

An Eulerian-Lagrangian Runge-Kutta finite volume (EL-RK-FV) method for solving convection and convection-diffusion equations

Joseph Nakao¹, Jiajie Chen¹, and Jingmei Qiu¹

¹Department of Mathematical Sciences, University of Delaware, Newark, DE 19716

Abstract

We propose a new Eulerian-Lagrangian Runge-Kutta finite volume method for numerically solving convection and convection-diffusion equations. Eulerian-Lagrangian and semi-Lagrangian methods have grown in popularity mostly due to their ability to allow large time steps. Our proposed scheme is formulated by integrating the PDE on a space-time region partitioned by approximations of the characteristics determined from the Rankine-Hugoniot jump condition; and then rewriting the time-integral form into a time differential form to allow application of Runge-Kutta (RK) methods via the method-of-lines approach. The scheme can be viewed as a generalization of the standard Runge-Kutta finite volume (RK-FV) scheme for which the space-time region is partitioned by approximate characteristics with zero velocity. The high-order spatial reconstruction is achieved using the recently developed weighted essentially non-oscillatory schemes with adaptive order (WENO-AO); and the high-order temporal accuracy is achieved by explicit RK methods for convection equations and implicit-explicit (IMEX) RK methods for convection-diffusion equations. Our algorithm extends to higher dimensions via dimensional splitting. Numerical experiments demonstrate our algorithm's robustness, high-order accuracy, and ability to handle extra large time steps.

Keywords: Eulerian-Lagrangian, semi-Lagrangian, convection-diffusion equation, WENO-AO, IMEX Runge-Kutta

1 Introduction

In this paper, we are concerned with numerically solving convection-diffusion equations of the form

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

where $\epsilon \geq 0$. We propose an Eulerian-Lagrangian Runge-Kutta finite volume (EL-RK-FV) scheme utilizing weighted essentially non-oscillatory (WENO) schemes with adaptive order (WENO-AO) for spatial reconstruction. The proposed method is designed for one-dimensional problems of the form (1.1) and extended to higher-dimensional problems via dimensional splitting.

Eulerian-Lagrangian (EL) and semi-Lagrangian (SL) schemes [14, 41, 48] have proven to be computationally effective when solving hyperbolic problems because of their ability to employ high spatial resolution schemes while admitting very large CFL with numerical stability. Generally speaking, an EL or SL method involves working on a background grid so that high-order spatial resolutions can be used (the Eulerian part), and tracing characteristics of the cell boundaries backward/forward in time to relax the CFL constraint (the Lagrangian part). Such methods have been developed in a wide variety of frameworks: discontinuous Galerkin [12, 20, 39, 40], finite difference [13, 15, 29, 32, 37, 38, 47], finite volume [1, 7, 19, 22, 27, 28]. Although the SL and EL frameworks are similar in spirit, the SL framework assumes exact characteristic tracing and hence poses difficulties when considering nonlinear problems. Two other methods similar to EL and SL methods are the arbitrary Lagrangian-Eulerian (ALE) methods where an arbitrary mesh velocity not necessarily aligned with the fluid velocity is defined [8, 9, 10, 11, 21, 26, 36], and moving mesh methods where the PDE is first evolved in time and then followed by some mesh-redistribution procedure [33, 34, 35, 41, 44, 45]. The main difference between these two methods and the previously mentioned methods is that they move the mesh adaptively to focus resolving the solution around sharp transitions; whereas EL and SL methods evolve the equation by following characteristics.

Our goal in this paper is to develop a new high-order EL method in the finite volume framework using method-of-lines (MOL) RK time discretizations. Finite volume methods are attractive since they are naturally mass conservative, easy to physically interpret, and modifiable for nonuniform grids. Similar to the recent developments made by Huang, Arbogast, and Qiu [27, 28], we use approximate characteristics to define a traceback space-time region, and then use WENO reconstructions to evaluate the modified flux. Huang and Arbogast developed a re-averaging technique that allows high-order reconstruction of the solution at arbitrary points by applying a standard WENO scheme [17, 43] over a uniform reconstruction grid that is defined separately. They then used a natural continuous extension [51] of Runge-Kutta schemes, which requires the solution at several Gaussian nodes of the interval $[t^n, t^{n+1}]$, to evolve the solution along the approximate characteristics.

The novelty of our proposed method is twofold: (1) the partition of space-time regions formed by linear approximations of the characteristic curves, and (2) integrating the differential equation over the partitioned space-time regions, followed by rewriting the space-time integral form of the equation into a spatial-integral time-differential form. In this way, a MOL RK type method can be directly applied for time discretization, thus avoiding the need to use a natural continuous extension of RK schemes. To be more precise, we construct linear approximate characteristics by using the Rankine-Hugoniot jump condition to define the traceback space-time regions. If the linear approximate characteristics are defined with zero velocity, then the proposed EL-RK-FV scheme reduces to the standard RK-FV method. Whereas, when linear space-time curves adequately approximate the exact characteristics, a large time stepping size is still permitted. We use WENO-AO to perform a solution remapping of the uniform cell averages onto the possibly nonuniform traceback cells. The recently

developed WENO-AO schemes [4, 5, 6] are robust and guarantee the existence of the linear weights at arbitrary points. We note that Chen, et al. used WENO-AO schemes in the SL framework [15], and Huang and Arbogast have recently used WENO-AO schemes in the Eulerian framework [3, 4]. RK methods are used to evolve the MOL system along the approximate characteristics. Explicit RK methods, such as the strong stability-preserving (SSP) RK methods [24], are used for convection equations; and implicit-explicit (IMEX) RK methods [2, 18, 25] are used for convection-diffusion equations. In the latter case, the non-stiff convective term is treated explicitly and the stiff diffusive term is treated implicitly. Dimensional splitting is used to extend the one-dimensional algorithm to solve multi-dimensional problems. The proposed method is high-order accurate, capable of resolving discontinuities without oscillations, mass conservative, and stable with large time stepping sizes.

The paper is organized as follows. We discuss the EL-RK-FV algorithm for pure convection problems in Section 2. In Section 3, we discuss the EL-RK-FV algorithm, coupled with IMEX RK schemes, for convection-diffusion equations. Numerical performance of the EL-RK-FV algorithm is shown in Section 4 by applying the algorithm to several linear and nonlinear test problems. Conclusions are made in Section 5. Appendices are listed after the references.

2 The EL-RK-FV method for pure convection problems

The spirit of the EL-RK-FV method is best demonstrated by starting with a pure convection problem in one dimension, i.e., equation (1.1) with $\epsilon = 0$ and $g(x) = 0$. Let the flux be denoted $f(x)$. We first discuss the formulation of the scheme in Section 2.1, followed by discussion of high-order spatial reconstruction in Section 2.2 and time discretization in Section 2.3.

2.1 Scheme formulation

We discretize the spatial domain $[a, b]$ into N_x intervals with $N_x + 1$ uniformly distributed nodes

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \dots < x_{N_x - \frac{1}{2}} < x_{N_x + \frac{1}{2}} = b.$$

Define the cells $I_j := [x_{j - \frac{1}{2}}, x_{j + \frac{1}{2}}]$ with centers $x_j = (x_{j - \frac{1}{2}} + x_{j + \frac{1}{2}})/2$ and widths $\Delta x_j = \Delta x$ for $j = 1, \dots, N_x$. We let

$$\Delta t = \frac{CFL \Delta x}{\max|f'(u)|}, \quad (2.1)$$

where CFL defines the time stepping size. In contrast to Eulerian methods, which evolve the solution on a stationary mesh, our EL algorithm proposes tracing the characteristics *backwards* in time from t^{n+1} to t^n to partition a set of space-time regions based on the computational grid. Since tracing characteristics in the nonlinear case is often nontrivial, we consider computing *approximations of characteristics* that are linear space-time curves. In particular, the *approximate characteristic speeds* at nodes $x_{j + \frac{1}{2}}$ and time t^{n+1} are defined

using the Rankine-Hugoniot jump condition at time t^n ,

$$\nu_{j+\frac{1}{2}} = \begin{cases} \frac{f(\bar{u}_{j+1}^n) - f(\bar{u}_j^n)}{\bar{u}_{j+1}^n - \bar{u}_j^n}, & \bar{u}_{j+1}^n \neq \bar{u}_j^n, \\ f'(\bar{u}_j^n), & \bar{u}_{j+1}^n = \bar{u}_j^n, \end{cases} \quad (2.2)$$

where \bar{u}_j^n and \bar{u}_{j+1}^n are the cell averages at time t^n . In practice we tested if $|\bar{u}_{j+1}^n - \bar{u}_j^n| < \epsilon$, with $\epsilon = 1.0e-8$, in which case we took \bar{u} to be the average of \bar{u}^- and \bar{u}^+ . As seen in Figure 1, the (upstream) traceback nodes can be defined by

$$x_{j+\frac{1}{2}}^* := x_{j+\frac{1}{2}} - \nu_{j+\frac{1}{2}} \Delta t, \quad j = 1, \dots, N. \quad (2.3)$$

Define $\tilde{x}_{j+\frac{1}{2}}(t) = x_{j+\frac{1}{2}}^* + \nu_{j+\frac{1}{2}}(t-t^n)$ with $t^n \leq t \leq t^{n+1}$, for $j = 0, 1, 2, \dots, N$. The approximate characteristics are given by the space-time tracelines

$$\mathcal{S}_{left} := \{(\tilde{x}_{j-\frac{1}{2}}(t), t) : t^n \leq t \leq t^{n+1}\} \quad \text{and} \quad \mathcal{S}_{right} := \{(\tilde{x}_{j+\frac{1}{2}}(t), t) : t^n \leq t \leq t^{n+1}\}.$$

Note that $x_{j+\frac{1}{2}}^* = \tilde{x}_{j+\frac{1}{2}}(t^n)$, $x_{j+\frac{1}{2}} = \tilde{x}_{j+\frac{1}{2}}(t^{n+1})$, and the (upstream) traceback cells $I_j^* = \tilde{I}_j(t^n)$ are in general nonuniform. We define the space-time domain Ω_j as the region bounded by I_j , I_j^* , \mathcal{S}_{left} , and \mathcal{S}_{right} .

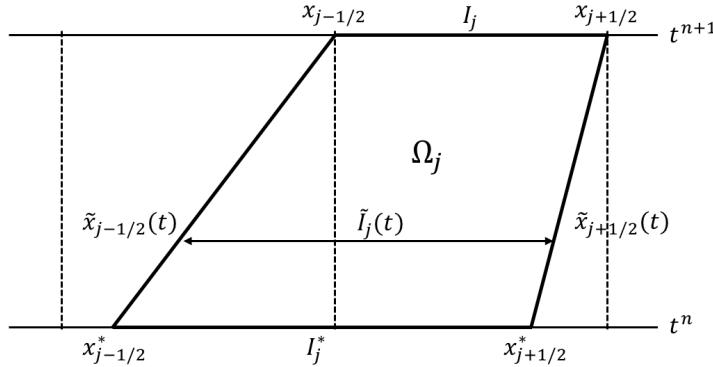


Figure 1: The space-time region Ω_j .

With the constructed space-time region Ω_j , we rewrite the one-dimensional pure convection problem in divergence form $\nabla_{t,x} \cdot (u, f(u))^T = 0$, integrate it over Ω_j , and apply the divergence theorem to get

$$\begin{aligned} & \int_{I_j} u(x, t^{n+1}) dx - \int_{I_j^*} u(x, t^n) dx \\ &= - \left[\int_{t^n}^{t^{n+1}} \left(f(u(\tilde{x}_{j+\frac{1}{2}}(t), t)) - \nu_{j+\frac{1}{2}} u(\tilde{x}_{j+\frac{1}{2}}(t), t) \right) dt - \int_{t^n}^{t^{n+1}} \left(f(u(\tilde{x}_{j-\frac{1}{2}}(t), t)) - \nu_{j-\frac{1}{2}} u(\tilde{x}_{j-\frac{1}{2}}(t), t) \right) dt \right]. \end{aligned} \quad (2.4)$$

We rewrite the time-integral form (2.4) into the time-differential form to get

$$\frac{d}{dt} \int_{\tilde{I}_j(t)} u(x, t) dx = - \left[F_{j+\frac{1}{2}}(t) - F_{j-\frac{1}{2}}(t) \right], \quad (2.5)$$

where $F_{j+\frac{1}{2}}(t) := f(u(\tilde{x}_{j+\frac{1}{2}}(t), t)) - \nu_{j+\frac{1}{2}} u(\tilde{x}_{j+\frac{1}{2}}(t), t)$ is called the *modified flux function*. Choosing any appropriate monotone numerical flux function

$$\hat{F}_{j+\frac{1}{2}}(t) = \hat{F}_{j+\frac{1}{2}}(u_{j+\frac{1}{2}}^-, u_{j+\frac{1}{2}}^+; t) \quad (2.6)$$

(e.g., Lax-Friedrichs flux) for the modified flux function, equation (2.5) can be rewritten as the following semi-discrete finite volume scheme:

$$\frac{d}{dt} \int_{\tilde{I}_j(t)} u(x, t) dx = - \left[\hat{F}_{j+\frac{1}{2}}(t) - \hat{F}_{j-\frac{1}{2}}(t) \right]. \quad (2.7)$$

Here, the starting condition is obtained by a solution remapping onto a trackback grid, discussed in Section 2.2; $\hat{F}_{j+\frac{1}{2}}(t)$ are computed from (2.6) with reconstructed values from neighboring cell averages at time t , discussed in Section 2.3. We evolve equation (2.7) by the MOL approach with explicit RK schemes, discussed in Section 2.4.

2.2 Solution remapping onto a traceback grid

Referring back to Figure 1 and equation (2.4), we see that the solution will need to be projected onto the traceback grid in order to compute the *traceback cell averages*

$$\tilde{u}_j^n := \frac{1}{\Delta x_j^*} \int_{I_j^*} u(x, t^n) dx, \quad j = 1, 2, \dots, N, \quad (2.8)$$

that is, the starting condition for (2.7). Hence, we desire a procedure for an *integral reconstruction* that—in general—uses known uniform cell averages $\{\bar{u}_j(t) : j = 1, 2, \dots, N\}$ to approximate the desired cell averages $\{\tilde{u}_j(t) : j = 1, 2, \dots, N\}$ defined by

$$\tilde{u}_j(t) := \frac{1}{\Delta \tilde{x}_j(t)} \int_{\tilde{I}_j(t)} u(x, t) dx, \quad j = 1, 2, \dots, N. \quad (2.9)$$

Unless otherwise stated, overlines (e.g., $\bar{u}_j(t)$) denote *uniform* cell averages, and tildes (e.g., $\tilde{u}_j(t)$) denote *nonuniform* cell averages.

Since discontinuities and sharp gradients can occur when solving pure convection problems, we use high-resolution schemes to control spurious oscillations, e.g., weighted essential non-oscillatory (WENO) methods [5, 17, 31, 43]. Again, we assume to only be given the uniform cell averages at some time t . One might apply the well known WENO procedure presented in [17, 43] to obtain the *reconstruction polynomials* $\mathcal{R}_j(x \in I_j)$ in terms of the neighboring uniform cell averages (at time t) \bar{u}_i , $i = j-p, \dots, j+q$. Referring to Figure 1 for the sake of demonstration, we might then approximate the cell averages \tilde{u}_j^* by

$$\tilde{u}_j^* \approx \frac{1}{\Delta x_j^*} \left(\int_{I_j^* \cap I_{j-1}} \mathcal{R}_{j-1}(x) dx + \int_{I_j^* \cap I_j} \mathcal{R}_j(x) dx \right). \quad (2.10)$$

However, the linear weights in the WENO reconstruction are not guaranteed to exist or be positive at arbitrary points. To alleviate this issue, we instead use the WENO schemes with adaptive order (WENO-AO) presented in [5] since the linear weights exist at arbitrary points.

The overarching idea of WENO-AO methods is to provide high order accuracy for smooth solutions over a large center stencil and adaptively reduce to lower order accuracy when the solution does not permit the high order accuracy. This is done by creating a nonlinear hybridization between a large center stencil with high order accuracy, and very stable lower order WENO schemes (e.g., CWENO schemes [31]). Aside from the high order accuracy and existence of linear weights at arbitrary points, the robustness of these WENO-AO schemes is particularly attractive for our purposes. The authors in [5] write WENO-AO(p, r) to denote an adaptive order that is at best p th order (from the large center stencil) and at worst r th order (from the stable lower order stencils). For our purposes we use WENO-AO(5,3). The end product of WENO and WENO-AO methods is ultimately a reconstruction polynomial $\mathcal{R}_j(x \in I_j)$ that we shall use for reconstruction.

Equation (2.10) is valid when using WENO-AO reconstruction polynomials $\mathcal{R}_j(x \in I_j)$. In general, if the traceback points $\tilde{x}_{j-\frac{1}{2}}(t)$ and $\tilde{x}_{j+\frac{1}{2}}(t)$ reside in cells I_ℓ and I_r respectively, then

$$\int_{\tilde{I}_j(t)} u(x, t) dx \approx \int_{\tilde{I}_j(t) \cap I_\ell} \mathcal{R}_\ell(x) dx + \Delta x \sum_{i=\ell+1}^{r-1} \bar{u}_i(t) + \int_{\tilde{I}_j(t) \cap I_r} \mathcal{R}_r(x) dx. \quad (2.11)$$

Below, we summarize the integral reconstruction procedure using WENO-AO as the Algorithm 1.

Algorithm 1. Integral reconstruction using WENO-AO

Input: uniform cell averages $\bar{u}_j(t)$ on the background grid of nodes $x_{j+\frac{1}{2}}$.

Output: possibly nonuniform cell averages $\tilde{u}_j(t)$ on the traceback grid of nodes $\tilde{x}_{j+\frac{1}{2}}(t)$.

```

do  $j = 1, 2, \dots, N$ 
  Locate the uniform background cells that  $\tilde{x}_{j-\frac{1}{2}}(t)$  and  $\tilde{x}_{j+\frac{1}{2}}(t)$  reside in.
  Call these cells  $I_\ell$  and  $I_r$ , respectively.
  if  $\ell = r$ 
    Compute  $\tilde{u}_j(t) \approx \frac{1}{\Delta \tilde{x}_j(t)} \int_{\tilde{I}_j(t)} \mathcal{R}_\ell(x) dx$ .
  else
    do  $k = \ell, r$ 
      Compute  $\int_{\tilde{I}_j(t) \cap I_k} \mathcal{R}_k(x) dx$ .
    end do
    Compute  $\tilde{u}_j(t) \approx \frac{1}{\Delta \tilde{x}_j(t)} \left( \int_{\tilde{I}_j(t) \cap I_\ell} \mathcal{R}_\ell(x) dx + \Delta x \sum_{i=\ell+1}^{r-1} \bar{u}_i(t) + \int_{\tilde{I}_j(t) \cap I_r} \mathcal{R}_r(x) dx \right)$ .
  
```

```

    end if
end do

```

2.3 Reconstruction of point values

Referring back to Figure 1 and equations (2.4)-(2.7), we also need to reconstruct the point values $u(\tilde{x}_{j+\frac{1}{2}}^-(t), t)$ and $u(\tilde{x}_{j+\frac{1}{2}}^+(t), t)$ for the modified flux function. However, the WENO-AO schemes in [5] assume a uniform grid for convenience and efficiency. We wish to avoid using nonuniform WENO methods since the linear weights need to be recomputed every step and will quickly become expensive. However, nonuniform WENO methods for two-dimensional problems [6, 52, 53, 54, 55] might become a more reasonable and realistic choice when dealing with non-splitting algorithms. Yet, we still need to reconstruct the left and right limits on the generally nonuniform traceback grid. We propose using the continuous piecewise-linear transformation from the nonuniform grid consisting of nodes $\tilde{x}_{j+\frac{1}{2}}(t)$ to the uniform background grid; performing a WENO-AO reconstruction on the uniform grid; and mapping back to the nonuniform grid to obtain the desired limits. We call this the *nonuniform-to-uniform transformation*. Given a fixed $t \in [t^n, t^{n+1}]$, consider the linear bijection $\phi_j : \tilde{I}_j(t) \rightarrow I_j$ for $j = 1, 2, \dots, N$. Letting $x \in \tilde{I}_j(t)$ and $\xi \in I_j$,

$$\int_{\tilde{I}_j(t)} u(x, t) dx = |J| \int_{I_j} u(x(\xi), t) d\xi, \quad (2.12)$$

where $|J|(\xi) = |dx/d\xi|(\xi)$ is the Jacobian of the bijection ϕ_j . In particular, the Jacobian is constant for the linear interpolant,

$$|J| = \left| \frac{d}{d\xi} \left(\frac{\Delta \tilde{x}_j(t)}{\Delta x} \xi - \frac{x_{j+\frac{1}{2}} \tilde{x}_{j-\frac{1}{2}}(t) + x_{j-\frac{1}{2}} \tilde{x}_{j+\frac{1}{2}}(t)}{\Delta x} \right) \right| = \frac{\Delta \tilde{x}_j(t)}{\Delta x}. \quad (2.13)$$

Since we want to apply WENO-AO over the uniform grid, we define the *auxiliary uniform cell averages* as

$$\tilde{u}_j(t) := \frac{1}{\Delta x} \int_{I_j} u(x(\xi), t) d\xi = \frac{1}{\Delta x |J|} \int_{\tilde{I}_j(t)} u(x, t) dx = \frac{1}{\Delta \tilde{x}_j(t)} \int_{\tilde{I}_j(t)} u(x, t) dx = \tilde{u}_j(t). \quad (2.14)$$

Note that we have shown the auxiliary uniform cell averages (on the uniform background grid) are identical to the nonuniform cell averages. Hence, under this continuous piecewise-linear mapping, we can directly use the nonuniform cell averages at time t in the (uniform) WENO-AO procedure to obtain the left and right limits $u(\tilde{x}_{j+\frac{1}{2}}^-(t), t)$ and $u(\tilde{x}_{j+\frac{1}{2}}^+(t), t)$. We further note that the high-order accuracy is preserved with such a strategy only when there is a smooth mapping with equation (2.12). Theoretical justification for the existence of such a mapping is highly nontrivial. Yet, our numerical tests verify that high-order spatial accuracy is still achieved under this continuous, piecewise linear, C^0 mapping so long as the (approximate) characteristic field that controls the traceback grid is smooth. We found that the accuracy drops to first or second order when the (approximate) characteristic field that

controls the traceback grid is not smooth (e.g., a fixed traceback grid with alternating cell lengths $\frac{4}{3}\Delta x : \frac{2}{3}\Delta x$ or $\frac{9}{5}\Delta x : \frac{1}{5}\Delta x$). However, since the traceback grid is determined by the smooth velocity field via the Rankine-Hugoniot jump condition, the high-order accuracy is still observed.

Algorithm 2. Reconstruction using WENO-AO with the nonuniform-to-uniform transformation

Input: nonuniform cell averages $\tilde{u}_j(t)$ on the traceback grid.

Output: left and right limits $u(\tilde{x}_{j+\frac{1}{2}}^-(t), t)$ and $u(\tilde{x}_{j+\frac{1}{2}}^+(t), t)$.

do $j = 1, 2, \dots, N$

 Calculate the (uniform) WENO-AO reconstruction polynomial $\mathcal{R}_j(x \in I_j)$ in terms of

 the neighboring auxiliary uniform cell averages $\tilde{u}_i(t) = \tilde{u}_i(t), i = j-p, \dots, j+q$.

 Compute the left limit $u(\tilde{x}_{j+\frac{1}{2}}^-(t), t) \approx \mathcal{R}_j(x_{j+\frac{1}{2}}^-)$.

 Compute the right limit $u(\tilde{x}_{j-\frac{1}{2}}^+(t), t) \approx \mathcal{R}_j(x_{j-\frac{1}{2}}^+)$.

end do

2.4 Time evolution with explicit Runge-Kutta methods

Algorithms 1 and 2 now allow us to perform (integral) reconstruction on a traceback grid consisting of nodes $\tilde{x}_{j+\frac{1}{2}}(t)$. With these two tools we can evolve equation (2.7) using any explicit RK method. Recall that our primary goal is to achieve high order accuracy while also taking large time steps. In our numerical tests we use WENO-AO(5,3) for the spatial approximation; although higher order WENO-AO methods can certainly be used. As such, we would like to use high-order time stepping methods. In the cases where the solution is smooth, the standard fourth-order RK method suffices. However, if the solution is discontinuous (e.g., a travelling Heaviside step function), then we require an explicit SSP RK method [24]. Explicit SSP RK methods are especially attractive when numerically solving hyperbolic conservation laws because they maintain strong stability while also achieving high order accuracy in time. When applicable, we use the optimal three-stage, third-order explicit SSP RK method outlined below for demonstrative purposes.

$$\begin{aligned}
\Delta \tilde{x}_j^{(0)} \tilde{u}_j^{(0)} &= \Delta x_j^* \bar{u}_j^*, \\
\Delta \tilde{x}_j^{(1)} \tilde{u}_j^{(1)} &= \Delta \tilde{x}_j^{(0)} \tilde{u}_j^{(0)} - \Delta t \left(\hat{F}_{j+\frac{1}{2}}^{(0)}(u^-, u^+; t^n) - \hat{F}_{j-\frac{1}{2}}^{(0)}(u^-, u^+; t^n) \right), \\
\Delta \tilde{x}_j^{(2)} \tilde{u}_j^{(2)} &= \Delta \tilde{x}_j^{(0)} \tilde{u}_j^{(0)} - \frac{\Delta t}{4} \left[\left(\hat{F}_{j+\frac{1}{2}}^{(0)}(u^-, u^+; t^n) - \hat{F}_{j-\frac{1}{2}}^{(0)}(u^-, u^+; t^n) \right) \right. \\
&\quad \left. + \left(\hat{F}_{j+\frac{1}{2}}^{(1)}(u^-, u^+; t^{n+1}) - \hat{F}_{j-\frac{1}{2}}^{(1)}(u^-, u^+; t^{n+1}) \right) \right], \\
\Delta x_j \bar{u}_j^{n+1} &= \Delta \tilde{x}_j^{(0)} \tilde{u}_j^{(0)} - \frac{2\Delta t}{3} \left(\hat{F}_{j+\frac{1}{2}}^{(2)}(u^-, u^+; t^{n+\frac{1}{2}}) - \hat{F}_{j-\frac{1}{2}}^{(2)}(u^-, u^+; t^{n+\frac{1}{2}}) \right) \\
&\quad - \frac{\Delta t}{6} \left[\left(\hat{F}_{j+\frac{1}{2}}^{(0)}(u^-, u^+; t^n) - \hat{F}_{j-\frac{1}{2}}^{(0)}(u^-, u^+; t^n) \right) \right. \\
&\quad \left. + \left(\hat{F}_{j+\frac{1}{2}}^{(1)}(u^-, u^+; t^{n+1}) - \hat{F}_{j-\frac{1}{2}}^{(1)}(u^-, u^+; t^{n+1}) \right) \right], \tag{2.15}
\end{aligned}$$

where $\Delta \tilde{x}_j^{(k)} = \tilde{x}_{j+\frac{1}{2}}(t^{(k)}) - \tilde{x}_{j-\frac{1}{2}}(t^{(k)})$ denotes the cell lengths at stage k , $\hat{F}_{j\pm\frac{1}{2}}^{(k)}(u^-, u^+; t)$ denotes using the cell averages $\tilde{u}_j^{(k)}$ from stage k to approximate the limits $u_{j\pm\frac{1}{2}}^-$ and $u_{j\pm\frac{1}{2}}^+$ in the numerical flux function at time t using Algorithms 1 and 2. The Butcher tables for other explicit RK methods are provided in Appendix A.

Remark 1. If the approximate characteristics are defined such that $\nu_{j+\frac{1}{2}} = 0$ for all j , then the EL-RK-FV scheme reduces to the standard RK-FV scheme [43]. Referring to Figure 1, the approximate characteristics would be vertical lines.

Remark 2. In the presence of shocks, the approximate characteristics will intersect even under reasonable time-stepping sizes. The proposed EL-RK-FV scheme can handle shocks only under very small time-stepping sizes, making the scheme not ideal when going to large times after shock formation. The EL-RK-FV scheme is modified and extended to allow large time-stepping sizes in the presence of shocks in [16, 49]. In particular, it is proved that the first-order version of the modified scheme with forward Euler time discretization is total-variation-diminishing and maximum-principle-preserving under a time-stepping size at least twice as large as the CFL constraint for an Eulerian first-order finite volume method [49]. The method is extended to high-order accuracy with WENO type spatial reconstruction and RK time discretization in [16].

Remark 3. Once the space-time partition is defined, the EL-RK-FV scheme evolves from time t^n to time t^{n+1} in a similar fashion to ALE methods. In the ALE mindset, one can imagine that $\nu_{j+\frac{1}{2}} = 0$ fixes the grid point over this time interval (Eulerian), and $\nu_{j+\frac{1}{2}} \neq 0$ moves the grid point according to the approximate characteristic speed (Lagrangian). However, the goal of ALE methods is to move the grid to better resolve sharp gradients. Whereas, the EL-RK-FV method is designed to approximate the characteristics with the goal of large time-stepping sizes.

2.5 Two-dimensional problems

We now consider the two-dimensional equation

$$u_t + f(u)_x + g(u)_y = 0. \tag{2.16}$$

The spatial domain is discretized into $N_x N_y$ cells, $I_{i,j} := I_i \times I_j$ with centers $x_{i,j} = ((x_{i-\frac{1}{2}} + x_{i+\frac{1}{2}})/2, (y_{j-\frac{1}{2}} + y_{j+\frac{1}{2}})/2)$, where the spatial discretizations in x and y are respectively given by

$$0 = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \dots < x_{N_x - \frac{1}{2}} < x_{N_x + \frac{1}{2}} = 2\pi \quad \text{and} \quad 0 = y_{\frac{1}{2}} < y_{\frac{3}{2}} < \dots < y_{N_y - \frac{1}{2}} < y_{N_y + \frac{1}{2}} = 2\pi.$$

The CFL number is defined as

$$\Delta t = \frac{CFL}{\frac{\max |f'(u)|}{\Delta x} + \frac{\max |g'(u)|}{\Delta y}}, \quad (2.17)$$

and the uniform cell averages at time t^n are defined by

$$\tilde{u}_{i,j}^n := \frac{1}{\Delta x \Delta y} \int_{I_{i,j}} u(x, y, t^n) dx dy. \quad (2.18)$$

In the two-dimensional case, we also want to compute the *interval averages* over the interval I_i at a fixed y , or over the interval I_j at a fixed x . Define the uniform *interval averages* at time t^n with one variable fixed by

$$\bar{u}_{i|y}^n := \frac{1}{\Delta x} \int_{I_i} u(x, y, t^n) dx, \quad (2.19a)$$

$$\tilde{u}_{j|x}^n := \frac{1}{\Delta y} \int_{I_j} u(x, y, t^n) dy. \quad (2.19b)$$

Note that *only in this subsection* will the superscript tilde denote uniform interval averages in y , not nonuniform interval averages.

Dimensional splitting methods are commonly used to solve two (or higher) dimensional problems, with second-order Strang splitting being a popular choice. Dimensional splitting methods solve equation (2.16) by alternating between solving the easier problems

$$u_t + f(u)_x = 0, \quad (2.20a)$$

$$u_t + g(u)_y = 0. \quad (2.20b)$$

Strang splitting uses $\tilde{u}_{i,j}^n$ to solve (2.20a) over a half time step $\Delta t/2$ for intermediate solution $\tilde{u}_{i,j}^*$; uses $\tilde{u}_{i,j}^*$ to solve (2.20b) over a full time step Δt for intermediate solution $\tilde{u}_{i,j}^{**}$; and uses $\tilde{u}_{i,j}^{**}$ to solve (2.20a) over another half time step $\Delta t/2$ for solution $\tilde{u}_{i,j}^{n+1}$. As such, we can reduce the two-dimensional problem to solving several (easier) one-dimensional problems. Since the one-dimensional EL-RK-FV algorithm evolves interval averages, we need a way to go between two-dimensional cell averages and one-dimensional interval averages. We note that the nonlinear weights in the WENO-AO method [5] are the same for all nodes within the same cell.

2.5.1 Going from/to cell averages to/from interval averages

Consider the interval averages as functions of y and x , respectively defined by

$$\psi_i(y) := \bar{u}_{i|y}^n = \frac{1}{\Delta x} \int_{I_i} u(x, y, t^n) dx, \quad (2.21a)$$

$$\phi_j(x) := \bar{u}_{j|x}^n = \frac{1}{\Delta y} \int_{I_j} u(x, y, t^n) dy. \quad (2.21b)$$

For any given $i = 1, 2, \dots, N_x$ or $j = 1, 2, \dots, N_y$, consider the respective Gauss-Legendre quadrature nodes $\{x_k^{(i)} \in I_i : k = 1, \dots, K\}$ and $\{y_l^{(j)} \in I_j : l = 1, \dots, L\}$. Observing that the cell averages at time t^n can be expressed as the interval averages of $\psi(y)$ and $\phi(x)$,

$$\tilde{u}_{i,j}^n = \frac{1}{\Delta y} \int_{I_j} \psi_i(y) dy = \frac{1}{\Delta x} \int_{I_i} \phi_j(x) dx, \quad (2.22)$$

we can use WENO-AO to go from cell averages to interval averages evaluated at the Gauss-Legendre nodes,

$$\tilde{u}_{i,j}^n \longrightarrow \psi_i(y_l^{(j)}) = \bar{u}_{i|l}^n, \quad (2.23a)$$

$$\tilde{u}_{i,j}^n \longrightarrow \phi_j(x_k^{(i)}) = \bar{u}_{j|k}^n. \quad (2.23b)$$

Algorithm 3 presents how to implement WENO-AO to go from cell averages to x -interval averages at the fixed y -Gauss-Legendre nodes. A similar algorithm can be done to go from cell averages to y -interval averages at the fixed x -Gauss-Legendre nodes.

Algorithm 3. Going from cell averages to x –interval averages

Input: uniform cell averages $\tilde{u}_{i,j}$.

Output: uniform x –interval averages $\bar{u}_{i|l}$ at fixed Gauss-Legendre nodes $\{y_l^{(j)} \in I_j : l = 1, \dots, L\}$.

```

do  $i = 1, 2, \dots, N_x$ 
  do  $j = 1, 2, \dots, N_y$ 
    Calculate the WENO-AO reconstruction polynomial  $\mathcal{R}_j^{(i)}(y \in I_j)$  in terms
    of the
    neighboring averages  $\tilde{u}_{i,k}$ ,  $k = j - p, \dots, j + q$ .
    do  $l = 1, \dots, L$ 
      Store  $\bar{u}_{i|l} = \psi_i(y_l^{(j)}) \approx \mathcal{R}_j^{(i)}(y_l^{(j)})$ .
    end do
  end do
end do
The uniform  $x$ –interval averages at a fixed Gauss-Legendre node  $y_l^{(j)}$  are  $\{\bar{u}_{i|l} = \psi_i(y_l^{(j)}) : l = 1, 2, \dots, N_x\}$ .

```

Computing the cell averages from the interval averages at the Gauss-Legendre nodes is straightforward using a Gaussian quadrature. Let ξ and w denote the standard Gauss-Legendre nodes and weights over the interval $[-1, 1]$, respectively. Without loss of generality, we can go from x –interval averages to cell averages.

$$\begin{aligned}
\tilde{u}_{i,j} &= \frac{1}{\Delta y} \int_{I_j} \psi(y) dy \\
&= \frac{1}{2} \int_{-1}^1 \psi\left(y_{j-\frac{1}{2}} + \frac{\Delta y}{2}(y' + 1)\right) dy' \\
&\approx \frac{1}{2} \sum_{l=1}^L w_l \psi\left(y_{j-\frac{1}{2}} + \frac{\Delta y}{2}(\xi_l + 1)\right) \\
&= \frac{1}{2} \sum_{l=1}^L w_l \psi(y_l^{(j)}), \quad \text{where } y_l^{(j)} = y_{j-\frac{1}{2}} + \frac{\Delta y}{2}(\xi_l + 1).
\end{aligned} \tag{2.24}$$

2.5.2 Strang splitting

For demonstrative purposes, we outline the EL-RK-FV algorithm for two-dimensional problems via Strang splitting. Since Strang splitting is only second-order in time, higher-order splitting methods might be preferred. We present the fourth-order splitting method developed by Forest and Ruth [23] and Yoshida [50] in Appendix B; Rossmanith and Seal used this fourth-order splitting method in the semi-Lagrangian framework in [40].

Algorithm 4. Strang splitting for the EL-RK-FV method

Input: uniform cell averages $\tilde{u}_{i,j}^n$.

Output: uniform cell averages $\tilde{u}_{i,j}^{n+1}$.

Step 1 (x-direction). Solve equation (2.20a) for a half time step $\Delta t/2$; that is, over $[t^n, t^{n+1/2}]$.

Use Algorithm 3 to get x -interval averages; that is, $\tilde{u}_{i,j}^n \rightarrow \bar{u}_{i|l}^n$.

do $l = 1, 2, \dots, N_y \cdot L$

For each Gauss-Legendre node, update the x -interval averages by applying the 1D algorithm;

that is, $\bar{u}_{i|l}^n \rightarrow \bar{u}_{i|l}^*$.

end do

Use equation (2.24) to get updated cell averages; that is, $\bar{u}_{i|l}^* \rightarrow \tilde{u}_{i,j}^*$.

Step 2 (y-direction). Solve equation (2.20b) for a full time step Δt ; that is, over $[t^n, t^{n+1}]$.

Use Algorithm 3 analogue to get y -interval averages; that is, $\tilde{u}_{i,j}^* \rightarrow \tilde{u}_{j|k}^*$.

do $k = 1, 2, \dots, N_x \cdot K$

For each Gauss-Legendre node, update the y -interval averages by applying the 1D algorithm;

that is, $\tilde{u}_{j|k}^* \rightarrow \tilde{u}_{j|k}^{**}$.

end do

Use equation (2.24) analogue to get updated cell averages; that is, $\tilde{u}_{j|k}^{**} \rightarrow \tilde{u}_{i,j}^{**}$.

Step 3 (x-direction). Solve equation (2.20a) for a half time step $\Delta t/2$; that is, over $[t^{n+1/2}, t^{n+1}]$.

Use Algorithm 3 to get x -interval averages; that is, $\tilde{u}_{i,j}^{**} \rightarrow \bar{u}_{i|l}^{**}$.

do $l = 1, 2, \dots, N_y \cdot L$

For each Gauss-Legendre node, update the x -interval averages by applying the 1D algorithm;

that is, $\bar{u}_{i|l}^{**} \rightarrow \bar{u}_{i|l}^{n+1}$.

end do

Use equation (2.24) to get updated cell averages; that is, $\bar{u}_{i|l}^{n+1} \rightarrow \tilde{u}_{i,j}^{n+1}$.

3 The EL-RK-FV method for convection-diffusion equations

Throughout this section, overlines (e.g., $\bar{u}_j(t)$) denote *uniform* cell averages, and tildes (e.g., $\tilde{u}_j(t)$) denote *nonuniform* cell averages. We discuss the EL-RK-FV algorithm for

the convection-diffusion equation (1.1). Since we use dimensional splitting methods for two-dimensional problems, it suffices to discuss the 1D EL-RK-FV algorithm for problems of the form

$$u_t + f(u)_x = \epsilon u_{xx} + g(x, t), \quad (3.1)$$

where we impose periodic boundary conditions. Rewriting equation (3.1) in divergence form, integrating over the same space-time region Ω_j as in the $\epsilon = 0$ case, applying the divergence theorem and fundamental theorem of calculus, integrating over $[t^n, t^{n+1}]$, and converting to the time-differential form, we get the semi-discrete formulation

$$\frac{d}{dt} \int_{\tilde{I}_j(t)} u(x, t) dx = - \left[\hat{F}_{j+\frac{1}{2}}(t) - \hat{F}_{j-\frac{1}{2}}(t) \right] + \epsilon \int_{\tilde{I}_j(t)} u_{xx}(x, t) dx + \int_{\tilde{I}_j(t)} g(x, t) dx, \quad (3.2)$$

where $\hat{F}_{j+\frac{1}{2}}(t)$ is any monotone numerical flux (e.g., Lax-Friedrichs flux) and $F_{j+\frac{1}{2}}(t)$ is the modified flux function defined in (2.5).

3.1 Computing the uniform cell averages of u_{xx}

When evolving equation (3.2) we will need to compute the uniform cell averages of $u_{xx}(x, t)$, defined by

$$\bar{u}_{xx,j}(t) := \frac{1}{\Delta x} \int_{I_j} u_{xx}(x, t) dx. \quad (3.3)$$

We use the centered five-point stencil $\{j-2, j-1, j, j+1, j+2\}$ for linear reconstruction. Let $P(x \in I_j)$ be the linear reconstruction polynomial over the centered five-point stencil. After some tedious algebra, we find that the uniform cell averages $\bar{u}_{xx,j}(t)$ can be expressed in terms of the uniform cell averages $\bar{u}_j(t)$ with fourth-order accuracy using the equation

$$\bar{u}_{xx,j}(t) = \frac{1}{\Delta x^2} \left[-\frac{1}{12} \quad \frac{4}{3} \quad -\frac{5}{2} \quad \frac{4}{3} \quad -\frac{1}{12} \right] \begin{bmatrix} \bar{u}_{j-2}(t) \\ \bar{u}_{j-1}(t) \\ \bar{u}_j(t) \\ \bar{u}_{j+1}(t) \\ \bar{u}_{j+2}(t) \end{bmatrix}. \quad (3.4)$$

For implementation it is easier to express equation (3.4) in matrix form (assuming periodic or zero boundary conditions). We denote this matrix \mathbf{D}_4 in Algorithm 5. Computing the nonuniform cell averages $\tilde{u}_{xx,j}(t)$ can now be done using Algorithm 1.

3.2 Time evolution with implicit-explicit Runge-Kutta methods

We can split equation (3.1) into its non-stiff and stiff terms,

$$u_t = \mathcal{F}(u; x, t) + \mathcal{G}(u; x, t), \quad (3.5)$$

where $\mathcal{F}(u; x, t) = -f(u)_x$ is the non-stiff convective term, and $\mathcal{G}(u; x, t) = \epsilon u_{xx} + g(x, t)$ is the stiff diffusive (and source) term. Discretization methods used for such problems are implicit-explicit (IMEX) RK schemes [2, 18, 25]. The intuition behind these schemes is

straightforward – evolve the non-stiff term explicitly, and evolve the stiff term implicitly. As such, each stage in the RK method will involve explicitly evaluating the non-stiff term, and solving a linear system due to the stiff term.

All that's needed are the possibly nonuniform cell averages at each RK stage $\mu = 1, \dots, s$ over the space-time regions Ω_j . There are two approaches one can take to approximate the possibly nonuniform cell averages at each RK stage: (1) have a single space-time partition for multiple RK stages and directly update the solution to the possibly nonuniform traceback grid at each intermediate RK stage, or (2) have multiple space-time partitions for intermediate RK stages, and compute the uniform cell averages at time $t^{(\mu)}$ and project them onto the possibly nonuniform traceback grid via Algorithm 1. Although the first approach is more intuitive, it is computationally expensive since the system to be solved (from the implicit part) will depend on the measure of each cell. We choose the second approach, as computing the uniform cell averages at time $t^{(\mu)}$ requires solving a system dependent only on the background uniform mesh size Δx . Since we choose the second approach, we will need to: (1) solve for the *uniform* cell averages at each consecutive stage, and (2) project the uniform cell averages onto the possibly nonuniform traceback grid via Algorithm 1. Recall the uniform cell averages $\bar{u}_{xx,j}^{(\mu)}$ can be computed from the uniform cell averages $\bar{u}_j^{(\mu)}$ using equation (3.4).

For simplicity, we only use the IMEX RK schemes outlined in [2]. As per the notation used by Ascher, et al., $IMEX(s,\sigma,p)$ denotes using an s –stage diagonally-implicit Runge–Kutta (DIRK) scheme for $\mathcal{G}(u; x, t)$, using a σ –stage explicit RK scheme for $\mathcal{F}(u; x, t)$, and being of combined order p . Consider the semi-discrete formulation (3.2) rewritten as

$$\frac{d}{dt} \int_{\tilde{I}_j(t)} u(x, t) dx = \mathcal{F}(u; t) + \mathcal{G}(u; t), \quad (3.6)$$

where we redefine $\mathcal{F}(u; t) = -[\hat{F}_{j+\frac{1}{2}}(t) - \hat{F}_{j-\frac{1}{2}}(t)]$ and $\mathcal{G}(u; t) = \epsilon \int_{\tilde{I}_j(t)} u_{xx}(x, t) dx + \int_{\tilde{I}_j(t)} g(x, t) dx$. The IMEX(s,σ,p) RK schemes are expressed with two Butcher tables: one for the implicit RK method, and another for the explicit RK method.

Table 1: *

Implicit Scheme						
0	0	0	0	...	0	
c_1	0	a_{11}	0	...	0	
c_2	0	a_{21}	a_{22}	...	0	
\vdots	\vdots	\vdots	\ddots	\vdots		
c_s	0	a_{s1}	a_{s2}	...	a_{ss}	
	0	b_1	b_2	...	b_s	

Table 2: *

Explicit Scheme						
0	0	0	0	...	0	
c_1	\hat{a}_{21}	0	0	...	0	
c_2	\hat{a}_{31}	\hat{a}_{32}	0	...	0	
\vdots	\vdots	\vdots	\vdots	\ddots	\vdots	
c_s	$\hat{a}_{\sigma 1}$	$\hat{a}_{\sigma 2}$	$\hat{a}_{\sigma 3}$...	0	
	\hat{b}_1	\hat{b}_2	\hat{b}_3	...	\hat{b}_σ	

Defining $U^{(\mu)} := \int_{\tilde{I}_j(t^{(\mu)})} u(x, t^{(\mu)}) dx$, the IMEX(s,σ,p) scheme over the space-time region Ω_j

is as follows:

$$U^{n+1} = U^n + \Delta t \sum_{\mu=1}^s b_\mu K_\mu + \Delta t \sum_{\mu=1}^{\sigma} \hat{b}_\mu \hat{K}_\mu, \quad (3.7a)$$

$$K_\mu = \mathcal{G}(U^{(\mu)}; t^{(\mu)}), \quad \mu = 1, 2, \dots, s, \quad (3.7b)$$

$$\hat{K}_1 = \mathcal{F}(U^n; t^n), \quad (3.7c)$$

$$\hat{K}_{\mu+1} = \mathcal{F}(U^{(\mu)}; t^{(\mu)}), \quad \mu = 1, 2, \dots, s. \quad (3.7d)$$

More precisely,

$$K_\mu = \epsilon \int_{\tilde{I}_j(t^{(\mu)})} u_{xx}(x, t^{(\mu)}) dx + \int_{\tilde{I}_j(t^{(\mu)})} g(x, t^{(\mu)}) dx, \quad (3.8a)$$

$$\hat{K}_{\mu+1} = - \left[\hat{F}_{j+\frac{1}{2}}(t^{(\mu)}) - \hat{F}_{j-\frac{1}{2}}(t^{(\mu)}) \right]. \quad (3.8b)$$

Based on the IMEX RK method, the solution $U^{(\mu)}$ over the traceback grid can be approximated by

$$U^{(\mu)} = U^n + \Delta t \sum_{\nu=1}^{\mu} a_{\mu,\nu} K_\nu + \Delta t \sum_{\nu=1}^{\mu} \hat{a}_{\mu+1,\nu} \hat{K}_\nu, \quad \mu = 1, 2, \dots, s. \quad (3.9)$$

Recall that we choose *not* to directly update the solution over Ω_j . Instead, we opt to solve for the uniform cell averages at each RK stage and project them onto the corresponding possibly nonuniform traceback grid. We define sub-space-time regions ${}_\mu\Omega_j$ for the μ -th stage of the IMEX RK scheme, as shown in Figures 2 and 3. The space-time region ${}_\mu\Omega_j$ traces back to time t^n (using the same approximate characteristic speeds as in Ω_j) from time $t^{(\mu)}$. Hence, at time $t^{(\mu)}$ on the sub-space-time region ${}_\mu\Omega_j$ the grid is uniform. Lower left subscript μ denotes values confined to the sub-space-time region ${}_\mu\Omega_j$. For example, ${}_2U^n$ are the possibly nonuniform definite integrals (and hence the possibly nonuniform cell averages) at time t^n over the traceback cell in ${}_2\Omega_j$, as seen in Figure 3.

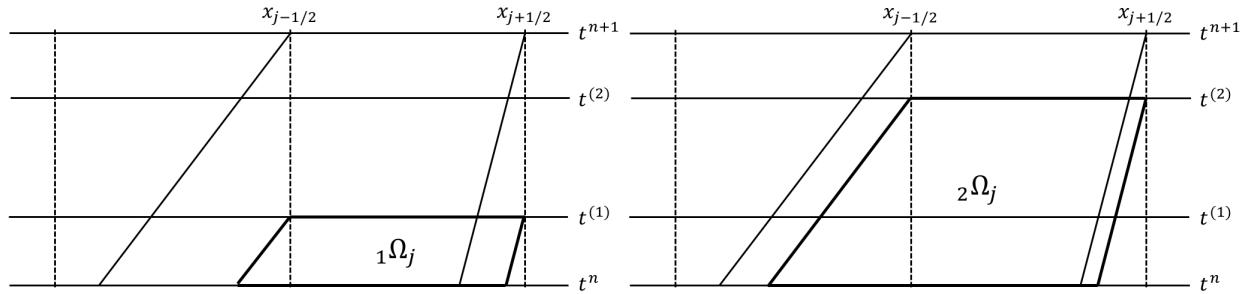


Figure 2: The space-time region ${}_1\Omega_j$.

Figure 3: The space-time region ${}_2\Omega_j$.

The desired uniform cell averages $\bar{u}_j^{(\mu)} = {}_\mu U^{(\mu)} / \Delta x$ at stage μ can be obtained by computing the first μ -stages of the IMEX RK scheme over the sub-space-time region ${}_\mu\Omega_j$, given below.

$${}_\mu U^{(\mu)} = {}_\mu U^n + \Delta t \sum_{\nu=1}^{\mu} (a_{\mu,\nu})({}_\mu K_\nu) + \Delta t \sum_{\nu=1}^{\mu} (\hat{a}_{\mu+1,\nu})({}_\mu \hat{K}_\nu) \quad (3.10a)$$

$${}_{\mu}K_{\nu} = \mathcal{G}({}_{\mu}U^{(\nu)}; t^{(\nu)}), \quad \nu = 1, 2, \dots, \mu, \quad (3.10b)$$

$${}_{\mu}\hat{K}_1 = \mathcal{F}({}_{\mu}U^n; t^n), \quad (3.10c)$$

$${}_{\mu}\hat{K}_{\nu+1} = \mathcal{F}({}_{\mu}U^{(\nu)}; t^{(\nu)}), \quad \nu = 1, 2, \dots, \mu. \quad (3.10d)$$

More precisely,

$${}_{\mu}K_1 = \epsilon \int_{\mu\tilde{I}_j(t^{(\nu)})} u_{xx}(x, t^{(\nu)}) dx + \int_{\mu\tilde{I}_j(t^{(\nu)})} g(x, t^{(\nu)}) dx, \quad (3.11a)$$

$${}_{\mu}\hat{K}_{\nu+1} = - \left[\hat{F}_{j+\frac{1}{2}}(t^{(\nu)}) - \hat{F}_{j-\frac{1}{2}}(t^{(\nu)}) \right]. \quad (3.11b)$$

Note that ${}_{\mu}\hat{K}_{\nu+1}$ uses the cell averages restricted to the sub-space-time regions ${}_{\mu}\Omega_j$. Further note that equation (3.10) can recycle and reuse the uniform cell averages already computed during stages $1, 2, \dots, \mu - 1$. Simply project the uniform cell averages $(\bar{u}_j^{(\nu)} \text{ or } \bar{u}_{xx,j}^{(\nu)})$, $\nu = 1, 2, \dots, \mu - 1$ onto the traceback grids formed by the space-time regions ${}_{\mu}\Omega_j$. Although IMEX RK schemes are straightforward, it is easy to get lost in the notation. To help demonstrate the EL-RK-FV algorithm coupled with an IMEX RK scheme, we present the IMEX(2,2,2) case in Appendix D. The Butcher tables for other IMEX RK schemes are included in Appendix C. There are IMEX SSP RK schemes that can be found in [18, 25].

Algorithm 5. EL-RK-FV algorithm coupled with an IMEX RK scheme [2], *IMEX*(s, σ, p)

Input: uniform cell averages \bar{u}_j^n .

Output: uniform cell averages \bar{u}_j^{n+1} .

Compute the possibly nonuniform traceback cell averages $\tilde{u}_j(t^n)$ using Algorithm 1.

Store $\hat{K}_1 = \mathcal{F}(U^n; t^n)$. This is the traceback grid in Ω_j .

do $\mu = 1, 2, \dots, s$

1. Compute the uniform cell averages $\bar{u}_j^{(\mu)}$ at time $t^{(\mu)}$ with equation (3.10) by:
 - i. Restrict yourself to the space-time region ${}_{\mu}\Omega_j$.
 - ii. Compute the values ${}_{\mu}K_{\nu}$ and ${}_{\mu}\hat{K}_{\nu+1}$ at each stage $\nu = 1, 2, \dots, \mu - 1$

using Algorithm 1

to compute the necessary nonuniform cell averages, and Algorithm 2 for the modified flux

function. Definite integrals of $g(x, t)$ can be computed using a Gaussian quadrature.

iii. Recalling equation (3.4), solve equation (3.10) by setting it up as the linear system

$$\left(\mathbf{I} - \frac{a_{\mu,\mu}\epsilon\Delta t}{\Delta x^2} \mathbf{D}_4 \right) {}_{\mu}\vec{U}^{(\mu)} = {}_{\mu}\vec{U}^n + \Delta t \sum_{\nu=1}^{\mu-1} (a_{\mu,\nu})({}_{\mu}K_{\nu}) + \Delta t \sum_{\nu=1}^{\mu} (\hat{a}_{\mu+1,\nu})({}_{\mu}\hat{K}_{\nu}) + a_{\mu,\mu}\Delta t \vec{g}(x, t^{(\mu)}),$$

where $\vec{g}_j(x, t^{(\mu)}) = \int_{I_j} g(x, t^{(\mu)}) dx$.

iv. Store the uniform cell averages $\bar{u}_j^{(\mu)} = {}_{\mu}U_j^{(\mu)} / \Delta x$.

2. Compute and store the uniform cell averages $\bar{u}_{xx,j}^{(\mu)}$ using equation (3.4).
3. Compute the possibly nonuniform traceback cell averages $\tilde{u}_j^{(\mu)}$ using Algorithm 1.
4. Compute the possibly nonuniform traceback cell averages $\tilde{u}_{xx,j}^{(\mu)}$ using Algorithm 1.
5. Compute and store K_μ and $\hat{K}_{\mu+1}$. Note that we are back on the possibly nonuniform traceback grid consisting of cells $\tilde{I}_j(t^{(\mu)})$.

end do

Compute $U^{n+1} = U^n + \Delta t \sum_{\mu=1}^s b_\mu K_\mu + \Delta t \sum_{\mu=1}^\sigma \hat{b}_\mu \hat{K}_\mu$.

Compute the uniform cell averages $\bar{u}_j^{n+1} = U_j^{n+1} / \Delta x$.

3.3 Mass conservation

We now show that the 1D EL-RK-FV algorithm is mass conservative when coupled with any IMEX RK scheme in [2]. Since we extend the EL-RK-FV algorithm to higher dimensions via dimensional splitting in this paper, showing mass conservation for the 1D problem suffices. Note that mass conservation of the $\epsilon = 0$ case can be proved just as easily.

Proposition 1. (Mass conservative). The 1D EL-RK-FV algorithm coupled with any IMEX RK scheme in [2] for (non)linear convection-diffusion equations is mass conservative, assuming the source term $g(x, t) = 0$ and periodic boundary conditions.

Proof. Making use of the semi-discrete form of the convection-diffusion equation,

$$\begin{aligned} \int_{I_j} u(x, t^{n+1}) dx &= \int_{\tilde{I}_j(t^n)} u(x, t^n) dx - \int_{t^n}^{t^{n+1}} \left\{ \hat{F}_{j+\frac{1}{2}}(t) - \hat{F}_{j-\frac{1}{2}}(t) - \epsilon \int_{\tilde{I}_j(t)} u_{xx}(x, t) dx \right\} dt \\ &= \int_{\tilde{I}_j(t^n)} u(x, t^n) dx - \int_{t^n}^{t^{n+1}} \left\{ \hat{F}_{j+\frac{1}{2}}(t) - \hat{F}_{j-\frac{1}{2}}(t) - \epsilon \left(u_x(\tilde{x}_{j+\frac{1}{2}}(t), t) - u_x(\tilde{x}_{j-\frac{1}{2}}(t), t) \right) \right\} dt. \end{aligned} \quad (3.12)$$

Summing over all $j = 1, 2, \dots, N_x$ and making use of the periodic boundary conditions,

$$\int_a^b u(x, t^{n+1}) dx = \int_a^b u(x, t^n) dx. \quad (3.13)$$

□

Remark 4. By virtue of approximating the exact characteristics, the CFL condition is much more relaxed when compared to the standard RK-FV method. On the other hand, Algorithm 2 is more expensive than a standard WENO procedure on a uniform grid. The computational savings of the EL-RK-FV method will be more significant for solving convection-diffusion

equations, as the computational overhead in the remapping algorithm is less significant compared with the implicit solver for the stiff diffusion term. In our computational experiments, we find that the proposed algorithm is most advantageous for convection-dominated problems. Moreover, it is advantageous to take larger time steps with numerical stability for convection-diffusion problems when convective and diffusive terms are of similar magnitude.

4 Numerical tests

In this section, we present results applying the proposed EL-RK-FV algorithm to various benchmark problems. In particular, we include error tables and error plots to investigate the spatial and temporal convergence of the algorithm. Mass conservation is also numerically verified by applying the proposed algorithm to the 0D2V (zero dimensions in physical space and two dimensions in velocity space) Leonard-Bernstein Fokker-Planck equation. We assume a uniform mesh, apply WENO-AO(5,3) in Algorithms 1 and 2, use three Gauss-Legendre nodes in Algorithm 3, and use the fourth-order approximation given by equation (3.4) for the diffusive term. Unless otherwise stated, for the time-stepping we use the fourth-order RK method for pure convection problems, and IMEX(2,3,3) for convection-diffusion equations. We also use second-order Strang splitting for two-dimensional convection-diffusion equations. Although higher-order splitting methods can be used for pure convection problems, it is well-known that negative time integration can lead to significant instabilities when dealing with a diffusive term.

There are three sources of error: spatial approximation, time-stepping, and splitting. Depending on the CFL number and test problem, these three sources of error will influence the observed order of convergence. We compute the L^1 , L^2 , and L^∞ errors (in one-dimension),

$$\|\bar{u} - \bar{u}_{exact}\|_1 = \Delta x \sum_{j=1}^{N_x} |\bar{u}_j - \bar{u}_{exact,j}| \quad (4.1a)$$

$$\|\bar{u} - \bar{u}_{exact}\|_2 = \sqrt{\Delta x \sum_{j=1}^{N_x} |\bar{u}_j - \bar{u}_{exact,j}|^2} \quad (4.1b)$$

$$\|\bar{u} - \bar{u}_{exact}\|_\infty = \max_{1 \leq j \leq N_x} |\bar{u}_j - \bar{u}_{exact,j}| \quad (4.1c)$$

Note that for the norms defined above, $\|\bar{u} - \bar{u}_{exact}\|_1 \leq |\mathcal{D}| \|\bar{u} - \bar{u}_{exact}\|_\infty$.

4.1 Pure convection problems: one-dimensional tests

Example 4.1. (1D transport with constant coefficient)

$$u_t + u_x = 0, \quad x \in [0, 2\pi] \quad (4.2)$$

with periodic boundary conditions and exact solution $u(x, t) = \sin(x - t)$. The errors provided in Table 3 verify the convergence of the EL-RK-FV method when using WENO-AO(5,3) and forward Euler. As expected, we see fifth-order convergence despite the large

CFL number. There is no temporal error for the convective part since the characteristics are traced exactly and hence $F_{j+\frac{1}{2}}(t) = 0$ for all $t \in [t^n, t^{n+1}]$ and $j = 1, 2, \dots, N_x$.

Table 3: Convergence study with spatial mesh refinement for equation (4.2) with forward Euler at $T_f = 1$.

<i>CFL = 8</i>						
N_x	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	1.09E-08	-	4.83E-09	-	2.86E-09	-
100	3.34E-10	5.03	1.48E-10	5.03	8.34E-11	5.10
200	9.86E-12	5.08	4.37E-12	5.08	2.51E-12	5.06
400	2.80E-13	5.14	1.43E-13	4.94	1.91E-13	3.72

Example 4.2. (1D transport with variable coefficient in space)

$$u_t + (\sin(x)u)_x = 0, \quad x \in [0, 2\pi] \quad (4.3)$$

with periodic boundary conditions and exact solution

$$u(x, t) = \frac{\sin(2 \arctan(e^{-t} \tan(x/2)))}{\sin(x)}.$$

As seen in Table 4, we observe the high-order convergence. As the CFL number (and hence the time step) increases, the temporal error starts to play more of a role, as evidenced by the fourth-order convergence. We verify the high-order temporal convergence in Figure 4 by fixing the mesh $N_x = 400$ and varying the CFL from 0.2 to 20.

Table 4: Convergence study with spatial mesh refinement for equation (4.3) with RK4 at $T_f = 1$.

<i>CFL = 0.3</i>						
N_x	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	2.76E-04	-	2.53E-04	-	3.42E-04	-
100	3.05E-06	6.50	2.53E-06	6.64	2.94E-06	6.87
200	9.78E-08	4.96	7.90E-08	5.00	9.86E-08	4.90
400	3.24E-09	4.91	2.59E-09	4.93	3.23E-09	4.93
<i>CFL = 8</i>						
N_x	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	6.81E-02	-	4.11E-02	-	4.51E-02	-
100	1.18E-03	5.85	6.23E-04	6.04	6.38E-04	6.15
200	5.89E-05	4.32	3.63E-05	4.10	5.46E-05	3.54
400	3.71E-06	3.99	2.57E-06	3.82	4.31E-06	3.66

Example 4.3. (1D transport with variable coefficient in time)

$$u_t + \left(\frac{u}{t+1} \right)_x = 0, \quad x \in [0, 2\pi] \quad (4.4)$$

with periodic boundary conditions and exact solution $u(x, t) = \exp(-5(x - \log(t+1) - \pi)^2)$. Periodic boundary conditions are a valid assumption for sufficiently thin Gaussian curves and small enough times. The expected high-order convergence for both small and large CFL numbers is seen in Table 5. Fixing the mesh $N_x = 400$ and varying the CFL number from 0.2 to 20, fifth-order convergence in time is seen in Figure 5. Observe that there are two optimal CFL numbers for this mesh.

Table 5: Convergence study with spatial mesh refinement for equation (4.4) with RK4 at $T_f = 1$.

<i>CFL = 0.95</i>						
N_x	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	6.36E-03	-	5.80E-03	-	8.23E-03	-
100	2.37E-04	4.75	2.28E-04	4.67	5.33E-04	3.95
200	1.71E-06	7.12	1.30E-06	7.46	2.36E-06	7.82
400	4.40E-08	5.28	3.84E-08	5.08	6.02E-08	5.29
<i>CFL = 8</i>						
N_x	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	6.71E-02	-	5.72E-02	-	8.15E-02	-
100	7.74E-04	6.44	6.48E-04	6.46	1.06E-03	6.27
200	1.52E-05	5.67	1.26E-05	5.69	1.76E-05	5.91
400	9.65E-07	3.98	8.11E-07	3.96	9.61E-07	4.20

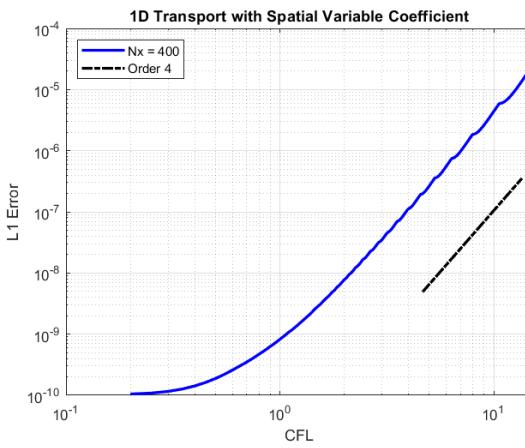


Figure 4: RK4, Final time $T_f = 0.5$.

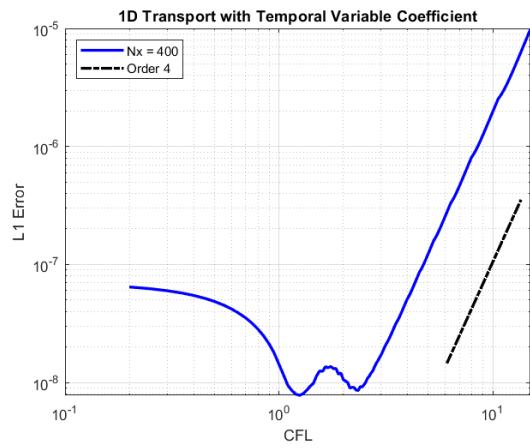


Figure 5: RK4, Final time $T_f = 0.5$.

4.2 Hyperbolic conservation laws: two-dimensional tests

Example 4.4. (2D transport with constant coefficient)

$$u_t + u_x + u_y = 0, \quad x, y \in [-\pi, \pi] \quad (4.5)$$

with periodic boundary conditions and exact solution $u(x, y, t) = \sin(x + y - 2t)$. The expected high-order convergence is shown in Table 6 when using Strang splitting. Just like equation (4.2), there is no temporal error for the convective part since the characteristics are traced exactly.

Table 6: Convergence study with spatial mesh refinement for equation (4.5) with forward Euler and $CFL = 8$ at $T_f = 1$.

Strang splitting						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
100	4.24E-09	-	7.49E-10	-	1.69E-10	-
200	1.27E-10	5.05	2.25E-11	5.05	5.15E-12	5.03
300	1.60E-11	5.11	2.84E-12	5.11	8.23E-13	4.52
400	3.57E-12	5.21	6.88E-13	4.92	3.68E-13	2.80

Example 4.5. (Rigid body rotation)

$$u_t - yu_x + xu_y = 0, \quad x, y \in [-\pi, \pi] \quad (4.6)$$

with periodic boundary conditions. We choose the exact solution $u(x, y, t) = u(x, y, t = 0) = \exp(-3(x^2 + y^2))$ for convergence tests. The convergence results are presented in Tables 7 and 8. Strang splitting dominates the error and we observe the expected second-order convergence. Whereas, the spatial error dominates when using fourth-order splitting as evidenced by the fifth-order convergence. The error plot using a fixed mesh $N_x = N_y = 200$ and varying the CFL number from 0.1 to 50 is shown in Figure 6. Second-order convergence in time is observed when using Strang splitting, and fourth-order convergence is observed when using fourth-order splitting. We note that comparable convergence results were observed for the non-symmetric initial condition $u(x, y, t = 0) = \exp(-3x^2 - 2y^2)$. To demonstrate the effectiveness of WENO-AO controlling spurious oscillations we choose the initial condition $u(x, y, t = 0) = 1$ if $x, y \in [-\pi/2, \pi/2]$; $u(x, y, t = 0) = 0$ otherwise. With a fixed mesh $N_x = N_y = 100$ and $CFL = 2.2$, we compute the solution up to time $T_f = 2\pi$ using SSP RK3. The discontinuities are smoothed out and no spurious oscillations occur.

Example 4.6. (Swirling deformation)

$$u_t - \left(\cos^2(x/2) \sin(y) g(t) u \right)_x + \left(\sin(x) \cos^2(y/2) g(t) u \right)_y = 0, \quad x, y \in [-\pi, \pi] \quad (4.7)$$

Table 7: Convergence study with spatial mesh refinement for equation (4.6) with RK4 and $CFL = 0.95$ at $T_f = 0.5$.

Strang splitting						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
100	3.57E-05	-	1.54E-05	-	2.43E-05	-
200	1.83E-06	4.29	9.94E-07	3.95	1.23E-06	4.31
300	8.01E-07	2.04	4.35E-07	2.04	4.72E-07	2.36
400	4.50E-07	2.01	2.44E-07	2.01	2.55E-07	2.13
Fourth-order splitting						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
100	1.05E-04	-	4.66E-05	-	8.95E-05	-
200	1.08E-06	6.60	6.13E-07	6.25	7.74E-07	6.85
300	1.39E-07	5.06	8.12E-08	4.99	1.02E-07	5.00
400	3.32E-08	4.99	1.93E-08	4.99	2.43E-08	4.99

Table 8: Convergence study with spatial mesh refinement for equation (4.6) with RK4 and $CFL = 8$ at $T_f = 0.5$.

Strang splitting						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
100	4.97E-04	-	2.68E-04	-	2.78E-04	-
200	1.24E-04	2.00	6.74E-05	1.99	6.86E-05	2.02
300	5.59E-05	1.97	3.03E-05	1.97	3.08E-05	1.97
400	3.20E-05	1.94	1.73E-05	1.94	1.76E-05	1.94
Fourth-order splitting						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
100	4.95E-05	-	2.46E-05	-	6.82E-05	-
200	4.84E-07	6.68	2.92E-07	6.39	4.77E-07	7.16
300	6.17E-08	5.08	3.86E-08	4.99	6.19E-08	5.03
400	1.47E-08	4.99	9.22E-09	4.88	1.47E-08	5.00

When testing convergence we set $g(t) = \cos(\pi t/T_f)\pi$ and choose the initial condition to be the smooth (with C^5 smoothness) cosine bell

$$u(x, y, t = 0) = \begin{cases} r_0^b \cos^6 \left(\frac{r^b(x, y)\pi}{2r_0^b} \right), & \text{if } r^b(x, y) < r_0^b, \\ 0, & \text{otherwise,} \end{cases} \quad (4.8)$$

where $r_0^b = 0.3\pi$ and $r^b(x, y) = \sqrt{(x - x_0^b)^2 + (y - y_0^b)^2}$ with $(x_0^b, y_0^b) = (0.3\pi, 0)$. The convergence results under spatial mesh refinement are presented in Tables 9 and 10. Surprisingly, high-order convergence is observed in all test cases, even for the large CFL number of 8. In particular, we observed super-convergence for $CFL = 8$ when using Strang splitting. We got comparable convergence results when letting the initial condition be: (1) a cosine bell

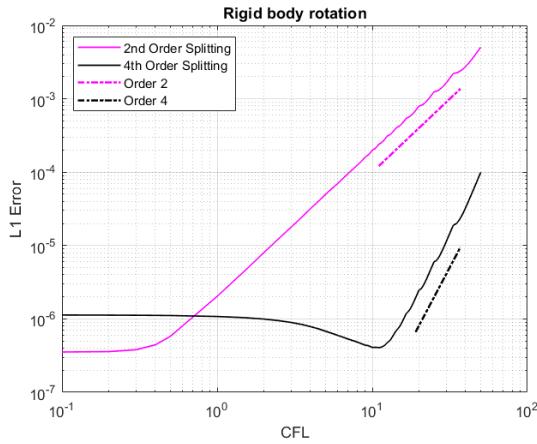


Figure 6: Error plot for (4.6) with RK4 at $T_f = 0.5$. $N_x = N_y = 200$.

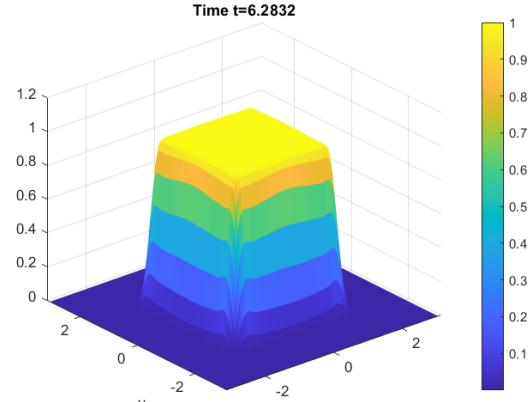


Figure 7: Plot of the numerical solution to (4.6) with SSP RK3 and $CFL = 2.2$ at $T_f = 2\pi$. $N_x = N_y = 100$.

of C^3 smoothness, and (2) the cosine bell (4.8) but with $x_0^b = 0.6\pi$ and the width in the x -direction scaled by a factor of $1/2$.

The temporal orders of convergence are shown in Figure 8 using a fixed mesh $N_x = N_y = 200$ and varying the CFL number from 0.1 to 25. When using Strang splitting the temporal convergence switches from second-order to third-order, indicating that for very large CFL numbers the splitting error does not dominate the time-stepping error as much. Fourth-order convergence is observed when using fourth-order splitting. To demonstrate the effectiveness of WENO-AO in controlling spurious oscillations we set $g(t) = 1$ and choose the initial condition [30]

$$u(x, y, t = 0) = \begin{cases} 1, & \text{if } r^b(x, y) < r_0^b, \\ 0, & \text{otherwise,} \end{cases} \quad (4.9)$$

where $r_0^b = 8\pi/5$ and $r^b(x, y) = \sqrt{(x - x_0^b)^2 + (y - y_0^b)^2}$ with $(x_0^b, y_0^b) = (\pi, \pi)$. With a fixed mesh $N_x = N_y = 100$ and $CFL = 8$, we compute the solution up to time $T_f = 5\pi$ using SSP RK3 and Strang splitting. The discontinuities are smoothed out and no spurious oscillations occur.

4.3 Convection-diffusion equations: one-dimensional tests

Example 4.7. (1D equation with constant coefficient)

$$u_t + u_x = \epsilon u_{xx}, \quad x \in [0, 2\pi] \quad (4.10)$$

with periodic boundary conditions and exact solution $u(x, t) = \sin(x - t)\exp(-\epsilon t)$. We set $\epsilon = 1$. The convergence results under spatial mesh refinement are shown in Table 11 for $CFL = 0.95$ and $CFL = 8$. In both cases we observe the expected third-order convergence since we are using IMEX(2,3,3) for the time-stepping. Note that there is no temporal error

Table 9: Convergence study with spatial mesh refinement for equation (4.7) with RK4 and $CFL = 0.95$ at $T_f = 1.5$.

Strang splitting						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
100	5.17E-03	-	6.11E-03	-	2.04E-02	-
200	1.69E-04	4.94	1.69E-04	5.18	4.80E-04	5.41
300	3.12E-05	4.16	3.85E-05	3.64	1.41E-04	3.01
400	8.29E-06	4.61	1.12E-05	4.66	4.66E-05	3.86
Fourth-order splitting						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
100	1.42E-02	-	1.73E-02	-	5.25E-02	-
200	3.98E-04	5.16	3.70E-04	5.54	9.15E-04	5.84
300	7.69E-05	4.06	9.02E-05	3.48	3.24E-04	2.56
400	2.20E-05	4.35	2.89E-05	3.96	1.17E-04	3.52

Table 10: Convergence study with spatial mesh refinement for equation (4.7) with RK4 and $CFL = 8$ at $T_f = 1.5$.

Strang splitting						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
100	1.90E-03	-	2.11E-03	-	6.83E-03	-
200	1.02E-04	4.23	8.88E-05	4.57	2.47E-04	4.79
300	1.90E-05	4.13	1.79E-05	3.94	6.36E-05	3.35
400	2.82E-06	6.63	4.11E-06	5.12	1.98E-05	4.05
Fourth-order splitting						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
100	3.89E-03	-	4.31E-03	-	1.43E-02	-
200	1.30E-04	4.90	1.34E-04	5.01	4.42E-04	5.01
300	2.41E-05	4.16	3.32E-05	3.44	1.40E-04	2.84
400	6.54E-06	4.53	1.02E-05	4.10	4.81E-05	3.70

for the convective part since the characteristics are traced exactly and hence $F_{j+\frac{1}{2}}(t) = 0$ for all $t \in [t^n, t^{n+1}]$ and $j = 1, 2, \dots, N_x$. Figure 10 shows the expected third-order convergence in time using fixed mesh $N_x = 400$ and varying the CFL number from 0.1 to 15.

Example 4.8. (1D equation with variable coefficient)

$$u_t + (\sin(x)u)_x = \epsilon u_{xx} + g, \quad x \in [0, 2\pi] \quad (4.11)$$

with periodic boundary conditions and $g(x, t) = \sin(2x)\exp(-\epsilon t)$ and exact solution $u(x, t) = \sin(x)\exp(-\epsilon t)$. We set $\epsilon = 1$. Table 12 shows the convergence results under spatial mesh refinement. Third-order convergence in space is observed for $CFL = 0.95$. Whereas, the

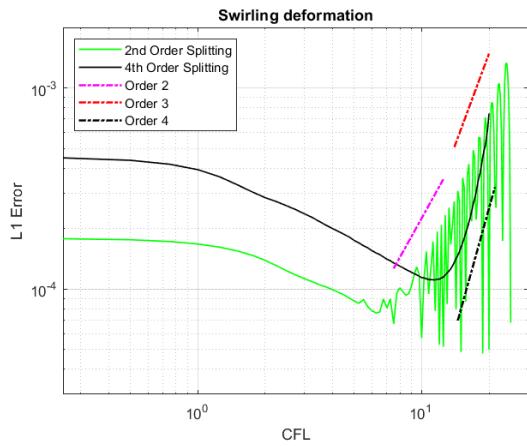


Figure 8: Error plot for (4.7) with RK4 at $T_f = 1.5$. $N_x = N_y = 200$.

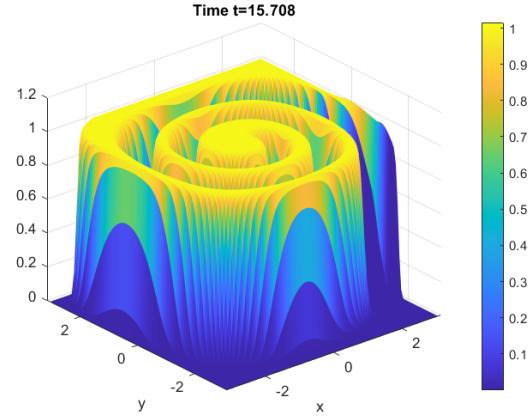


Figure 9: Plot of the numerical solution to (4.7) with $g(t) = 1$, SSP RK3 and $CFL = 8$ at $T_f = 5\pi$. $N_x = N_y = 100$.

Table 11: Convergence study with spatial mesh refinement for equation (4.10) with IMEX(2,3,3) at $T_f = 1$.

$CFL = 0.95$						
N_x	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	1.87E-04	-	8.26E-05	-	4.66E-05	-
100	2.55E-05	2.87	1.13E-05	2.87	6.36E-06	2.87
200	3.34E-06	2.93	1.48E-06	2.93	8.35E-07	2.93
400	4.31E-07	2.95	1.91E-07	2.95	1.08E-07	2.95
800	5.44E-08	2.99	2.41E-08	2.99	1.36E-08	2.99
$CFL = 8$						
N_x	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	6.87E-02	-	3.04E-02	-	1.72E-02	-
100	1.09E-02	2.66	4.82E-03	2.66	2.72E-03	2.66
200	1.63E-03	2.74	7.22E-04	2.74	4.07E-04	2.74
400	2.27E-04	2.84	1.01E-04	2.84	5.69E-05	2.84
800	3.03E-05	2.91	1.34E-05	2.91	7.57E-06	2.91

convergence for $CFL = 8$ is roughly order 2.6 since the time-stepping start to dominate. We note that the order of convergence for IMEX(2,3,3) under increasing the CFL number dips slightly below three for larger CFL numbers. We use fixed mesh $N_x = 400$ and vary the CFL number from 0.1 to 15 for the error plot showing the temporal order of convergence in Figure 11.

Example 4.9. (1D viscous Burgers' equation)

$$u_t + \left(\frac{u^2}{2} \right)_x = \epsilon u_{xx}, \quad x \in [0, 2] \quad (4.12)$$

Table 12: Convergence study with spatial mesh refinement for equation (4.11) with IMEX(2,3,3) at $T_f = 1$.

$CFL = 0.95$						
N_x	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	2.10E-03	-	9.99E-04	-	7.70E-04	-
100	3.99E-04	2.39	1.88E-04	2.41	1.44E-04	2.42
200	6.41E-05	2.64	2.99E-05	2.65	2.27E-05	2.66
400	9.84E-06	2.70	4.57E-06	2.71	3.44E-06	2.73
800	1.30E-06	2.92	6.04E-07	2.92	4.53E-07	2.92
$CFL = 8$						
N_x	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	7.50E-01	-	4.34E-01	-	3.92E-01	-
100	1.26E-01	2.58	6.60E-02	2.72	6.63E-02	2.56
200	1.74E-02	2.85	8.31E-03	2.99	6.50E-03	3.35
400	3.09E-03	2.50	1.46E-03	2.51	1.12E-03	2.53
800	5.03E-04	2.62	2.36E-04	2.63	1.81E-04	2.63

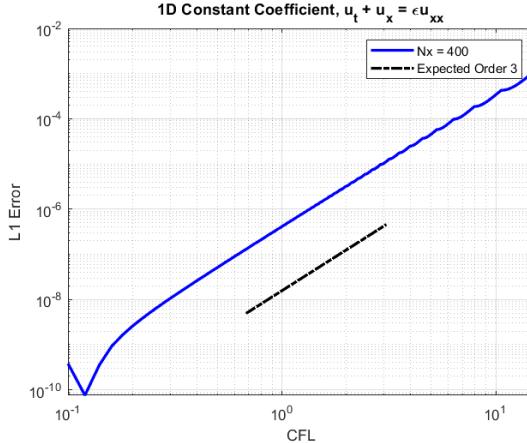


Figure 10: IMEX(2,3,3), $\epsilon = 1$, Final time $T_f = 0.5$.

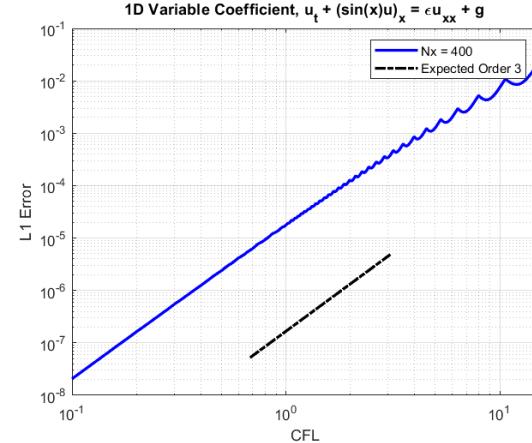


Figure 11: IMEX(2,3,3), $\epsilon = 1$, Final time $T_f = 0.5$.

with periodic boundary conditions. As in [42], we set $\epsilon = 0.1$ and choose the initial condition $u(x, t = 0) = 0.2 \sin(\pi x)$. The exact solution is

$$u(x, t) = 2\epsilon\pi \frac{\sum_{n=1}^{\infty} c_n \exp(-n^2\pi^2\epsilon t) n \sin(n\pi x)}{c_0 + \sum_{n=1}^{\infty} c_n \exp(-n^2\pi^2\epsilon t) \cos(n\pi x)},$$

where $c_0 = \int_0^1 \exp(-(1 - \cos(\pi x))/(10\pi\epsilon)) dx$ and $c_n = 2 \int_0^1 \exp(-(1 - \cos(\pi x))/(10\pi\epsilon)) \cos(n\pi x) dx$ for $n = 1, 2, 3, \dots$. We computed the first ten Fourier coefficients in Mathematica[®] for the exact solution; the eleventh Fourier coefficient was less than machine precision.

Table 13 shows the expected orders of convergence under spatial mesh refinement. Third-order convergence is observed for $CFL = 0.95$. Whereas, the convergence for $CFL = 8$ is slightly below order three since the time-stepping error starts to dominate. We note that the order of convergence for IMEX(2,3,3) under increasing the CFL number dips slightly below three for larger CFL numbers. The error plot in Figure 12 showing third-order convergence in time uses mesh $N_x = 400$ and CFL numbers varying from 0.1 to 15.

Table 13: Convergence study with spatial mesh refinement for equation (4.12) with IMEX(2,3,3) at $T_f = 1$.

$CFL = 0.95$						
N_x	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	9.50E-05	-	8.43E-05	-	1.17E-04	-
100	1.48E-05	2.68	1.32E-05	3.67	1.87E-05	2.65
200	2.42E-06	2.62	2.20E-06	2.59	3.21E-06	2.55
400	3.27E-07	2.88	3.00E-07	2.87	4.42E-07	2.86
800	4.28E-08	2.94	3.95E-08	2.93	5.86E-08	2.92
$CFL = 8$						
N_x	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	1.26E-02	-	1.31E-02	-	2.24E-02	-
100	2.79E-03	2.17	2.50E-03	2.38	3.70E-03	2.60
200	4.73E-04	2.56	4.16E-04	2.59	5.64E-04	2.71
400	1.28E-04	1.89	1.15E-04	1.85	1.64E-04	1.78
800	1.97E-05	2.70	1.78E-05	2.70	2.56E-05	2.68

Example 4.10. (The 0D1V Leonard-Bernstein (linearized) Fokker-Planck equation)

$$f_t - \frac{1}{\epsilon}((v_x - \bar{v}_x)f)_{v_x} = \frac{1}{\epsilon}Df_{v_x v_x}, \quad v_x \in [-2\pi, 2\pi] \quad (4.13)$$

with zero boundary conditions and equilibrium solution the Maxwellian

$$f_M(v_x) = \frac{n}{\sqrt{2\pi RT}} \exp\left(-\frac{(v_x - \bar{v}_x)^2}{2RT}\right), \quad (4.14)$$

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 velocity $\bar{v}_x = 0$. These quantities were chosen for scaling convenience. When testing convergence we set the initial distribution $f(v_x, t = 0) = f_M(v_x)$. Table 14 shows the convergence results, for which we use IMEX(4,4,3) for the time-stepping; we show the results using IMEX(4,4,3) because it gave slightly better convergence than IMEX(2,3,3). We observe fourth-order convergence under spatial mesh refinement for $CFL = 0.95$. Whereas, for $CFL = 8$ the time-stepping error starts to dominate and we observe third-order convergence. The error plot in Figure 13 showing third-order convergence

in time uses a fixed mesh $N_{v_x} = 400$ and CFL numbers varying from 0.1 to 15. We note that although high-order convergence is observed, the proposed EL-RK-FV algorithm is not equilibrium-preserving.

Table 14: Convergence study with spatial mesh refinement for equation (4.13) with IMEX(4,4,3) at $T_f = 1$.

$CFL = 0.95$						
N_{v_x}	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	8.02E-04	-	5.19E-04	-	5.65E-04	-
100	6.12E-05	3.71	3.83E-05	3.76	4.27E-05	3.73
200	4.41E-06	3.79	2.63E-06	3.87	2.84E-06	3.91
400	2.99E-07	3.88	1.73E-07	3.93	1.80E-07	3.98
800	2.03E-08	3.88	1.13E-08	3.94	1.10E-08	4.04
$CFL = 8$						
N_{v_x}	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	9.33E-03	-	4.52E-03	-	3.63E-03	-
100	1.34E-03	2.80	6.60E-04	2.78	5.63E-04	2.69
200	1.91E-04	2.81	9.57E-05	2.79	8.15E-05	2.79
400	3.21E-05	2.57	1.61E-05	2.57	1.34E-05	2.61
800	4.13E-06	2.96	2.09E-06	2.95	1.74E-06	2.94

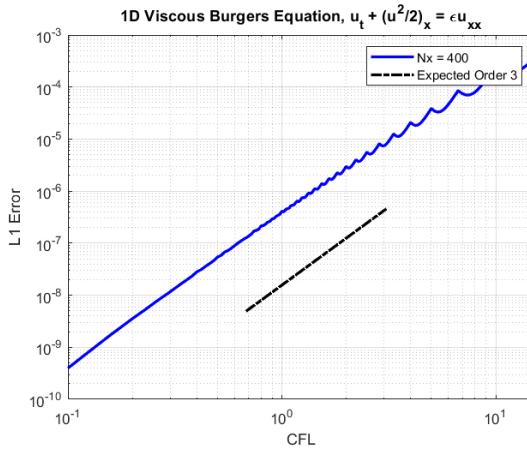


Figure 12: IMEX(2,3,3), $\epsilon = 0.1$, Final time $T_f = 0.5$.

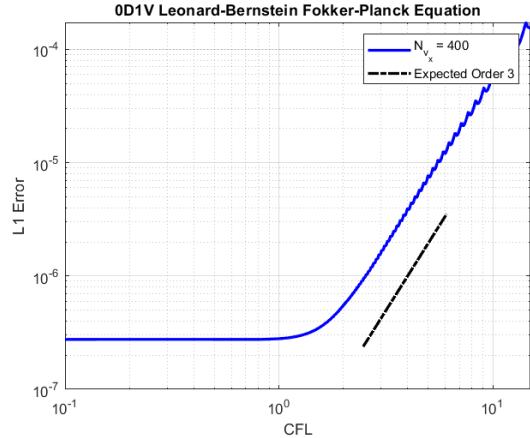


Figure 13: IMEX(4,4,3), Final time $T_f = 0.5$.

4.4 Convection-diffusion equations: two-dimensional tests

Example 4.11. (2D equation with constant coefficient)

$$u_t + u_x + u_y = \epsilon(u_{xx} + u_{yy}), \quad x, y \in [0, 2\pi] \quad (4.15)$$

with periodic boundary conditions and exact solution $u(x, y, t) = \exp(-2\epsilon t) \sin(x + y - 2t)$. We set $\epsilon = 1$. Third-order convergence under spatial mesh refinement is seen in Table 15 for $CFL = 0.95$ and $CFL = 8$. As with equation (4.10), there is no temporal error for the convective part since the characteristics are traced exactly. Note that the error is larger for $CFL = 8$ than $CFL = 0.95$ since this problem also has diffusion. Figure 14 shows the third-order convergence in time using fixed mesh $N_x = N_y = 400$ and varying the CFL number from 6 to 20.

Table 15: Convergence study with spatial mesh refinement for equation (4.15) with IMEX(2,3,3) and Strang splitting at $T_f = 0.5$.

$CFL = 0.95$						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	6.74E-05	-	1.19E-05	-	2.68E-06	-
100	1.02E-05	2.72	1.81E-06	2.72	4.07E-07	2.72
200	1.41E-06	2.86	2.49E-07	2.86	5.62E-08	2.86
400	1.86E-07	2.92	3.29E-08	2.92	7.41E-09	2.92
$CFL = 8$						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	3.92E-02	-	6.94E-03	-	1.56E-03	-
100	5.82E-03	2.75	1.03E-03	2.75	2.32E-04	2.75
200	8.09E-04	2.85	1.43E-04	2.85	3.22E-05	2.85
400	1.07E-04	2.92	1.90E-05	2.92	4.27E-06	2.92

Example 4.12. (Rigid body rotation with diffusion)

$$u_t - yu_x + xu_y = \epsilon(u_{xx} + u_{yy}) + g, \quad x, y \in [-2\pi, 2\pi] \quad (4.16)$$

with periodic boundary conditions, $g(x, y, t) = (6\epsilon - 4xy - 4\epsilon(x^2 + 9y^2))\exp(-(x^2 + 3y^2 + 2\epsilon t))$, and exact solution $u(x, y, t) = \exp(-(x^2 + 3y^2 + 2\epsilon t))$. We set $\epsilon = 1$. Table 16 shows the order of convergence when using IMEX(4,4,3). We use IMEX(4,4,3) instead of IMEX(2,3,3) because the latter choice, along with the Strang splitting, showed an order of convergence less than two for large CFL numbers. The expected second-order convergence in time (due to Strang splitting) is seen in Figure 15 assumes fixed mesh $N_x = N_y = 400$ and CFL numbers varying from 6 to 20.

Example 4.13. (Swirling deformation with diffusion)

$$u_t - \left(\cos^2(x/2)\sin(y)f(t)u\right)_x + \left(\sin(x)\cos^2(y/2)f(t)u\right)_y = \epsilon(u_{xx} + u_{yy}), \quad x, y \in [-\pi, \pi] \quad (4.17)$$

When testing the convergence we set $f(t) = \cos(\pi t/T_f)\pi$, $\epsilon = 1$, and choose the initial condition to be the cosine bell in equation (4.8). Since there is no analytic solution, we use a reference solution computed with a mesh size of 400×400 and $CFL = 0.1$. The convergence

Table 16: Convergence study with spatial mesh refinement for equation (4.16) with IMEX(4,4,3) and Strang splitting at $T_f = 0.5$.

$CFL = 0.95$						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	3.22E-03	-	1.42E-03	-	1.44E-03	-
100	3.92E-04	3.04	1.75E-04	3.02	2.00E-04	2.85
200	7.27E-05	2.43	3.14E-05	2.47	3.54E-05	2.50
400	1.65E-05	2.14	7.11E-06	2.15	7.80E-06	2.18
$CFL = 5$						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	2.21E-02	-	8.81E-03	-	7.44E-03	-
100	5.98E-03	1.89	2.48E-03	1.83	2.46E-03	1.60
200	1.61E-03	1.89	6.81E-04	1.86	7.14E-04	1.79
400	4.24E-04	1.93	1.81E-04	1.91	1.94E-04	1.88

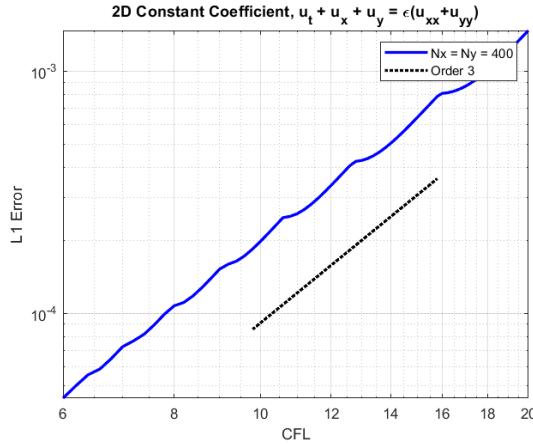


Figure 14: IMEX(2,3,3), $\epsilon = 1$, Final time $T_f = 0.5$.

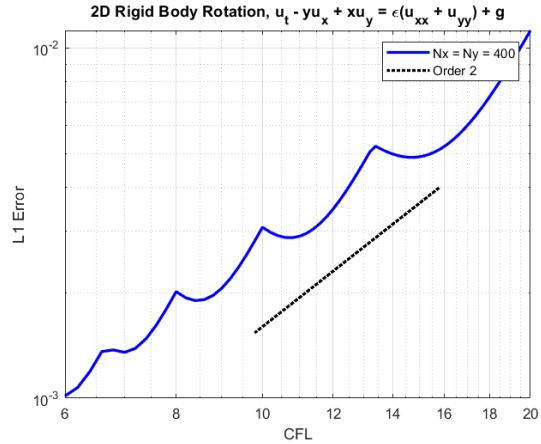


Figure 15: IMEX(4,4,3), $\epsilon = 1$, Final time $T_f = 0.1$.

results under spatial mesh refinement are presented in Table 17. The splitting error seems to dominate the time-stepping error for $CFL = 0.95$ as evidenced by the apparent second-order convergence. Whereas, the time-stepping error seems to contribute more for $CFL = 8$. Due to the interplay between the time-stepping and splitting errors, the temporal order 2.4 is also observed in Figure 16, for which we use fixed mesh $N_x = N_y = 400$ and CFL numbers varying from 6 to 20.

Example 4.14. (2D viscous Burgers' equation)

$$u_t + \left(\frac{u^2}{2} \right)_x + \left(\frac{u^2}{2} \right)_y = \epsilon(u_{xx} + u_{yy}) + g, \quad x, y \in [-\pi, \pi] \quad (4.18)$$

with periodic boundary conditions. As in [46], we set $\epsilon = 0.1$, $g(x, y, t) = \exp(-4\epsilon t) \sin(2(x + y))$,

Table 17: Convergence study with spatial mesh refinement for equation (4.17) with IMEX(2,3,3) and Strang splitting at $T_f = 0.1$.

$CFL = 0.95$						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	4.98E-04	-	2.57E-04	-	3.38E-04	-
100	9.13E-05	2.45	4.58E-05	2.49	6.05E-05	2.48
200	1.91E-05	2.26	9.56E-06	2.26	1.22E-05	2.31
400	4.48E-06	2.09	2.19E-06	2.13	2.61E-06	2.23
$CFL = 8$						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	5.93E-02	-	3.60E-02	-	5.82E-02	-
100	2.05E-02	1.53	1.16E-02	1.63	1.65E-02	1.82
200	3.14E-03	2.71	1.67E-03	2.80	2.04E-03	3.01
400	6.08E-04	2.37	3.13E-04	2.41	4.16E-04	2.30

and suppose the solution $u(x, y, t) = \exp(-2\epsilon t) \sin(x + y)$. The convergence results are presented in Table 18. The splitting error seems to dominate the time-stepping error for $CFL = 0.95$ as evidenced by the second-order convergence. Whereas, the time-stepping error seems to contribute more for $CFL = 8$ since the order is between two and three. The temporal order of convergence in the L^1 norm is roughly 2.3, as seen in Figure 17, for which we use fixed mesh $N_x = N_y = 400$ and CFL numbers varying from 6 to 20.

Table 18: Convergence study with spatial mesh refinement for equation (4.18) with IMEX(2,3,3) and Strang splitting at $T_f = 0.5$.

$CFL = 0.95$						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	3.01E-04	-	5.21E-05	-	1.19E-05	-
100	7.12E-05	2.08	1.30E-05	2.00	3.42E-06	1.80
200	1.76E-05	2.02	3.34E-06	1.96	9.42E-07	1.86
400	4.53E-06	1.95	8.60E-07	1.96	2.49E-07	1.92
$CFL = 8$						
$N_x = N_y$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	8.11E-02	-	1.60E-02	-	5.80E-03	-
100	1.02E-02	2.99	2.00E-03	3.00	7.78E-04	2.90
200	1.61E-03	2.66	3.01E-04	2.74	9.81E-05	2.99
400	3.47E-04	2.21	5.97E-05	2.33	1.35E-05	2.86

Example 4.15. (The 0D2V Leonard-Bernstein (linearized) Fokker-Planck equation)

$$f_t - \frac{1}{\epsilon}((v_x - \bar{v}_x)f)_{v_x} - \frac{1}{\epsilon}((v_y - \bar{v}_y)f)_{v_y} = \frac{1}{\epsilon}D(f_{v_x v_x} + f_{v_y v_y}), \quad v_x, v_y \in [-2\pi, 2\pi] \quad (4.19)$$

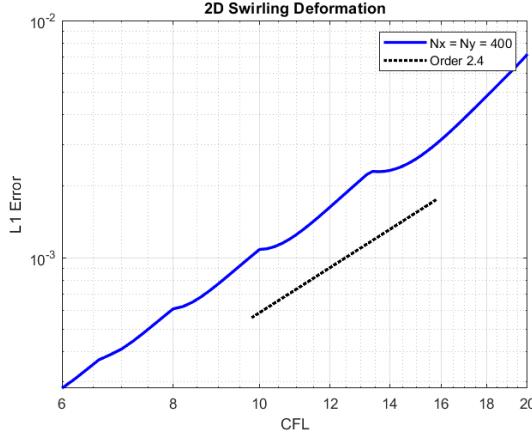


Figure 16: IMEX(2,3,3), $\epsilon = 1$, Final time $T_f = 0.1$.

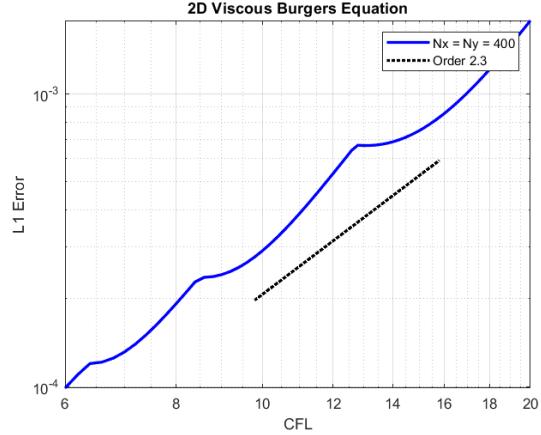


Figure 17: IMEX(2,3,3), $\epsilon = 0.1$, Final time $T_f = 0.2$.

with zero boundary conditions and equilibrium solution the Maxwellian

$$f_M(v_x, v_y) = \frac{n}{2\pi RT} \exp\left(-\frac{(v_x - \bar{v}_x)^2 + (v_y - \bar{v}_y)^2}{2RT}\right), \quad (4.20)$$

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 $\bar{v}_x = \bar{v}_y = 0$. These quantities were chosen for scaling convenience. When testing the spatial and temporal orders of accuracy we set the initial distribution $f(v_x, v_y, t = 0) = f_M(v_x, v_y)$. Table 20 shows the convergence results under spatial mesh refinement. We observe higher fourth-order convergence in space for $CFL = 0.95$. The time-stepping and splitting errors start to dominate the spatial error for larger CFL numbers, as observed for $CFL = 8$. Figure 18 shows the temporal order of convergence is roughly 2.6 for fixed mesh $N_{v_x} = N_{v_y} = 400$ and CFL numbers varying from 6 to 20. We again note the interplay between the third-order time-stepping and second-order splitting.

When testing for relaxation of the system, we choose the initial distribution $f(v_x, v_y, t = 0) = f_{M1}(v_x, v_y) + f_{M2}(v_x, v_y)$, that is, the sum of two randomly generated Maxwellians such that the total macro-parameters are preserved. The number density, bulk velocities, and temperature of each Maxwellian are listed in Table 19. We set $\bar{v}_y = 0$ so that the two generated Maxwellians are shifted only along the v_x axis.

The macro-parameters we want to conserve are number density, bulk velocity, and temperature, which in two dimensions are respectively given by

$$n = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\mathbf{v}) dv_y dv_x, \quad (4.21a)$$

$$\bar{\mathbf{v}} = \frac{1}{n} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{v} f(\mathbf{v}) dv_y dv_x, \quad (4.21b)$$

	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 19: $n = \pi$, $\bar{\mathbf{v}} = \mathbf{0}$, and $T = 3$.

$$T = \frac{1}{2Rn} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (\mathbf{v} - \bar{\mathbf{v}})^2 f(\mathbf{v}) dv_y dv_x. \quad (4.21c)$$

Figures 25(f) and 26 show the solution using fixed mesh $N_{v_x} = N_{v_y} = 200$ and $CFL = 6$. Although we computed the solution up to time $T_f = 20$, there was no difference (to the naked eye) after time $t = 3$. Although the solution appears to reach equilibrium, we again note that the proposed EL-RK-FV algorithm is not equilibrium-preserving. Figure 25(a) verifies mass conservation, but Figure 25(b) implies that the numerical solution has some negative values and is not positivity-preserving. Referring to Figure 25, momentum and energy are not conserved. As seen in Figure 25(d), the bulk velocity in the v_y -direction is on the order of machine epsilon because we constructed the two Maxwellians in Table 19 such that $\bar{v}_{M1,y} = \bar{v}_{M2,y} = 0$. Hence, there is no drift velocity in the v_y -direction.

Table 20: Convergence study with spatial mesh refinement for equation (4.19) with IMEX(2,3,3) and Strang splitting at $T_f = 0.5$.

$CFL = 0.95$						
$N_{v_x} = N_{v_y}$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	9.07E-04	-	4.22E-04	-	5.49E-04	-
100	7.19E-05	3.66	3.15E-05	3.74	4.36E-05	3.66
200	5.35E-06	3.75	2.15E-06	3.87	2.93E-06	3.89
400	3.54E-07	3.92	1.37E-07	3.98	1.82E-07	4.01
$CFL = 8$						
$N_{v_x} = N_{v_y}$	L^1 Error	Order	L^2 Error	Order	L^∞ Error	Order
50	5.70E-03	-	1.84E-03	-	1.26E-03	-
100	1.08E-03	2.40	3.53E-04	2.39	2.82E-04	2.16
200	1.69E-04	2.67	5.68E-05	2.64	5.02E-05	2.49
400	2.73E-05	2.63	9.30E-06	2.61	8.63E-06	2.54

5 Conclusion

In this paper, we proposed a new EL-RK-FV method for solving convection and convection-diffusion equations. Whereas SL methods require solving for the exact characteristics, which

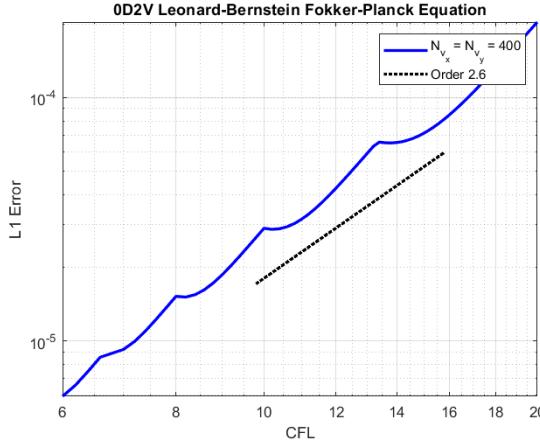


Figure 18: IMEX(2,3,3), Final time $T_f = 0.1$.

is often highly nontrivial for nonlinear problems, our EL method computes linear space-time curves as the approximate characteristics. WENO-AO schemes allowed us to perform spatial reconstruction at arbitrary points which was essential since the traceback grid was not necessarily the (uniform) background grid. By working with the time-differential form, we could use a method-of-lines approach. Explicit RK methods were used for pure convection problems, and IMEX RK methods were used for convection-diffusion equations. Dimensional splitting was used for higher-dimensional problems. Several one- and two-dimensional test problems demonstrated the algorithm's robustness, high-order accuracy, and ability to allow extra large time steps. Ongoing and future work includes modifying the algorithm to handle shocks and rarefaction waves (the authors already have promising results that will be written in another paper), and developing a non-splitting version of the EL-RK-FV algorithm.

Acknowledgements

Research is supported by NSF grant NSF-DMS-1818924 and NSF-DMS-2111253, Air Force Office of Scientific Research FA9550-18-1-0257 and University of Delaware. The authors would like to thank William Taitano (Air Force Research Laboratory) and Alexander Alekseenko (California State University at Northridge) for their help in constructing the Leonard-Bernstein Fokker-Planck equation test problems. Further thanks goes to Robert Martin (Army Research Laboratory) and Alexander Alekseenko for their mentorship during Joseph Nakao's summer internship at the Air Force Research Laboratory.

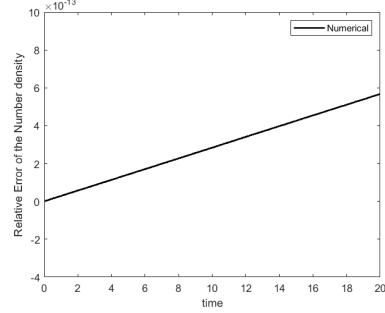


Figure 19: *

(a)

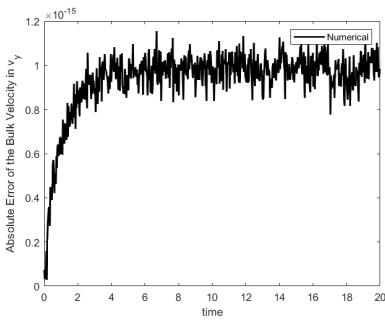


Figure 19: *

(b)

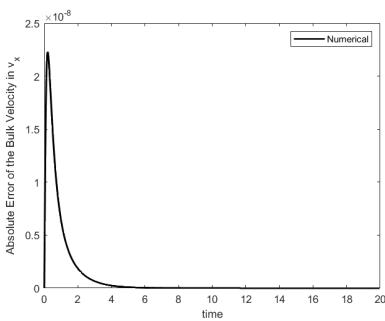


Figure 20: *

(d)

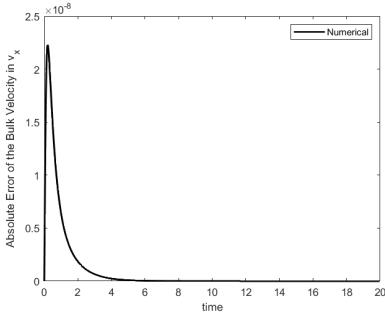


Figure 20: *

(e)

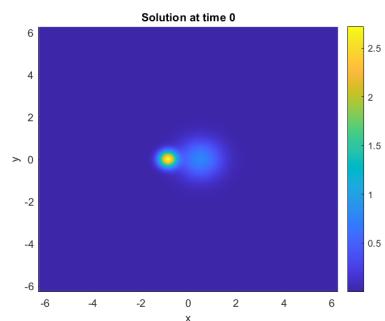


Figure 23: *

(c)

Figure 24: *

(f)

Figure 25: Figures (a)-(e): Relative macro-parameters for equation (4.19) with initial distribution of two Maxwellians defined by Table 19. Mesh $N_{v_x} = N_{v_y} = 200$, $CFL = 6$. Figure (f): The initial distribution.

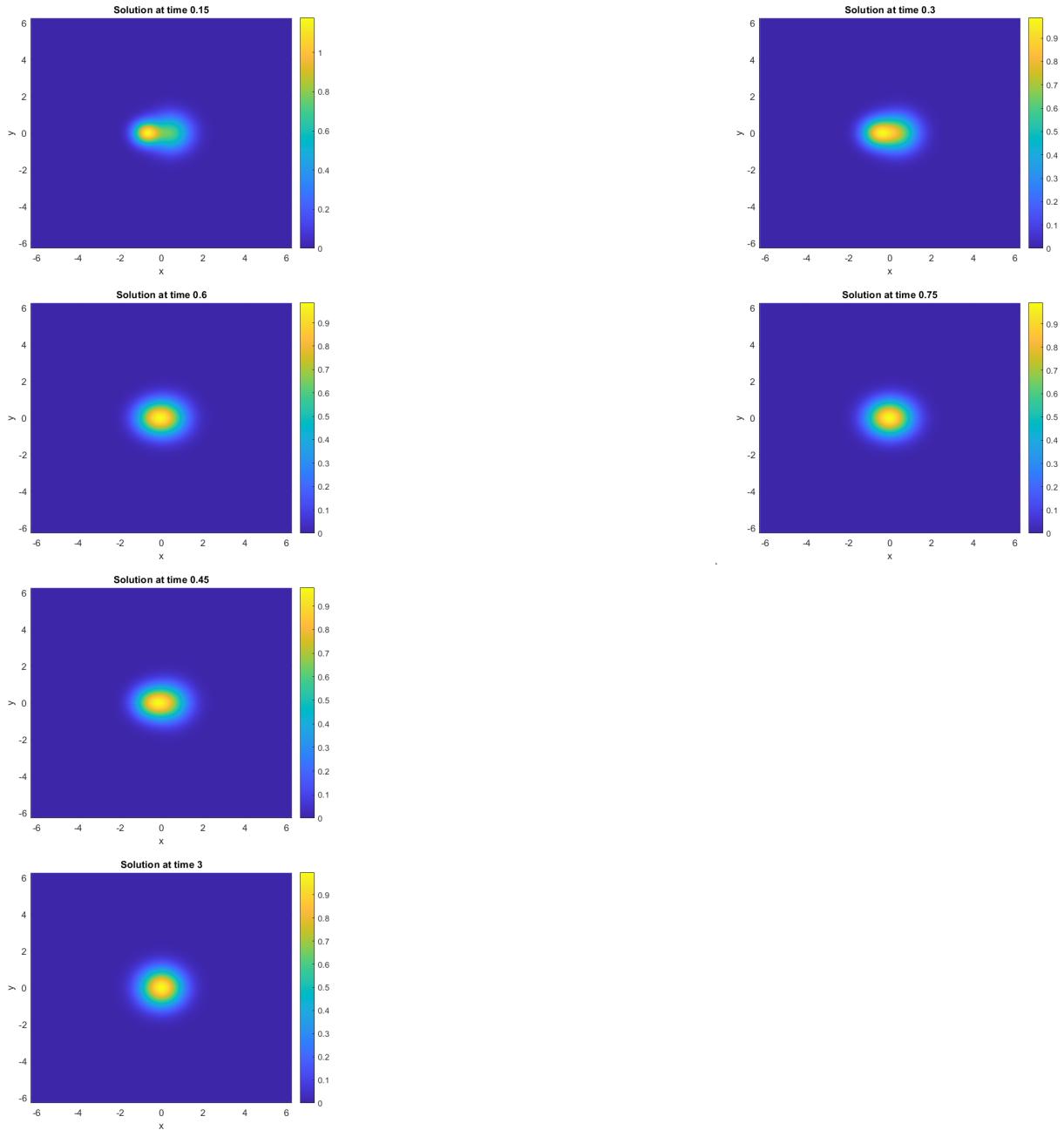


Figure 26: Various snapshots of the numerical solution to equation (4.19) with initial distribution of two Maxwellians defined by Table 19. Mesh $N_{v_x} = N_{v_y} = 200$, $CFL = 6$. Times: 0.15, 0.30, 0.45, 0.60, 0.75, 3.

Appendices

Appendix A. Butcher Tables for Explicit RK Methods

Table 21: *				Table 22: *			
SSP RK3	0	0	0	0	0	0	0
	1	1	0	1/2	1/2	0	0
	1/2	1/4	1/4	0	0	1/2	0
				1	0	0	1
				1/6	1/6	2/3	1/6
				1/6	1/3	1/3	1/6

Appendix B. Fourth-order operator splitting

Define two constants

$$\gamma_1 = \frac{1}{2 - 2^{1/3}} \approx 1.351207191959658 \quad \text{and} \quad \gamma_2 = \frac{-2^{1/3}}{2 - 2^{1/3}} \approx -1.702414383919315.$$

Given constants γ_1 and γ_2 , the fourth-order splitting method in [23, 50] has seven stages, compared to the three stages required for Strang splitting.

Step 1 (x-direction). Solve equation (2.20a) over a time step $\gamma_1 \Delta t/2$.

Step 2 (y-direction). Solve equation (2.20b) over a time step $\gamma_1 \Delta t$.

Step 3 (x-direction). Solve equation (2.20a) over a time step $(\gamma_1 + \gamma_2) \Delta t/2$.

Step 4 (y-direction). Solve equation (2.20b) over a time step $\gamma_2 \Delta t$.

Step 5 (x-direction). Solve equation (2.20a) over a time step $(\gamma_1 + \gamma_2) \Delta t/2$.

Step 6 (y-direction). Solve equation (2.20b) over a time step $\gamma_1 \Delta t$.

Step 7 (x-direction). Solve equation (2.20a) over a time step $\gamma_1 \Delta t/2$.

Note that steps 2 and 6 require steps larger than Δt , and steps 3, 4, and 5 require steps backwards in time.

Appendix C. Butcher Tables for IMEX RK Methods

All IMEX RK schemes included in this appendix are taken from [2]. By construction, each IMEX RK scheme has slightly different properties that are better suited from different problems. Some schemes might have better damping properties and stability regions, be stiffly accurate, etc. In this appendix we opt to pad the implicit Butcher tables with zeros.

Table 23: *				Table 24: *			
IMEX(1,1,1) – Implicit Table	0	0	0	0	0	0	0
	1	0	1	1	1	0	0
		0	1		1	0	

Table 25: *

IMEX(1,2,2) – Implicit Table

0	0	0
1/2	0	1/2
	0	1

Table 26: *

IMEX(1,2,2) – Explicit Table

0	0	0
1/2	1/2	0
	0	1

Table 27: *

IMEX(2,2,2) – Implicit Table

0	0	0	0
γ	0	γ	0
1	0	$1 - \gamma$	γ
	0	$1 - \gamma$	γ

Table 28: *

IMEX(2,2,2) – Explicit Table

0	0	0	0
γ	γ	0	0
1	δ	$1 - \delta$	0
	δ	$1 - \delta$	0

Table 29: *

Let $\gamma = 1 - \sqrt{2}/2$ and $\delta = 1 - 1/(2\gamma)$.

Table 30: *

IMEX(2,3,3) – Implicit Table

0	0	0	0
γ	0	γ	0
$1 - \gamma$	0	$1 - 2\gamma$	γ
	0	$1/2$	$1/2$

Table 31: *

IMEX(2,3,3) – Explicit Table

0	0	0	0
γ	γ	0	0
$1 - \gamma$	$\gamma - 1$	$2(1 - \gamma)$	0
	0	$1/2$	$1/2$

Table 32: *

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

Table 33: *

IMEX(2,3,2) – Implicit Table

0	0	0	0
γ	0	γ	0
1	0	$1 - \gamma$	γ
	0	$1 - \gamma$	γ

Table 34: *

IMEX(2,3,2) – Explicit Table

0	0	0	0
γ	γ	0	0
1	δ	$1 - \delta$	0
	0	$1 - \gamma$	γ

Table 35: *

Let $\gamma = (2 - \sqrt{2})/2$ and $\delta = -2\sqrt{2}/3$.

Table 36: *

IMEX(3,4,3) – Implicit Table					
0	0	0	0	0	
γ	0	γ	0	0	
0.717933	0	0.282067	γ	0	
1	0	1.208497	-0.644363	γ	
	0	1.208497	-0.644363	γ	

Table 37: *

IMEX(3,4,3) – Explicit Table					
0	0	0	0	0	0
γ	γ	0	0	0	0
0.717933	0.321279	0.396654	0	0	0
1	-0.105858	0.552929	0.552929	0	0
	0	1.208497	-0.644363	γ	

Table 38: *

Let $\gamma = 0.435867$.

Table 39: *

IMEX(4,4,3) – Implicit Table					
0	0	0	0	0	0
$1/2$	0	$1/2$	0	0	0
$2/3$	0	$1/6$	$1/2$	0	0
$1/2$	0	$-1/2$	$1/2$	$1/2$	0
1	0	$3/2$	$-3/2$	$1/2$	$1/2$
	0	$3/2$	$-3/2$	$1/2$	$1/2$

Table 40: *

IMEX(4,4,3) – Explicit Table					
0	0	0	0	0	0
$1/2$	$1/2$	0	0	0	0
$2/3$	$11/18$	$1/18$	0	0	0
$1/2$	$5/6$	$-5/6$	$1/2$	0	0
1	$1/4$	$7/4$	$3/4$	$-7/4$	0
	$1/4$	$7/4$	$3/4$	$-7/4$	0

Appendix D. An illustrative example with IMEX(2,2,2)

In this section, we couple the EL-RK-FV algorithm with IMEX(2,2,2), that is, two-stage implicit, two-stage explicit, and of combined order two. This scheme is L-stable and uses a second order DIRK method. The Butcher tables are given in Appendix C. Figure 27 shows the lone sub-space-time region ${}_1\Omega_j$.

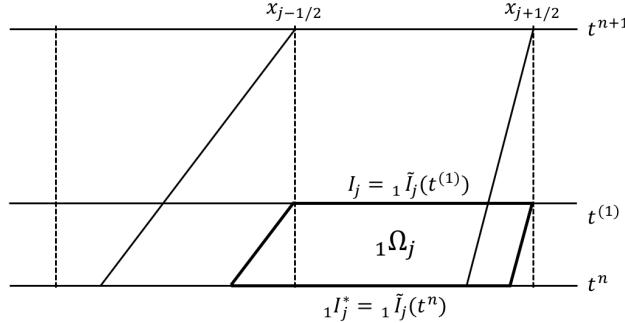


Figure 27: The space-time region ${}_1\Omega_j$ for IMEX(2,2,2).

Step 0a. Compute the approximate characteristic speeds using equation (2.2). After defining the space-time region Ω_j , compute the possibly nonuniform traceback cell averages $\tilde{u}_j(t^n)$ using Algorithm 1.

Step 0b. Use the possibly nonuniform traceback cell averages $\tilde{u}_j(t^n)$ in Algorithm 2 to compute $\hat{K}_1 = \mathcal{F}(U^n; t^n)$.

Step 1a. Using the same approximate characteristic speeds from step 1a, define the sub-space-time region ${}_1\Omega_j$ as seen in Figure 27. Compute the possibly nonuniform traceback cell averages ${}_1\tilde{u}_j^n$ using Algorithm 1.

Step 1b. Use the possibly nonuniform traceback cell averages ${}_1\tilde{u}_j^n$ in Algorithm 2 to compute ${}_1\hat{K}_1 = \mathcal{F}({}_1U^n; t^n)$.

Step 1c. Recalling equation (3.4), solve equation (3.10) by solving the linear system

$$\left(\mathbf{I} - \frac{\gamma\epsilon\Delta t}{\Delta x^2} \mathbf{D}_4 \right) {}_1\vec{U}^{(1)} = {}_1\vec{U}^n + \gamma\Delta t {}_1\hat{K}_1 + \gamma\Delta t \vec{g}(x, t^{(1)}), \quad (5.1)$$

where $\vec{g}_j(x, t^{(1)}) = \int_{I_j} g(x, t^{(1)}) dx$ can be computed with a Gaussian quadrature.

Step 1d. Compute the uniform cell averages $\bar{u}_j^{(1)} = {}_1U_j^{(1)} / \Delta x$.

Step 1e. Compute the uniform cell averages $\bar{u}_{xx,j}^{(1)}$ using equation (3.4),

$$\bar{u}_{xx}^{(1)} = \frac{1}{\Delta x^2} \mathbf{D}_4 \bar{u}^{(1)}. \quad (5.2)$$

Step 1f. Compute the possibly nonuniform traceback cell averages $\tilde{u}_j^{(1)}$ and $\tilde{u}_{xx,j}^{(1)}$ (we are now in the space-time region Ω_j) using Algorithm 1.

Step 1g. Compute $K_1 = \mathcal{G}(U^{(1)}; t^{(1)})$,

$$K_1 = \epsilon\Delta x_j^{(1)} \tilde{u}_{xx,j}^{(1)} + \int_{\tilde{I}_j(t^{(1)})} g(x, t^{(1)}) dx, \quad (5.3)$$

where the definite integral involving $g(x, t)$ can be evaluated using a Gaussian quadrature.

Step 1h. Use the possibly nonuniform traceback cell averages $\tilde{u}_j^{(1)}$ in Algorithm 2 to compute $\hat{K}_2 = \mathcal{F}(U^{(1)}; t^{(1)})$.

Step 2. Recalling equation (3.4), solve equation (3.7a) by solving the linear system

$$\left(\mathbf{I} - \frac{\gamma\epsilon\Delta t}{\Delta x^2} \mathbf{D}_4 \right) \vec{U}^{n+1} = \vec{U}^n + (1 - \gamma)\Delta t \vec{K}_1 + \Delta t(\delta \vec{K}_1 + (1 - \delta) \vec{K}_2) + \gamma \Delta t \vec{g}(x, t^{n+1}), \quad (5.4)$$

where $\vec{g}_j(x, t^{n+1}) = \int_{I_j} g(x, t^{n+1}) dx$ can be computed with a Gaussian quadrature.

References

- [1] E. Abreu, W. Lambert, J. Perez, and A. Santo, A new finite volume approach for transport models and related applications with balancing source terms, *Math. and Comput. in Simul.*, **137** (2017), pp. 2-28.
- [2] U.M. Ascher, S.J. Ruuth, and R.J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations, *App. Numer. Math.*, **25:2-3** (1997), pp. 151-167.
- [3] T. Arbogast, C.-S. Huang, and X. Zhao, Finite volume WENO schemes for nonlinear parabolic problems with degenerate diffusion on non-uniform meshes, *J. Comput. Phys.*, **399** (2019), pp. 108921.
- [4] T. Arbogast, C.-S. Huang, X. Zhao, and D.N. King, A third order, implicit, finite volume, adaptive Runge–Kutta WENO scheme for advection–diffusion equations, *Comp. Meth. in Appl. Mech. and Eng.*, **368** (2020), pp. 113155.
- [5] D.S. Balsara, S. Garain, and C.-W. Shu, An efficient class of WENO schemes with adaptive order, *J. Comput. Phys.*, **326** (2016), pp. 780-804.
- [6] D.S. Balsara, S. Garain, V. Florinski, and W. Boscheri, An efficient class of WENO schemes with adaptive order for unstructured meshes, *J. Comput. Phys.*, **404** (2020), pp. 109062.
- [7] F. Benkhaldoun, S. Sari, and M. Seaid, A family of finite volume Eulerian-Lagrangian methods for two-dimensional conservation laws, *J. Comput. and App. Math.*, **285** (2015), pp. 181-205.
- [8] W. Boscheri and M. Dumbser, A direct Arbitrary-Lagrangian–Eulerian ADER-WENO finite volume scheme on unstructured tetrahedral meshes for conservative and non-conservative hyperbolic systems in 3D, *J. Comput. Phys.*, **275** (2014), pp. 484-523.
- [9] W. Boscheri and M. Dumbser, Arbitrary-Lagrangian-Eulerian One-Step WENO Finite Volume Schemes on Unstructured Triangular Meshes, *Comm. in Comput. Phys.*, **14:5** (2013), pp. 1174-1206.

[10] W. Boscheri, R. Loubere, and M. Dumbser, Direct Arbitrary-Lagrangian–Eulerian ADER-MOOD finite volume schemes for multidimensional hyperbolic conservation laws, *J. Comput. Phys.*, **292** (2015), pp. 56-87.

[11] W. Boscheri, High Order Direct Arbitrary-Lagrangian-Eulerian (ALE) Finite Volume Schemes for Hyperbolic Systems on Unstructured Meshes, *Arch. Comput. Meth. Eng.*, **24** (2017), pp. 751-801.

[12] X. Cai, W. Guo, and J.-M. Qiu, A high order conservative semi-Lagrangian discontinuous Galerkin method for two-dimensional transport simulations, *J. Sci. Comput.*, **73** (2017) pp. 514-542.

[13] J.A. Carrillo and F. Vecil, Nonoscillatory interpolation methods applied to Vlasov-Based models, *SIAM J. Sci. Comput.*, **29** (2007), pp. 1179-1206.

[14] M.A. Celia, T.F. Russell, I. Herrera, and R.E. Ewing, An Eulerian-Lagrangian localized adjoint method for the advection-diffusion equation, *Adv. Water Resour.*, **13** (1990), pp. 187-206.

[15] J. Chen, X. Cai, J. Qiu, and J.-M. Qiu, Adaptive Order WENO Reconstructions for the Semi-Lagrangian Finite Difference Scheme for Advection Problem, *Comm. in Comput. Phys.*, **30:1** (2021), pp. 67-96.

[16] J. Chen, J. Nakao, and J.-M. Qiu, High-order Eulerian-Lagrangian Runge-Kutta finite volume (EL-RK-FV) methods for nonlinear hyperbolic problems with shocks, *In preparation.*

[17] B. Cockburn, C. Johnson, C.-W. Shu, and E. Tadmor, Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, A. Quarteroni (ed.), *Lecture Notes in Math 1697*, Springer, New York, 1997.

[18] S. Conde, S. Gottlieb, Z.J. Grant, and J.N. Shadid, Implicit and Implicit-Explicit Strong Stability Preserving Runge-Kutta Methods with High Linear Order, *J. Sci. Comput.*, **73** (2017), pp. 667-690.

[19] N. Crouseilles, M. Mehrenberger, and E. Sonnendrucker, Conservative semi-Lagrangian schemes for Vlasov equations, *J. Comput. Phys.*, **229:6** (2010), pp. 1927-1953.

[20] M.-C. Ding, X. Cai, W. Guo, and J.-M. Qiu, A semi-Lagrangian discontinuous Galerkin (DG)-local DG method for solving convection-diffusion equations, *J. Comput. Phys.*, **409** (2020), pp. 109295.

[21] J. Donea, A. Huerta, J.-Ph. Ponthot, and A. Rodriguez-Ferran, Arbitrary Lagrangian-Eulerian Methods, Chapter 14 in *The Encyclopedia of Computational Mechanics, Volume 1*, Wiley (2004), pp. 413-437.

[22] F. Filbet, E. Sonnendrucker, and P. Bertrand, Conservative numerical schemes for the Vlasov equation, *J. Comput. Phys.*, **172:1** (2001), pp. 166-187.

- [23] E. Forest and R.D. Ruth, Fourth-order symplectic integration, *Phys. D Nonlinear Phenom.*, **43** (1990), pp. 105-117.
- [24] S. Gottlieb, C.-W. Shu, and E. Tadmor, Strong stability-preserving high-order time discretization methods, *SIAM Review*, **43:1** (2001), pp. 89-112.
- [25] I. Higueras, N. Happenhofer, O. Koch, and F. Kupka, Optimized strong stability preserving IMEX Runge-Kutta methods, *J. Comput. and Appl. Math.*, **272** (2014), pp. 116-140.
- [26] C. Hirt, A. Amsden, and J. Cook, An arbitrary lagrangian eulerian computing method for all flow speeds, *J. Comput. Phys.*, **14** (1974), pp. 227-253.
- [27] C.-S. Huang, T. Arbogast, and J. Qiu, An Eulerian-Lagrangian WENO finite volume scheme for advection problems, *J. Comput. Phys.*, **231** (2012), pp. 4028-4052.
- [28] C.-S. Huang and T. Arbogast, An Eulerian-Lagrangian Weighted Essentially Nonoscillatory scheme for Nonlinear Conservation Laws, *Num. Meth. for Part. Diff. Eq.*, **33:3** (2017), pp. 651-680.
- [29] F. Huot, A. Ghizzo, P. Bertrand, E. Sonnendrucker, and O. Coulard, Instability of the time splitting scheme for the one-dimensional and relativistic Vlasov-Maxwell system, *J. Comput. Phys.*, **185:2** (2003), pp. 512-531.
- [30] R.J. Leveque, High-resolution conservative algorithms for advection in incompressible flow, *SIAM J. Numer. Anal.*, **33:2** (1996), pp. 627-665.
- [31] D. Levy, G. Puppo, and G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, *ESAIM: Math. Model. and Numer. Anal.*, **33:3** (1999), pp. 547-571.
- [32] L. Li, J. Qiu, and G. Russo, A High-Order Semi-Lagrangian Finite Difference Method for Nonlinear Vlasov and BGK Models. *Comm. on Applied Math. and Comput.*, (2022), pp. 1-29.
- [33] R. Li, T. Tang, and P.W. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, *J. Comput. Phys.*, **170** (2001), pp. 562-588.
- [34] S. Li and L. Petzold, Moving mesh methods with upwinding schemes for time-dependent PDEs, *J. Comput. Phys.*, **131** (1997), pp. 368-377.
- [35] D. Luo, W. Huang, and J. Qiu, A quasi-Lagrangian moving mesh discontinuous Galerkin method for hyperbolic conservation laws, *J. Comput. Phys.*, **396** (2019), pp. 544-578.
- [36] J.S. Peery and D.E. Carroll, Multi-material ale methods in unstructured grids, *Comp. Meth. in App. Mech. and Eng.*, **187** (2000), pp. 591-619.
- [37] J.-M. Qiu and A. Christlieb, A conservative high order semi-Lagrangian WENO method for the Vlasov equation, *J. Comput. Phys.*, **229:4** (2010), pp. 1130-1149.

[38] J.-M. Qiu and C.-W. Shu, Conservative high order semi-Lagrangian finite difference WENO methods for advection in incompressible flow, *J. Comput. Phys.*, **230** (2011), pp. 863-889.

[39] J.-M. Qiu and C.-W. Shu, Positivity preserving semi-Lagrangian discontinuous Galerkin formulation: Theoretical analysis and application to the Vlasov-Poisson system, *J. Comput. Phys.*, **230** (2011), pp. 8386-8409.

[40] J.A. Rossmanith and D.C. Seal, A positivity-preserving high-order semi-Lagrangian discontinuous Galerkin scheme for the Vlasov-Poisson equations, *J. Comput. Phys.*, **230** (2011), pp. 6203-6232.

[41] T.F. Russell and M.A. Celia, An overview of research on Eulerian-Lagrangian localized adjoint methods (ELLAM), *Adv. Water Resour.* **25** (2002), pp. 1215-1231.

[42] M. Seydaoglu, U. Erdogan, and T. Ozis, Numerical solution of Burgers' equation with high order splitting methods, *J. Comput. and Appl. Math.*, **291** (2016), pp. 410-421.

[43] C.-W. Shu, High Order Weighted Essentially Nonoscillatory Schemes for Convection Dominated Problems, *SIAM Review*, **51:1** (2009), pp. 82-126.

[44] J.M. Stockie, J.A. Mackenzie, and R.D. Russell, A moving mesh method for one-dimensional hyperbolic conservation laws, *SIAM J. Sci. Comput.*, **22** (2001), pp. 1791-1813.

[45] H. Tang and T. Tang, Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws, *SIAM J. Numer. Anal.*, **41:2** (2003), pp. 487-515.

[46] H. Wang, S. Wang, Q. Zhang, and C.-W. Shu, Local discontinuous Galerkin methods with implicit-explicit time-marching for multi-dimensional convection-diffusion problems, *ESAIM: Math. Model. and Numer. Anal.*, **50:4** (2016), pp. 1083-1105.

[47] T. Xiong, G. Russo, and J.-M. Qiu, Conservative Multi-dimensional Semi-Lagrangian Finite Difference Scheme: Stability and Applications to the Kinetic and Fluid Simulations, *J. Sci. Comp.*, **79** (2019), pp. 1241-1270.

[48] D. Xiu and G.E. Karniadakis, A semi-Lagrangian high-order method for Navier-Stokes equations, *J. Comput. Phys.*, **172** (2001) pp. 658-684.

[49] Y. Yang, J. Chen, and J.-M. Qiu, Stability analysis of the Eulerian-Lagrangian finite volume method for nonlinear hyperbolic equations in one spatial dimension, *In preparation.*

[50] H. Yoshida, Construction of higher order symplectic integrators, *Phys. Lett. A*, **150** (1990), pp. 262-268.

[51] M. Zennaro, Natural continuous extensions of Runge-Kutta methods, *Math Comp.*, **46** (1986), pp. 119-133.

- [52] J. Zhu and J. Qiu, A new third order finite volume weighted essentially non-oscillatory scheme on tetrahedral meshes, *J. Comput. Phys.*, **349** (2017), pp. 220-232.
- [53] J. Zhu and J. Qiu, A New Type of Finite Volume WENO Schemes for Hyperbolic Conservation Laws, *J. Sci. Comput.*, **73** (2017), pp. 1338-1359.
- [54] J. Zhu and C.-W. Shu, A new type of multi-resolution WENO schemes with increasingly higher order of accuracy on triangular meshes, *J. Comput. Phys.*, **392** (2019), pp. 19-33.
- [55] J. Zhu and J. Qiu, New finite volume weighted essentially nonoscillatory schemes on triangular meshes, *SIAM J. Sci. Comput.*, **40:2** (2018), pp. A903-A928.