_x v_x} + f_{v_y v_y}), \quad (30)$$
$$v_x, v_y \in [-2\pi, 2\pi].$$

We consider zero boundary conditions and an equilibrium solution by the given Maxwellian:

$$f_M(v_x, v_y) = \frac{n}{2\pi R T} \exp\left(\frac{(v_x - \bar{v}_x)^2 + (v_y - \bar{v}_y)^2}{2RT}\right), \quad (31)$$

where, $\epsilon = 1$, gas constant $R = 1/6$, temperature $T = 3$, thermal velocity $v_{th} = \sqrt{2RT} = \sqrt{2D} = 1$, number density $n = \pi$, and bulk velocities $\bar{v}_x = \bar{v}_y = 0$.

Linear models

For spatial and temporal accuracy tests, we choose
 $f(v_x, v_y, 0) = f_M(v_x, v_y)$.

Table 2: (The 0D2V Leonard-Bernstein linearized Fokker-Planck equation) L^1 , L^2 , and L^∞ errors and corresponding orders of accuracy of the EL-RK-FV-WENO scheme for (30) with initial condition (31) at $t = 0.5$ with CFL = 1.

mesh	L^1 error	order	L^2 error	order	L^∞ error	order
20×20	2.18E-03	—	1.38E-02	—	2.95E-01	—
40×40	3.01E-04	2.85	2.05E-03	2.75	9.50E-02	1.63
80×80	3.33E-05	3.18	1.58E-04	3.70	6.40E-03	3.89
160×160	4.10E-06	3.02	1.80E-05	3.13	2.88E-04	4.47
320×320	4.77E-07	3.10	2.10E-06	3.11	3.20E-05	3.17

Linear models

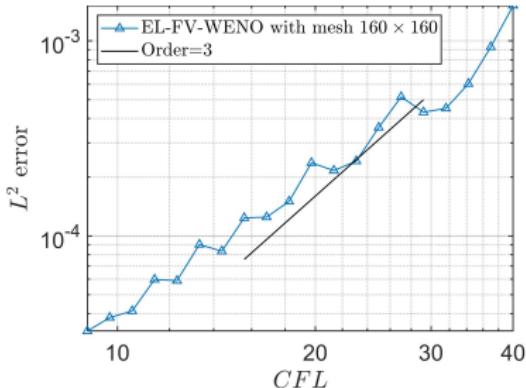


Figure 9: (The 0D2V Leonard-Bernstein linearized Fokker-Planck equation) Log-log plot of CFL numbers versus L^2 errors with fixed mesh 160×160 at $t = 0.5$ for the EL-RK-FV-WENO scheme.

Linear models

To test the relaxation of the system, we choose the initial condition $f(v_x, v_y, t = 0) = f_{M1}(v_x, v_y) + f_{M2}(v_x, v_y)$, where the parameters of each Maxwellian, f_{M1} and f_{M2} , are detailed in Table 3.

	f_{M1}	f_{M2}
n	1.990964530353041	1.150628123236752
\bar{v}_x	0.4979792385268875	-0.8616676237412346
\bar{v}_y	0	0
T	2.46518981703837	0.4107062104302872

Table 3: (The 0D2V Leonard-Bernstein linearized Fokker-Planck equation) Settings of the Maxwellians f_{M1} and f_{M2} .

Linear models

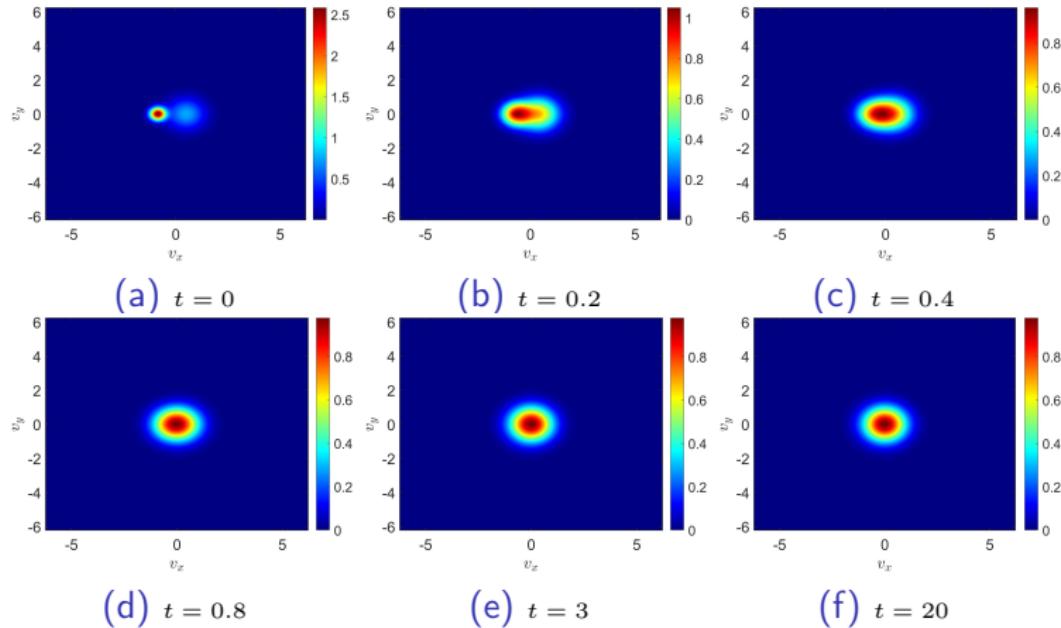


Figure 10: (The 0D2V Leonard-Bernstein linearized Fokker-Planck equation) Contour plots of the numerical results of the EL-RK-FV-WENO scheme with $CFL = 3$ and mesh size 100×100 at $t = 0$ (initial condition), $t = 0.2$, $t = 0.4$, $t = 0.6$, $t = 3$, and $t = 20$.

Linear models

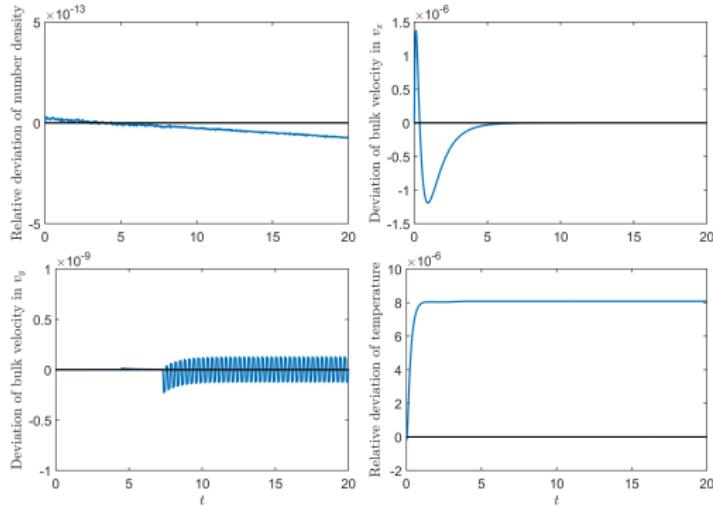


Figure 11: (The 0D2V Leonard-Bernstein linearized Fokker-Planck equation) Relative deviation (or deviation) of number density, bulk velocity in v_x , bulk velocity in v_y , and temperature for the EL-RK-FV-WENO scheme with CFL = 10.2 and with mesh 160×160 from $t = 0$ to $t = 20$.

Nonlinear model

Example 3

(Kelvin-Helmholtz instability problem) Consider the guiding center Vlasov model:

$$\begin{aligned}\rho_t + \nabla \cdot (\mathbf{E}^\perp \rho) &= 0, \\ -\Delta \Phi &= \rho, \quad \mathbf{E}^\perp = (-\Phi_y, \Phi_x),\end{aligned}\tag{32}$$

with the periodic boundary condition and the following initial condition:

$$\rho(x, y, 0) = \sin(y) + 0.015\cos(0.5x), \quad x \in [0, 4\pi], \quad y \in [0, 2\pi],\tag{33}$$

where ρ is the charge density and \mathbf{E} is the electric field.

Nonlinear model

Table 4: (Kelvin-Helmholtz instability problem) L^1 , L^2 , and L^∞ errors and corresponding orders of accuracy of the EL-RK-FV-WENO scheme for (32) with initial condition (33) at $t = 5$ with CFL = 1.

mesh	L^1 error	order	L^2 error	order	L^∞ error	order
16×16	6.79E-03	—	1.11E-02	—	6.68E-02	—
32×32	5.48E-04	3.63	9.46E-04	3.55	1.18E-02	2.50
64×64	4.35E-05	3.66	7.15E-05	3.73	1.43E-03	3.05
128×128	4.37E-06	3.31	6.45E-06	3.47	1.98E-04	2.85
256×256	5.23E-07	3.06	6.72E-07	3.26	1.73E-05	3.51

Nonlinear model

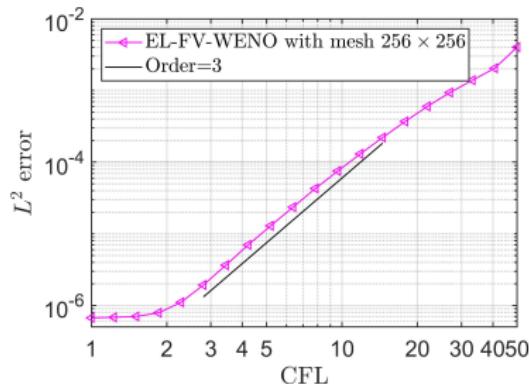


Figure 12: (Kelvin-Helmholtz instability problem) Log-log plot of CFL numbers versus L^2 errors with fixed mesh 256×256 at $t = 5$ of the EL-RK-FV-WENO scheme.

Nonlinear model

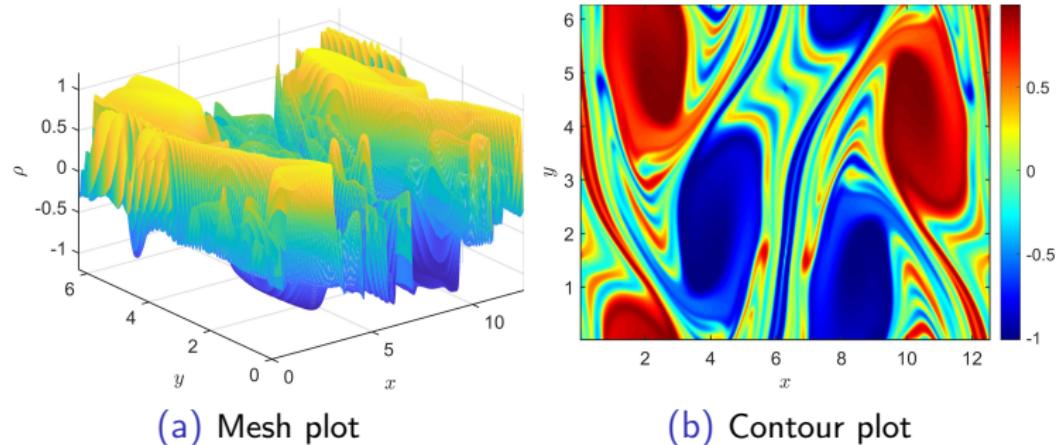
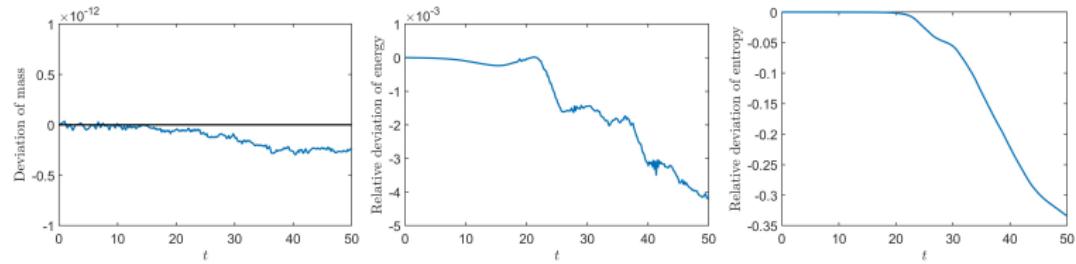


Figure 13: (Kelvin-Helmholtz instability problem) Mesh plot and contour plot of the numerical solution of the EL-RK-FV-WENO scheme with CFL = 10.2 and with mesh 256×256 at $t = 40$.

Nonlinear model



(a) Deviation of mass

(b) Relative deviation of energy

(c) Relative deviation of entropy

Figure 14: (Kelvin-Helmholtz instability problem) Deviation of mass, relative deviation of energy and entropy for the EL-RK-FV-WENO scheme with CFL = 10.2 and with mesh 256×256 from $t = 0$ to $t = 50$.

Nonlinear model

Example 4

(Incompressible Navier-Stokes equations) The governing equations are as follows:

$$\begin{aligned}\omega_t + (u\omega)_x + (v\omega)_y &= \nu(\omega_{xx} + \omega_{yy}), \\ \Delta\psi = \omega, \quad (u, v) &= (\psi_y, \psi_x),\end{aligned}\tag{34}$$

where ω is the vorticity of the flow, (u, v) is the velocity field, and ν is the kinematic viscosity, which is set to be $\frac{1}{100}$.

We first consider an initial condition given by:

$$\omega(x, y, 0) = -2 \sin(x) \sin(y), \quad x \in [0, 2\pi], \quad y \in [0, 2\pi]\tag{35}$$

with the exact solution $\omega(x, y, t) = -2 \sin(x) \sin(y) \exp(-2t\nu)$.

Nonlinear model

Table 5: (Incompressible Navier-Stokes equations) L^1 , L^2 , and L^∞ errors and corresponding orders of accuracy of the EL-RK-FV-WENO scheme for (34) with initial condition (35) at $t = 0.5$ with CFL = 1.

mesh	L^1 error	order	L^2 error	order	L^∞ error	order
16×16	3.90E-03	—	4.65E-03	—	1.14E-02	—
32×32	4.88E-04	3.00	5.82E-04	3.00	1.45E-03	2.98
64×64	6.09E-05	3.00	7.27E-05	3.00	1.82E-04	2.99
128×128	7.61E-06	3.00	9.08E-06	3.00	2.29E-05	2.99
256×256	9.51E-07	3.00	1.13E-06	3.00	2.86E-06	3.00

Nonlinear model

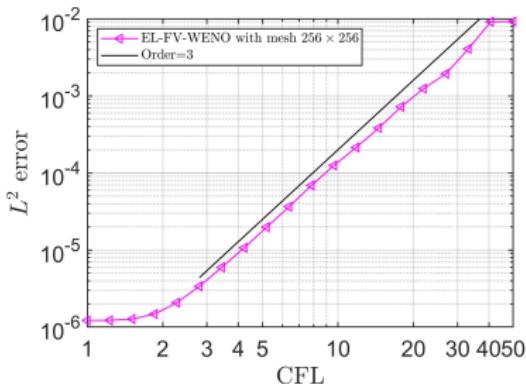


Figure 15: (Incompressible Navier-Stokes equations) Log-log plot of CFL numbers versus L^2 errors with fixed mesh 256×256 at $t = 0.5$ of the EL-RK-FV-WENO scheme.

Nonlinear model

For a more complex scenario, we consider the incompressible Navier-Stokes equation (34) on $[0, 2\pi]^2$ with the following initial condition (the vortex patch problem)

$$\omega(x, y, 0) = \begin{cases} -1, & \frac{\pi}{2} \leq x \leq \frac{3\pi}{2}, \quad \frac{\pi}{4} \leq y \leq \frac{3\pi}{4}; \\ 1, & \frac{\pi}{2} \leq x \leq \frac{3\pi}{2}, \quad \frac{5\pi}{4} \leq y \leq \frac{7\pi}{4}; \\ 0 & \text{otherwise} \end{cases} \quad (36)$$

with zero boundary condition.

Nonlinear model

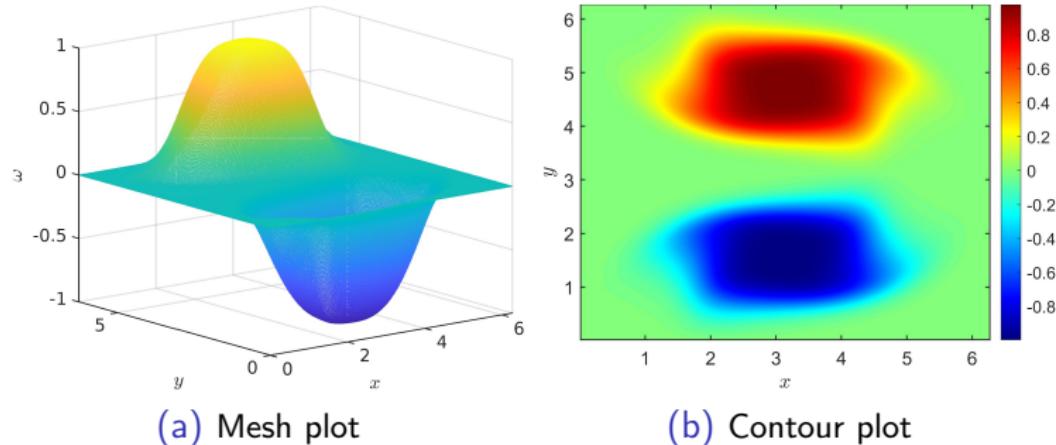


Figure 16: (Vortex patch problem) Mesh plot and contour plot of the numerical solution of the EL-RK-FV-WENO scheme with CFL = 10.2 and with mesh 256×256 at $t = 5$.

Outline

Background

Design of the EL-RK-FV-WENO scheme

Numerical tests

Concluding Remarks

Concluding Remarks

- ▶ We proposed a **novel scheme** for high dimensional nonlinear convection-diffusion simulations that can ensure
 1. mass conservation
 2. high order accuracy in both space and time
 3. high resolution for sharp solution/discontinuous solution by introducing **an improved WENO scheme**
 4. allowing explicitly large time stepping size of

$$\Delta t = O(\sqrt{h})$$

5. robustness

- ▶ We presented a **novel remapping method** that transfers the piecewise WENO reconstruction polynomial on the Eulerian mesh to a new piecewise polynomial on the Lagrangian mesh.

Thanks for your attention!