energies

MDPI

Article

A Hybrid Time Integration Scheme for the Discontinuous Galerkin Discretizations of Convection-Dominated Problems

Liang Li * and Songping Wu

School of Aeronautic Science and Engineering, Beihang University, Beijing 100191, China; wusping825@163.com
* Correspondence: dlamliliang@buaa.edu.cn

Received: 27 February 2020; Accepted: 5 April 2020; Published: 11 April 2020



Abstract: Discontinuous Galerkin (DG) method is a popular high-order accurate method for solving unsteady convection-dominated problems. After spatially discretizing the problem with the DG method, a time integration scheme is necessary for evolving the result. Owing to the stability-based restriction, the time step for an explicit scheme is limited by the smallest element size within the mesh, making the calculation inefficient. In this paper, a hybrid scheme comprising a three-stage, third-order accurate, and strong stability preserving Runge–Kutta (SSP-RK3) scheme and the three-stage, third-order accurate, L-stable, and diagonally implicit Runge–Kutta (LDIRK3) scheme is proposed. By dealing with the coarse and the refined elements with the explicit and implicit schemes, respectively, the time step for the hybrid scheme is free from the limitation of the smallest element size, making the simulation much more efficient. Numerical tests and comparison studies were made to show the performance of the hybrid scheme.

Keywords: unsteady convection-dominated problem; time integration; discontinuous Galerkin; explicit scheme; implicit scheme

1. Introduction

The convection-dominated problem plays a key role in fluid dynamics, mass and heat transfer, and many other fields. The discontinuous Galerkin (DG) method has become popular for numerically solving convection-dominated problems mainly due to the attractive features of the method. Firstly, a polynomial is used to locally approximate the solution to the governing equation, allowing the method to easily achieve high-order accuracy on unstructured meshes. Secondly, within the framework of the DG method, an element only communicates with its nearest face neighbors. The compact stencil greatly benefits parallel computation. Thirdly, upwinding is introduced in the DG method using numerical flux. The agreement between the numerical method and the convective nature of the governing equation helps to stabilize the calculation.

In 1973, the DG method was first proposed by Reed and Hill in [1]. During the 1980s and the 1990s, the method was fully developed by Cockburn, Shu, Lin, and Hou in a series of papers [2–6]. In those papers, a method-of-lines approach was used for solving the time-dependent conservation laws. That is, the DG method was used to spatially discretize the governing equation; then, a time integration scheme was used to evolve the result in the time domain. This was followed by many other studies. For a comprehensive review of the approach, the reader is referred to [7] (pp. 97–108) and [8] (pp. 171–188). The former discussed cases where the time integration scheme is explicit; the latter concentrated on the cases where the scheme is implicit.

Both the explicit and the implicit schemes have their pros and cons. In realistic applications, the explicit scheme is easy to implement and the computing cost per time step is small. However, the time

step length for the explicit scheme is strongly limited by the stability-based Courant–Friedrichs–Lewy (CFL) condition. Although the CFL restriction for the implicit scheme is rather loose, solving nonlinear equations is inevitable for the implicit scheme, making the coding complex and the computing per time step expensive. Therefore, the explicit scheme has prevailed in the research on unsteady problems, whereas the implicit scheme is applied in solving steady problems.

When the mesh for an unsteady problem contains tiny elements, the choice of the time integration scheme was unclear. On one hand, the available time step for an explicit scheme is limited by the smallest element size. Too small an element leads to too many time steps during the calculation. On the other hand, the accuracy with respect to time matters in solving unsteady problems, forbidding the usage of a too big time step. Without a large time step, the implicit scheme is inefficient.

One natural idea to deal with the issue is blending the two schemes. An intuitive research can be found in [9]. For simulating compressible flows, the authors used the additive Runge–Kutta (ARK) scheme [10] to separately deal with the refined and coarse regions of the mesh. In [11], after spatially discretizing the Maxwell's equations with the DG method, a combination of the explicit leap-frog scheme and the implicit Crank–Nicolson scheme was used to integrate the result. The region-based hybrid scheme was named "locally implicit scheme" by the authors. In [12], a scheme comprising the Verlet scheme and the implicit Crank–Nicolson scheme was chosen to integrate the DG discretization. An error analysis for the hybrid scheme was conducted within the paper. In [13], a locally implicit DG method was used to solve the Maxwell's equations. The time integration scheme used therein comprised of the implicit Crank–Nicolson scheme and the explicit leap-frog scheme.

In this paper, we develop a new hybrid time integration scheme comprising two high-order schemes for the evolution of the DG discretization. The explicit scheme is the three-stage, third-order accurate, and strong stability preserving Runge–Kutta (SSP-RK3) scheme [14,15] and the implicit one is the three-stage, third-order accurate, L-stable, and diagonally implicit Runge–Kutta (LDIRK3) scheme [16–18]. For the purpose of stabilizing the calculation and minimizing the computing consumption, we propose an element grouping strategy corresponding to the hybrid scheme.

The remainder of the paper is organized as follows: In Section 2, we briefly introduce the DG discretization using a model problem. In Section 3, the hybrid time integration scheme, the element grouping strategy, and the nonlinear-solver-related techniques are introduced. Numerical tests are conducted in Section 4 to illustrate the effectiveness of the hybrid scheme. Concluding remarks are provided in Section 5.

2. DG Discretization

In this section, we introduce DG discretization using a two-dimensional (2D) convection-diffusion problem:

$$\begin{cases} \frac{\partial u}{\partial t} + \nabla \cdot \mathbf{f} = \varepsilon \Delta u, & \mathbf{x} \in \Omega, \quad t \geq 0, \\ u(0, \mathbf{x}) = u_0(\mathbf{x}), & \\ u(t, \mathbf{x}) = b(t, \mathbf{x}), & \mathbf{x} \in \partial\Omega, \quad t \geq 0. \end{cases} \tag{1}$$

where $u = u(t, \mathbf{x})$, $\mathbf{f} = (f_1(u), f_2(u))$ is the convective flux, $u_0(\mathbf{x})$ defines the initial condition, $b(t,\mathbf{x})$ defines the boundary conditions, and $\Omega \subset \mathbb{R}^2$ is a bounded domain with a Lipschitz boundary $\partial\Omega$. For a deeper understanding for the DG discretization, the reader is referred to, e.g., Chapter 4 in [8].

We start by introducing notations. After partitioning Ω into K non-overlapping elements, the gathering of the elements is denoted as $\mathcal{T}_h = \{\Omega_k\}$. For simplicity, we assume the mesh is conforming and shape-regular. We further assume all elements in $\mathcal{T}_h$ are triangular. By $\mathcal{F}_h$, we denote the gathering of element faces in $\mathcal{T}_h$. We say Ω_{k1} "neighbors on" Ω_{k2} when $\partial\Omega_{k1} \cap \partial\Omega_{k2} \in \mathcal{F}_h$. For $\forall\Omega_k \in \mathcal{T}_h$, $\partial\Omega_k$ denotes the boundary of the element, $\mathbf{n}^k$ denotes the unit outward normal vector on $\partial\Omega_k$, and h_k denotes a characteristic size of the element. The solution space for the DG method is defined as:

$$S_h = \left\{ \; w: \quad w \in L^2(\Omega); \quad w|_{\Omega_k} \in P_N(\Omega_k); \quad \forall \Omega_k \in \mathcal{T}_h \; \right\}, \tag{2}$$

where P_N denotes the space of polynomials of degree $\leq N$. We choose the Lagrange polynomials $l_i^k(\mathbf{x})$, $\mathbf{x} \in \Omega_k$, $i = 1, 2, \cdots, N_p$, defined on uniformly distributed nodes as the bases of $P_N(\Omega_k)$. For a triangular element, $N_p = (N+1)(N+2)/2$. In addition, $l_i^k(\mathbf{x})$, $i = 1, 2, \cdots, N_p$, $k = 1, 2, \cdots, K$, form the bases of S_h. On $\partial\Omega_k$, with w^- and w^+ denoting the inner and outer traces of function w, respectively, we denote the jump and mean of w as:

$$[\![w]\!] = w^- - w^+, \quad \langle w \rangle \; = \frac{1}{2}\left(w^- + w^+\right). \tag{3}$$

We approximate the solution of Equation (1) with:

$$u_h = \sum_{k=1}^{K} u_h^k = \sum_{k=1}^{K} \left(\sum_{j=1}^{N_p} u_j^k(t) l_j^k(\mathbf{x}) \right), \tag{4}$$

where u_h^k denotes the local approximation on Ω_k. By multiplying the governing equation with a test function $l_i^k(\mathbf{x})$ and integrating by parts on Ω_k, we obtain:

$$\int_{\Omega_k} \frac{\mathrm{d}u_h^k}{\mathrm{d}t} l_i^k \mathrm{d}\Omega = C_i^k + D_i^k, \tag{5}$$

where C_i^k and D_i^k are the DG discretizations for the convective item and the diffusive item, respectively [19]. The expression for C_i^k is:

$$\int_{\Omega_k} \mathbf{f}(u_h^k) \cdot \nabla l_i^k d\Omega - \int_{\partial\Omega_k} \hat{H} l_i^k dS \tag{6}$$

where $\hat{H}$ is the inviscid numerical flux. A popular type is the local Lax–Friedriches (LLF) flux [7] (p. 170):

$$\hat{H} = \langle \mathbf{f}(u) \rangle \cdot \mathbf{n}^k + \frac{1}{2}\lambda [\![u]\!], \tag{7}$$

where λ is defined as:

$$\lambda = \max_{u \in [\min\{u^-, u^+\}, \max\{u^-, u^+\}]} \left| \mathbf{n}^k \cdot \frac{\partial \mathbf{f}}{\partial u} \right|. \tag{8}$$

If a face of Ω_k coincides with $\partial\Omega$, then the u^+ on the face is defined using the boundary conditions. An introduction of various inviscid numerical fluxes can be found in [20].

Within the frame of the DG method, different approaches are possible for dealing with the diffusive item: the Bassi–Rebay approach [21], the local DG approach [22], the compact DG approach [23], and the direct DG approach [24]. In our work, we apply the non-symmetric interior penalty Galerkin (NIPG) approach [8] (pp. 37–43). Therefore,

$$D_i^k = -\int_{\Omega_k} \varepsilon \nabla u_h^k \cdot \nabla l_i^k d\Omega + \int_{\partial\Omega_k} \varepsilon \left(\langle \nabla u_h \rangle \cdot \mathbf{n}^k l_i^k - \sigma [\![u_h]\!] l_i^k - [\![u_h]\!] \mathbf{n}^k \cdot \frac{1}{2} \nabla l_i^k \right) dS, \tag{9}$$

where the penalty weight σ is defined as:

$$\sigma|_\Gamma = \frac{1}{h_\Gamma}, \quad \Gamma \in \mathcal{F}_h. \tag{10}$$

where h_Γ denotes the length of element face Γ. Similar to the case for the convective item, the outer traces on $\partial\Omega$ are provided by the boundary conditions.

Formulating Equation (5) using all base functions of S_h, we obtain a system of ordinary differential equations (ODEs):

$$\frac{d}{dt}\mathbf{U} = \mathbf{M}^{-1}\mathbf{R} \equiv \mathbf{F}, \tag{11}$$

where:

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}^1 \\ \mathbf{U}^2 \\ \vdots \\ \mathbf{U}^K \end{pmatrix}, \quad \mathbf{M} = \begin{pmatrix} \mathbf{M}^1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{M}^2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{M}^K \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} \mathbf{R}^1 \\ \mathbf{R}^2 \\ \vdots \\ \mathbf{R}^K \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \mathbf{F}^1 \\ \mathbf{F}^2 \\ \vdots \\ \mathbf{F}^K \end{pmatrix}. \tag{12}$$

For $\forall \Omega_k \in \mathcal{T}_h$:

$$\mathbf{U}^k = \begin{pmatrix} u_1^k(t) \\ \vdots \\ u_{N_p}^k(t) \end{pmatrix}, \quad \mathbf{M}^k = \begin{pmatrix} \int_{\Omega_k} l_1^k l_1^k \mathrm{d}\Omega & \cdots & \int_{\Omega_k} l_{N_p}^k l_1^k \mathrm{d}\Omega \\ \vdots & \ddots & \vdots \\ \int_{\Omega_k} l_1^k l_{N_p}^k \mathrm{d}\Omega & \cdots & \int_{\Omega_k} l_{N_p}^k l_{N_p}^k \mathrm{d}\Omega \end{pmatrix}, \tag{13}$$

$\mathbf{R}^k$ denotes the right-hand sides of Equation (5) with i ranging from 1 to N_p and $\mathbf{F}^k = (\mathbf{M}^k)^{-1}\mathbf{R}^k$.

In our applications, the spatial integrations are calculated using the Gauss quadrature rule. Specifically, the accuracy orders for the integrations over an element and an element face are $2N$ and $2N+1$, respectively, with N denoting the degree for the polynomial used for local approximation.

3. Time Integration

3.1. Basic Schemes

Multi-stage Runge–Kutta (RK) schemes are considered in this paper to integrate Equation (11) from time t_n to t_{n+1}. An s-stage RK scheme has the form [25]:

$$\begin{cases} \mathbf{U}^{(i)} = \mathbf{U}_n + \Delta t \sum_{j=1}^{s} a_{ij}\mathbf{F}(t_n + c_j\Delta t, \mathbf{U}^{(j)}), \\ \mathbf{U}_{n+1} = \mathbf{U}_n + \Delta t \sum_{j=1}^{s} b_j\mathbf{F}(t_n + c_j\Delta t, \mathbf{U}^{(j)}), \end{cases} \tag{14}$$

where $\Delta t = t_{n+1} - t_n$, $\mathbf{U}_n = \mathbf{U}(t_n)$ and the coefficients a_{ij}, b_j, and c_j differ with specific RK schemes. The coefficients for SSP-RK3 and LDIRK3 are given by the following Butcher tableaus:

$$\left(\begin{array}{c|c} c_j & a_{ij} \\ \hline & b_j \end{array}\right)_{\mathrm{SSP-RK3}} = \left(\begin{array}{c|ccc} 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1/2 & 1/4 & 1/4 & 0 \\ \hline & 1/6 & 1/6 & 2/3 \end{array}\right), \tag{15}$$

and

$$\left(\begin{array}{c|c} c_j & a_{ij} \\ \hline & b_j \end{array}\right)_{\mathrm{LDIRK3}} = \left(\begin{array}{c|ccc} \alpha & \alpha & 0 & 0 \\ \gamma & \gamma-\alpha & \alpha & 0 \\ 1 & \beta_1 & \beta_2 & \alpha \\ \hline & \beta_1 & \beta_2 & \alpha \end{array}\right), \begin{cases} \alpha = 0.43586652150846, \\ \beta_1 = -(6\alpha^2 - 16\alpha + 1)/4, \\ \beta_2 = (6\alpha^2 - 20\alpha + 5)/4, \\ \gamma = (\alpha + 1)/2. \end{cases} \tag{16}$$

For SSP-RK3, Equation (15) characterizes $a_{ij} = 0, i \leq j$. That is, the calculation for the ith stage only requires the data of previous stages. This makes the explicit scheme easy to implement.

However, its time step length is severely restricted by the CFL condition. When the DG solution for a d-dimensional time-dependent conservation law is sought from S_h, the time step corresponding to SSP-RK3 has to satisfy [3,26]:

$$\Delta t \leq \frac{1}{d}\frac{1}{(2N+1)}\frac{h_k}{\lambda_k}, \quad \forall \Omega_k \in \mathcal{T}_h, \tag{17}$$

where λ_k is the biggest characteristic velocity on Ω_k. A rather small Δt is required when tiny elements exist in $\mathcal{T}_h$, making the simulation less efficient. For LDIRK3, the stability-based restriction on the time step is loose. Its available time step can be hundreds of times bigger than that for an explicit scheme [27]. However, due to the non-zero diagonal coefficients in Equation (16), nonlinear equations have to be solved during the time integration. Taking the first stage of Equation (16) as an example, we need to solve

$$\mathbf{U}^{(1)} = \mathbf{U}_n + \alpha\Delta t \cdot \mathbf{F}(t_n + \alpha\Delta t, \mathbf{U}^{(1)}), \tag{18}$$

to get $\mathbf{U}^{(1)}$. This is usually achieved with the Newton–Raphson method, i.e.,

$$\begin{cases} \mathbf{J}(\mathbf{U}^{(1),m})\Delta\mathbf{U}^{(1),m} = \mathbf{U}_n + \alpha\Delta t \cdot \mathbf{F}(t_n + \alpha\Delta t, \mathbf{U}^{(1),m}) - \mathbf{U}^{(1),m}, \\ \mathbf{U}^{(1),m+1} = \mathbf{U}^{(1),m} + \Delta\mathbf{U}^{(1),m}, \quad m = 0, 1, \cdots \end{cases} \tag{19}$$

where the global Jacobian matrix is defined as:

$$\mathbf{J}(\mathbf{U}^{(1),m}) \equiv \alpha\Delta t \cdot \frac{\partial \mathbf{F}}{\partial \mathbf{U}^{(1),m}} - \mathbf{I}, \tag{20}$$

where $\mathbf{I}$ is the identity matrix. In solving a scalar problem with the DG method, there are $(N_p \cdot K)^2$ values in $\mathbf{J}$. Both the construction of the matrix and the solution of the linear equations, e.g., Equation (19), are computing-expensive and time-consuming. When the nonlinear-solver-related calculation cannot be compensated by a big time step, the implicit scheme is inefficient.

3.2. A New Hybrid Scheme

3.2.1. Elements Grouping and Scheme Coupling

According to their size, the elements in $\mathcal{T}_h$ are gathered into the explicit set $\mathcal{T}_{h,ex}$ and the implicit set $\mathcal{T}_{h,im}$. Generally, all elements in $\mathcal{T}_{h,ex}$ are coarse, whereas all fine elements are collected in $\mathcal{T}_{h,im}$. An index quantifying the element size variance is defined as [9]:

$$S = \frac{\min_{\Omega_k \in \mathcal{T}_{h,ex}} h_k}{\min_{\Omega_k \in \mathcal{T}_{h,im}} h_k}. \tag{21}$$

Similar to [9], the ODEs in Equation (11) are divided into two groups:

$$\begin{cases} \frac{d}{dt}\mathbf{U}_{ex} = \mathbf{F}_{ex}, \\ \frac{d}{dt}\mathbf{U}_{im} = \mathbf{F}_{im}, \end{cases} \tag{22}$$

where $\mathbf{U}_{ex}$ and $\mathbf{U}_{im}$ correspond to $\mathcal{T}_{h,ex}$ and $\mathcal{T}_{h,im}$, respectively.

From t_n to t_{n+1}, the ith stage for an s-stage RK scheme corresponds to $t_n + c_i\Delta t$, where $\Delta t = t_{n+1} - t_n$ and c_i can be found in the left row of the Butcher tableau. That is, $\mathbf{U}^{(i)}$ in Equation (14) can be explained as the numerical solution corresponding to $t_n + c_i\Delta t$. The ARK scheme developed in [10] consists of an explicit RK scheme, denoted by ARK(ex), and an implicit RK scheme, denoted by ARK(im). Their Butcher tableaus characterize that $c_{i,\mathrm{ARK}(ex)} = c_{i,\mathrm{ARK}(im)}$, $i = 1, 2, \cdots, s$. Therefore, the ith stage of ARK(ex) and that of ARK(im) correspond to the same time, which considerably simplifies the coupling between the two schemes. In [9], for the ith stage of ARK, $\mathbf{U}_{ex}^{(i)}$ is firstly

calculated with the ith stage of ARK(ex), then $\mathbf{U}_{im}^{(i)}$ is calculated with the ith stage of ARK(im), using $\mathbf{U}_{ex}^{(i)}$ as boundary conditions.

In Equations (15) and (16), the coefficients $c_i, i = 1, 2, 3$, for SSP-RK3 and LDIRK3 are different. However, the two ends of the time step are universal for the two schemes. Therefore, our hybrid scheme consists of two phases: achieving $\mathbf{U}_{ex}(t_{n+1})$ with SSP-RK3 first, then achieving $\mathbf{U}_{im}(t_{n+1})$ with LDIRK3.

According to Equation (15), the SSP-RK3 scheme can be written as:

$$\begin{cases} \mathbf{U}^{(1)} = \mathbf{U}_n, \\ \mathbf{U}^{(2)} = \mathbf{U}_n + \Delta t \cdot \mathbf{F}(t_n, \mathbf{U}^{(1)}), \\ \mathbf{U}^{(3)} = \mathbf{U}_n + \frac{1}{4}\Delta t \cdot \left[\mathbf{F}(t_n, \mathbf{U}^{(1)}) + \mathbf{F}(t_n + \Delta t, \mathbf{U}^{(2)})\right], \\ \mathbf{U}_{n+1} = \mathbf{U}_n + \Delta t \cdot \left[\frac{1}{6}\mathbf{F}(t_n, \mathbf{U}^{(1)}) + \frac{1}{6}\mathbf{F}(t_n + \Delta t, \mathbf{U}^{(2)}) + \frac{2}{3}\mathbf{F}(t_n + \frac{1}{2}\Delta t, \mathbf{U}^{(3)})\right]. \end{cases} \tag{23}$$

A sketch of a one-dimensional (1D) mesh for the DG method is provided in Figure 1, where the elements around the interface between $\mathcal{T}_{h,ex}$ and $\mathcal{T}_{h,im}$ are marked by δ, θ, ζ and η. In the following descriptions, we let the element marker represent the local approximation on the element and omit the time in $\mathbf{F}$, e.g., δ_{n+1} denotes $\mathbf{U}_{n+1}(\delta)$ and $\mathbf{F}(\delta^{(1)})$ denotes $\mathbf{F}(t_n, \mathbf{U}^{(1)}(\delta))$. The elements in $\mathcal{T}_{h,im}$ that are necessary in computing δ_{n+1} can be identified using Equation (23).

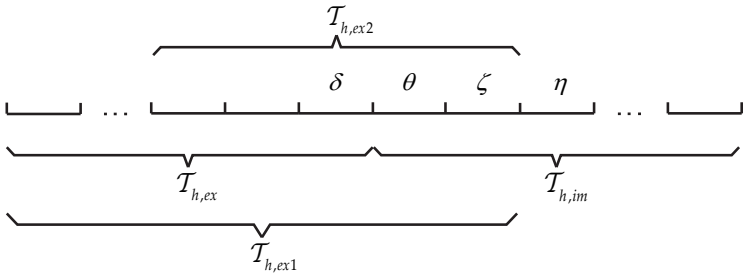


Figure 1. Elements grouping for a one-dimensional (1D) mesh.

Theorem 1. *Within time step $[t_n, t_{n+1}]$, when SSP-RK3 is taken to achieve the local DG solution on a given element, the nearest three layers of elements are involved in the computation.*

Proof. We restrict our discussion to one side of a 1D element for simplicity.

According to Equation (5), when applying the operator $\mathbf{F}$ to the local DG approximation on a given element, the local approximations on the face neighbors of the element are involved. In Figure 1, $\mathbf{F}(\delta^{(1)})$ involves $\theta^{(1)}$, $\mathbf{F}(\theta^{(1)})$ involves $\zeta^{(1)}$, $\mathbf{F}(\zeta^{(1)})$ involves $\eta^{(1)}$, $\mathbf{F}(\delta^{(2)})$ involves $\theta^{(2)}$, $\mathbf{F}(\theta^{(2)})$ involves $\zeta^{(2)}$, and $\mathbf{F}(\delta^{(3)})$ involves $\theta^{(3)}$.

According to the second line in Equation (23), $\mathbf{F}(\delta^{(1)})$ contributes to $\delta^{(2)}$. Therefore, $\delta^{(2)}$ requires $\theta^{(1)}$ because $\mathbf{F}(\delta^{(1)})$ involves $\theta^{(1)}$. Similarly, $\theta^{(2)}$ requires $\zeta^{(1)}$.

According to the third line in Equation (23), $\mathbf{F}(\delta^{(2)})$ contributes to $\delta^{(3)}$. Therefore, $\delta^{(3)}$ requires $\theta^{(2)}$ because $\mathbf{F}(\delta^{(2)})$ involves $\theta^{(2)}$. $\theta^{(2)}$ requiring $\zeta^{(1)}$ is already known. Therefore, $\delta^{(3)}$ requires $\zeta^{(1)}$. Similarly, $\theta^{(3)}$ requires $\eta^{(1)}$.

According to the last line in Equation (23), $\mathbf{F}(\delta^{(3)})$ contributes to δ_{n+1}. Therefore, δ_{n+1} requires $\theta^{(3)}$ because $\mathbf{F}(\delta^{(3)})$ involves $\theta^{(3)}$. $\theta^{(3)}$ requiring $\eta^{(1)}$ is already known. Therefore, δ_{n+1} requires $\eta^{(1)}$. □

SSP-RK3 depends on the solutions of t_n. According to Theorem 1, to achieve a reliable $\mathbf{U_{ex}}(t_{n+1})$ with SSP-RK3, the t_n-approximations on the outer two layers beyond $\mathcal{T}_{h,ex}$, together with the trace value of the outer third layer, must be reliable. In our applications, a region two layers wider than $\mathcal{T}_{h,ex}$ is named $\mathcal{T}_{h,ex1}$. All elements in $\mathcal{T}_{h,ex1}$ undergo the SSP-RK3 integration during the explicit phase of the hybrid scheme. Only the solutions corresponding to $\mathcal{T}_{h,ex}$ are retained as reliable approximations for t_{n+1}.

The calculation of $\mathbf{U}_{im}(t_{n+1})$ is straightforward. At the ith stage of LDIRK3, the local approximations of $t_n + c_{i,\mathrm{LDIRK3}}\Delta t$ are first achieved using SSP-RK3, providing the boundary conditions for the implicit integration on $\mathcal{T}_{h,im}$. Here, the region for the SSP-RK3 is denoted by $\mathcal{T}_{h,ex2}$. In Figure 1, $\mathcal{T}_{h,ex2}$ is comprised of five elements centered at δ because only the trace value of $\mathbf{U}(t_n + c_{i,\mathrm{LDIRK3}}\Delta t, \delta)$ is useful for the implicit integration on $\mathcal{T}_{h,im}$.

3.2.2. Constructing the Jacobian Matrix

During the implicit phase of the hybrid scheme, a Newton-type nonlinear solver is necessary. As is shown in Equation (19), the Jacobian matrix is a key to the solver.

Here, we describe an analytic method of constructing the matrix $\mathbf{J_R} \equiv \partial\mathbf{R}/\partial\mathbf{U}$, where $\mathbf{R}$ and $\mathbf{U}$ are defined in Equation (12). There are $(N_p \times K)^2$ values in $\mathbf{J_R}$. Being divided into K^2 squares, the (m, n)th square, denoted by $\mathbf{J}_\mathbf{R}^{m,n}$, equals $\partial\mathbf{R}^m/\partial\mathbf{U}^n$. According to Equation (5), $\mathbf{J}_\mathbf{R}^{m,n}$ is full of zeros except for the cases where $m = n$ or Ω_m neighbors on Ω_n.

We first consider $\mathbf{J}_\mathbf{R}^{m,m}$. By taking the partial derivative of the right-hand sides of Equation (5) with respect to u_j^m, the (i, j)th value in the square is calculated by:

$$\begin{aligned}\mathbf{J}_\mathbf{R}^{m,m}(i,j) = &\int_{\Omega_m} \left.\frac{\partial \mathbf{f}}{\partial u}\right|_{u_h^m} l_j^m \cdot \nabla l_i^m d\Omega - \int_{\partial\Omega_m} \frac{\partial \hat{H}}{\partial u_h^m} l_j^m l_i^m dS \\ &- \int_{\Omega_m} \varepsilon \nabla l_j^m \cdot \nabla l_i^m \mathrm{d}\Omega + \int_{\partial\Omega_m} \varepsilon \left(\frac{1}{2}\nabla l_j^m \cdot \mathbf{n}^m l_i^m - \sigma l_j^m l_i^m - l_j^m \mathbf{n}^m \cdot \frac{1}{2}\nabla l_i^m\right) \mathrm{d}S.\end{aligned} \tag{24}$$

For the LLF flux in Equation (7), we take:

$$\frac{\partial \hat{H}}{\partial u_h^m} = \frac{1}{2}\left.\frac{\partial \mathbf{f}}{\partial u}\right|_{u_h^m} \cdot \mathbf{n}^m + \frac{1}{2}\lambda. \tag{25}$$

Next, we consider $\mathbf{J}_\mathbf{R}^{m,n}$ where Ω_m neighbors on Ω_n. By taking the partial derivative of the right-hand sides of Equation (5) with respect to u_j^n, the (i, j)th value of the square is calculated as:

$$\mathbf{J}_\mathbf{R}^{m,n}(i,j) = -\int_{\partial\Omega_m} \frac{\partial \hat{H}}{\partial u_h^n} l_j^n l_i^m dS + \int_{\partial\Omega_m} \varepsilon \left(\frac{1}{2}\nabla l_j^n \cdot \mathbf{n}^m l_i^m + \sigma l_j^n l_i^m + l_j^n \mathbf{n}^m \cdot \frac{1}{2}\nabla l_i^m\right) \mathrm{d}S, \tag{26}$$

where, for the LLF flux,

$$\frac{\partial \hat{H}}{\partial u_h^n} = \frac{1}{2}\left.\frac{\partial \mathbf{f}}{\partial u}\right|_{u_h^m} \cdot \mathbf{n}^m - \frac{1}{2}\lambda. \tag{27}$$

Within $\mathbf{J_R}$, only the portions related to $\mathcal{T}_{h,im}$ are useful to the hybrid scheme. We restrict ourselves to calculating the squares corresponding to $\mathcal{T}_{h,im}$ and leave out the the squares related to $\mathcal{T}_{h,ex}$. This greatly reduces the required time and computing resources.

3.3. Nonlinear Solver

In our applications, an inexact Newton–Raphson method is used for solving the nonlinear equations. That is, within the time step $[t_n, t_{n+1}]$, the Jacobian matrix is computed with the data corresponding to t_n, then the matrix is reused in the remaining iterations as well as stages [25]. Without hardly affecting the convergence rate, the inexact method leads to a large reduction in computing expense.

For the purpose of solving the linear equations arising from the inexact Newton–Raphson method, a preconditioned generalized minimal residual (GMRES) method is adopted. The preconditioner is constructed using the incomplete lower-upper (ILU) decomposition method. To further accelerate

the convergence of the GMRES method, the reverse Cuthill–Mckee (RCM) algorithm is taken to narrow the difference between the preconditioner and the coefficient matrix for the linear equations. A comprehensive view of the linear solver can be found in, e.g., [28].

3.4. Stability

Here we discuss the regions of absolute stability of the time integrations schemes mentioned above. For the purpose of deciding the region, we apply the scheme to the model ODE:

$$\frac{du}{dt} = \lambda u, \tag{28}$$

where λ is a complex constant. For RK schemes, the numerical solution corresponding to t_{n+1} can be expressed as the product of that corresponding to t_n and an increment function, i.e.,

$$u_{n+1} = R(z)u_n, \tag{29}$$

where $z \equiv \lambda\Delta t$ and $\Delta t = t_{n+1} - t_n$. A time integration scheme is stable when $|R(z)| \leq 1$.

For the SSP-RK3 in Equation (15), the increment function is expressed as:

$$R(z)_{\mathrm{SSP-RK3}} = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3. \tag{30}$$

For the LDIRK3 in Equation (16),

$$R(z)_{\mathrm{LDIRK3}} = \frac{(\alpha\beta_1 - \alpha^2 + 2\alpha\beta_2 - \beta_2\gamma)z^2 + (2\alpha - \beta_1 - \beta_2)z - 1}{(\alpha z - 1)^3}, \tag{31}$$

where the coefficients $\alpha, \beta_1, \beta_2 and \gamma$ can be found in Equation (16).

In Figure 2 we plot the absolute stability regions for the SSP-RK3 scheme and the LDIRK3 scheme. For the evolution of the ODEs, e.g., Equation (11), the complex constant λ in Equation (28) can be explained as the eigenvalue with the biggest norm of the system. It quantifies the stiffness of the system and is determined by the by the specific way of spatial discretization. In [9], the stiffness portion related to the element size is denoted as "geometry-induced stiffness". This can be expressed as $\lambda = \lambda(h_k), \forall\Omega_k \in \mathcal{T}_h$.

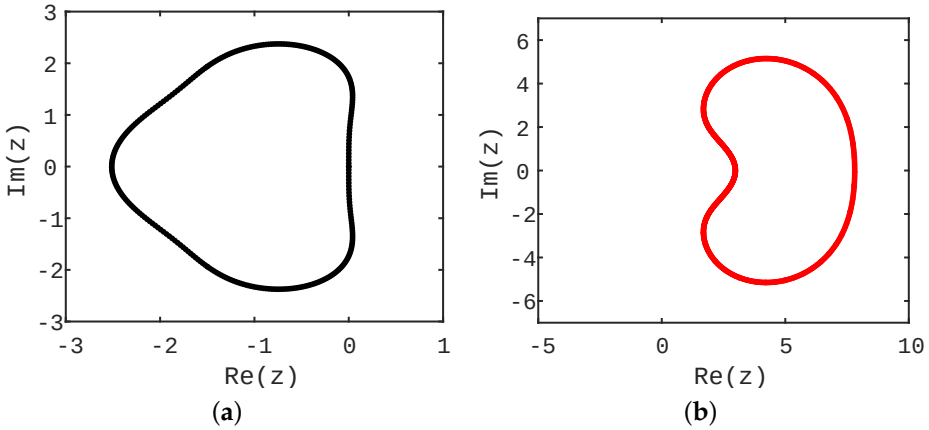


Figure 2. Absolute stability regions for different schemes. (**a**) SSP-RK3, interior of the black line. (**b**) LDIRK3, exterior of the red line.

As is shown in Figure 2a, the stability region of SSP-RK3 is bounded by a closed curve. Its time step have to vary with λ, i.e., $\Delta t = \Delta t(\lambda(h_k)) = \Delta t(h_k), \forall\Omega_k \in \mathcal{T}_h$, to ensure that $z \equiv \lambda\Delta t$ lying within the stability region. On the other hand, as is seen in Figure 2b, $\forall z, |z| = \infty$ belongs to the stability region of LDIRK3, making whose time step free from the affection of λ. In our hybrid scheme,

the elements in $\mathcal{T}_{h,ex}$ and $\mathcal{T}_{h,im}$ are separately treated with the SSP-RK3 scheme and LDIRK3 scheme. Therefore, its time step is limited by the elements sizes in $\mathcal{T}_{h,ex}$, but free from the element size restriction of $\mathcal{T}_{h,im}$, i.e., $\Delta t = \Delta t(h_k), \forall \Omega_k \in \mathcal{T}_{h,ex}$. To summarize, the time step for our hybrid scheme is decided in such a way that it maintains the stability of SSP-RK3 on $\mathcal{T}_{h,ex}$. Therefore, we believe Figure 2a can be reused to illustrate the stability region of our hybrid scheme, on condition taking in mind that $\Delta t = \Delta t(h_k), \forall \Omega_k \in \mathcal{T}_{h,ex}$.

4. Numerical Tests

We execute numerical tests with unsteady problems in this section. All computations are performed on a laptop equipped with an Intel Core 2.6 GHz processor and 6GB memory. For each problem, we apply the hybrid scheme to evolve the DG discretization. In addition, we also integrate the DG discretization with the SSP-RK3 scheme and a four-stage, third-order accurate SSP-RK scheme [29], denoted by SSP-RK3-4, to compare the performances of the schemes in terms of efficiency. In Appendix A, we provide the Butcher tableau of the SSP-RK3-4 scheme and plot its region of absolute stability. Compared with Figure 2a, the stability region for the SSP-RK3-4 scheme is bigger. According to [29], the available CFL number for the SSP-RK3-4 scheme is twice that for the SSP-RK3 scheme.

For our hybrid scheme, the time step length is decided by:

$$\begin{cases} \Delta t_{\text{hybrid}} = \frac{1}{d}\frac{1}{(2N+1)} \min\left\{\Delta t_{ex}, \Delta t_{im}\right\}, \\ \Delta t_{ex} = C_{ex} \cdot \min\limits_{\Omega_k \in \mathcal{T}_{h,ex}} \left\{\frac{h_k}{\lambda_k}\right\}, \\ \Delta t_{im} = C_{im} \cdot \min\limits_{\Omega_k \in \mathcal{T}_{h,im}} \left\{\frac{h_k}{\lambda_k}\right\}, \end{cases} \tag{32}$$

where d denotes the spatial dimension of the problem, h_k takes the element length for 1D problems or the inscribed circle diameter of the element in 2D problems, λ_k differs with the specific problem, and C_{ex} and C_{im} are adjustable parameters. We take $C_{im}/C_{ex} \approx S$, with S defined in Equation (21). For the SSP-RK3 scheme, the time step is decided by:

$$\Delta t_{\text{RK3}} = \frac{C_{ex}}{(2N+1)} \cdot \min_{\Omega_k \in \mathcal{T}_h} \left\{\frac{h_k}{\lambda_k}\right\}. \tag{33}$$

For the SSP-RK3-4 scheme, we set the time step two times as big as that for the SSP-RK3 scheme, i.e.,

$$\Delta t_{\text{RK3-4}} = 2\Delta t_{\text{RK3}}. \tag{34}$$

For problems with analytic solutions, we calculate the L^2 errors of the numerical results. Given the numerical solution, u_h, and the analytic solution, u_{analy}, the L^2 error is defined as [30]:

$$\|e\|_2 = \sqrt{\sum_{k=1}^{K} \int_{\Omega_k} \left(u_h - u_{analy}\right)^2 d\Omega}. \tag{35}$$

4.1. 1D Linear Advection Problem

The problem is governed by:

$$\begin{cases} \frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = 0, & x \in [-1,1], \quad t \geq 0, \\ q\,(0,x) = 1 + \cos\left(2\pi x\right), \end{cases} \tag{36}$$

and the boundary conditions are periodic. The analytic solution to the problem is $q(t,x) = 1 + \cos(2\pi(x - at))$.

We numerically solve the problem with $a = 1$ on three meshes, denoted by "Coarse 1D", "Middle 1D", and "Fine 1D". During the mesh generation, the computational domain is divided into three regions: the left, the central and the right. The left and the right regions are symmetric with respect to

the origin. The grids are uniformly distributed in the two regions. The range of the central region is kept four times the element size in the left/right region. In the central region, the grids are symmetrically distributed. The smallest element size in the central region is kept one-tenth of that in the left/right region, resulting in $S = 10$. In the central region, the ratio between the adjacent element sizes is kept less than 1.5. The biggest element sizes in Coarse 1D, Middle 1D, and Fine 1D are 0.1, 0.05, and 0.025, respectively.

Here are some remarks on the meshes. Generally the ratio between the adjacent element sizes should not exceed 1.2 in order to avoid big interpolation error. However, in the DG method, the high-order accuracy is achieved by increasing node numbers within the elements. There is no need to expand the stencil, as is done in the finite difference (FD) method or the finite volume (FV) method. In addition, as is shown in Equations (6) and (9), the DG solution is allowed to be discontinuous on element interfaces. The coupling between the local approximations is weak. Therefore, at least for the problems with smooth solutions, we believe the impact of the big element size change is endurable for the DG method. Last but not least, this kind of sudden jump in the size of element also appears in [9]. See Figure 6a therein for reference.

Figure 3 shows the Coarse 1D mesh and its elements grouping. The roles of $\mathcal{T}_{h,im}$, $\mathcal{T}_{h,ex}$, $\mathcal{T}_{h,ex1}$, and $\mathcal{T}_{h,ex2}$ during the temporal integration are same as in Figure 1. Since the time step is restricted by the elements that subjected to the explicit integration, i.e., the elements in $\mathcal{T}_{h,ex1}$, the outer two layers of elements in $\mathcal{T}_{h,im}$ are coarse. The elements groupings for Middle 1D and Fine 1D are similar to that for Coarse 1D.

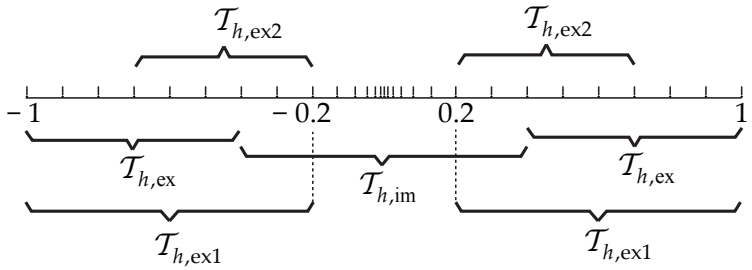


Figure 3. Coarse 1D mesh.

For each calculation, we take $N = 2$ for local approximations. We spatially discretize the problem according to Section 2. Since there is no diffusive item in the governing equation, we leave the diffusion-related items in the discretizations as zeros. For deciding the time steps, we take $d = 1$, $C_{ex} = 1$, $C_{im} = 10$ and $\lambda_k = a$. Each calculation is run up to $t = 1$.

Table 1 shows the L^2 error of the numerical result for each mesh and the convergence order of the errors. We found that the convergence order in space is $N + 1$, which is optional for the DG method in solving conservation laws [6]. According to [6], to guarantee the domination of the spatial error, the accurate order for the time integration scheme is at least $N + 1$. The optional convergence order in Table 1 indicates that the hybrid scheme maintains the high-order accuracy of SSP-RK3 or LDIRK3. It also proves that the big size changes in the mesh has little effect on the accuracy of the method.

Table 2 compares the efficiency of the SSP-RK3 scheme (RK3), the SSP-RK3-4 scheme (RK3-4) and our hybrid scheme (hybrid). The second to the fourth columns show the average time step lengths for different schemes. The fifth to the last columns record the CPU time (in seconds) for the simulations. In accordance with Equations (32)–(34), the average time steps for the SSP-RK3-4 scheme and the hybrid scheme are twice and S times as that for the SSP-RK3 scheme, respectively. On the other hand, the CPU time indicate that the acceleration of the computation does not scale linearly with the amplification of the time step. For the SSP-RK3-4 scheme, this is owing to the computation related to the extra fourth stage. For our hybrid scheme, this may comes from the nonlinear-solver-related calculations, e.g., constructing the Jacobian matrix and solving linear systems. Nevertheless, based on

the given meshes, it is clear that the hybrid scheme is superior towards the SSP-RK3 scheme or the SSP-RK3-4 scheme in terms of efficiency.

Table 1. L^2 errors, 1D linear advection problem.

Mesh	L^2 Error	Order
Coarse 1D	1.76×10^{-3}	
Middle 1D	2.35×10^{-4}	2.90
Fine 1D	2.97×10^{-5}	2.98

Table 2. Comparison of different schemes, 1D linear advection problem.

Mesh	Δt_{RK3}	Δt_{RK3-4}	Δt_{hybrid}	$T_{CPU,RK3}/s$	$T_{CPU,RK3-4}/s$	$T_{CPU,hybrid}/s$
Coarse 1D	4.0×10^{-3}	8.0×10^{-3}	4.0×10^{-2}	3.73	2.14	1.46
Middle 1D	2.0×10^{-3}	4.0×10^{-3}	2.0×10^{-2}	7.46	3.90	2.84
Fine 1D	1.0×10^{-3}	2.0×10^{-3}	1.0×10^{-2}	14.51	7.47	4.59

4.2. 1D Linear Advection-Difussion Problem

The problem is governed by:

$$\begin{cases} \frac{\partial q}{\partial t} + a\frac{\partial q}{\partial x} = \epsilon \frac{\partial^2 q}{\partial x^2}, & x \in [-1,1], \quad t \geq 0, \\ q(0,x) = \sin(\pi x). \end{cases} \tag{37}$$

The analytic solution to the problem is $q(t,x) = \exp\left(-\pi^2 \epsilon t\right) \sin\left(\pi\left(x - at\right)\right)$. The boundary conditions are applied using the analytic solution.

We numerically solve the problem with $a = 1$ and $\epsilon = 0.01$. The meshes being used are same as in Section 4.1. $N = 1$ is taken for the local approximations. The spatial discretization for the problem is the same as in Section 2. To find the upper limit of C_{em}, we first apply the SSP-RK3 scheme. The definition of λ_k is given by:

$$\lambda_k = a + \frac{N}{2}\frac{\epsilon}{h_k}. \tag{38}$$

Through numerical tests, we find that the maximum of C_{ex} for maintaining the calculation stability is 0.3. For defining the time step for the hybrid scheme, we take $d = 1$, $C_{ex} = 0.3$, $C_{im} = 3$. Each calculation is run up to $t = 0.1$.

Table 3 shows the L^2 errors of the numerical results and their convergence order. According to [8] (p. 75), when the NIPG approach is selected for dealing with the diffusive item, the theoretical convergence order measured in L^2-norm is $N + 1$ when N is odd. We found that the numerical results matched well with the theory. Table 4 compares the efficiency of different schemes. Similar to the results in Section 4.1, in every simulation, the hybrid scheme enjoyed the biggest available time step and consumed the least CPU time.

Table 3. L^2 errors, 1D linear advection-diffusion problem.

Mesh	L^2 Error	Order
Coarse 1D	6.56×10^{-3}	
Middle 1D	1.45×10^{-3}	2.18
Fine 1D	3.10×10^{-4}	2.23

Table 4. Comparison of different schemes, 1D linear advection-diffusion problem.

Mesh	$\Delta t_{\mathbf{RK3}}$	$\Delta t_{\mathbf{RK3-4}}$	$\Delta t_{\mathbf{hybrid}}$	$T_{\mathbf{CPU,RK3}}/s$	$T_{\mathbf{CPU,RK3-4}}/s$	$T_{\mathbf{CPU,hybrid}}/s$
Coarse 1D	6.7×10^{-4}	1.3×10^{-3}	6.7×10^{-3}	2.35	1.40	0.97
Middle 1D	2.5×10^{-4}	5.0×10^{-4}	2.5×10^{-3}	5.81	3.35	2.53
Fine 1D	8.3×10^{-5}	1.7×10^{-4}	8.3×10^{-4}	17.55	10.84	5.98

4.3. 1D Euler System

The problem is governed by:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} = \mathbf{0}, \quad x \in [-1, 1], \quad t \geq 0, \tag{39}$$

where:

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (E + p)u \end{pmatrix}, \tag{40}$$

where ρ is the density, u is the velocity, p is the pressure, and $E = \rho u^2/2 + p/(\gamma - 1)$ is the total energy. Here $\gamma = 1.4$.

We define the initial condition as:

$$\begin{cases} \rho\,(0, x) = 1 + 0.2 \sin(\pi x), \\ u\,(0, x) = 1, \\ p\,(0, x) = 1, \end{cases} \tag{41}$$

and apply periodic boundary conditions. The analytic solution to the problem is:

$$\begin{cases} \rho\,(t, x) = 1 + 0.2 \sin(\pi (x - t)), \\ u\,(t, x) = 1, \\ p\,(t, x) = 1. \end{cases} \tag{42}$$

The meshes being used are same as in Section 4.1. $N = 2$ is taken for local approximations. During the spatial discretization, the Vijayasundaram flux [31]:

$$\hat{\mathbf{H}}_{Vija} \equiv (\hat{H}_1, \hat{H}_2, \hat{H}_3)^T = \mathbf{P}^- \left(\langle \mathbf{q} \rangle, \mathbf{n}^k \right) \mathbf{q}^- + \mathbf{P}^+ \left(\langle \mathbf{q} \rangle, \mathbf{n}^k \right) \mathbf{q}^+, \tag{43}$$

is chosen as the inviscid numerical flux. The treatment for each equation of the governing system is similar to that in Section 2. In our application, the ODEs pending to the temporal integration are:

$$\frac{d}{dt} \mathbf{Q}_{sys} = \mathbf{M}_{sys}^{-1} \mathbf{R}_{sys} \equiv \mathbf{F}_{sys}, \tag{44}$$

where:

$$\mathbf{Q}_{sys} = \begin{pmatrix} \mathbf{Q}_1 \\ \mathbf{Q}_2 \\ \mathbf{Q}_3 \end{pmatrix}, \quad \mathbf{M}_{sys} = \begin{pmatrix} \mathbf{M}_1 & & \\ & \mathbf{M}_2 & \\ & & \mathbf{M}_3 \end{pmatrix}, \quad \mathbf{R}_{sys} = \begin{pmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{R}_3 \end{pmatrix}, \quad \mathbf{F}_{sys} = \begin{pmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \mathbf{F}_3 \end{pmatrix}, \tag{45}$$

where $\mathbf{Q}_e$, $\mathbf{M}_e$, $\mathbf{R}_e$, $\mathbf{F}_e$, $e = 1, 2, 3$, correspond to the eth equation of the governing system. Their definitions are similar to that in Equation (12).

When integrating Equation (44) with the hybrid scheme, it is necessary to construct the matrix $\mathbf{J}_{\mathbf{R},sys} \equiv \partial \mathbf{R}_{sys} / \partial \mathbf{Q}_{sys}$. For the 1D Euler system, $\mathbf{J}_{\mathbf{R},sys}$ consists of 3×3 blocks. The calculation for the (e_1, e_2)th block, i.e., $\partial \mathbf{R}_{e1} / \partial \mathbf{Q}_{e2}$, is similar to that in Section 3.2.2. The matrices $\mathbf{P}^-$ and $\mathbf{P}^+$ are directly

taken as the partial derivatives of $\hat{\mathbf{H}}_{Vija}$ with respect to $\mathbf{q}^-$ and $\mathbf{q}^+$, respectively. For defining time steps using Equations (32)–(34), we take $d = 1$, $C_{em} = 1$ and $C_{im} = 10$. The definition of λ_k is:

$$\lambda_k = \max_{\mathbf{x}\in\partial\Omega_k} \left\{ \ |u|\,, \quad ||u| - c|\,, \quad |u| + c \ \right\}, \tag{46}$$

where $c = \sqrt{\gamma p/\rho}$ is the speed of sound. Each calculation is run up to $t = 0.5$.

Table 5 shows the L^2 errors of the densities and their convergence order. We found that the optimal $(N+1)$th order is well attained. Table 6 compares the available time step lengths and the CPU time for different schemes. Again the hybrid scheme enjoyed the biggest time step and the least CPU time.

Table 5. L^2 errors, density, 1D Euler system.

Mesh	L^2 Error	Order
Coarse 1D	4.62×10^{-5}	
Middle 1D	5.89×10^{-6}	2.97
Fine 1D	7.52×10^{-7}	2.97

Table 6. Comparison of different schemes, 1D Euler system.

Mesh	$\Delta t_{\mathbf{RK3}}$	$\Delta t_{\mathbf{RK3-4}}$	$\Delta t_{\mathbf{hybrid}}$	$T_{\mathbf{CPU,RK3}}/s$	$T_{\mathbf{CPU,RK3-4}}/s$	$T_{\mathbf{CPU,hybrid}}/s$
Coarse 1D	6.9×10^{-4}	1.4×10^{-3}	6.9×10^{-3}	13.25	7.46	7.00
Middle 1D	3.4×10^{-4}	6.9×10^{-4}	3.4×10^{-3}	24.60	13.66	11.51
Fine 1D	1.7×10^{-4}	3.4×10^{-4}	1.7×10^{-3}	45.24	27.71	22.51

4.4. 2D Euler System

The problem is governed by:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \mathbf{0}, \quad (x, y) \in [-5, 5] \times [-5, 5], \quad t \geq 0, \tag{47}$$

where:

$$\mathbf{q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (E + p)u \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (E + p)v \end{pmatrix}. \tag{48}$$

In Equation (48), ρ is the density, u and v are the velocity components, p is the pressure, and $E = \rho(u^2 + v^2)/2 + p/(\gamma - 1)$ with $\gamma = 1.4$ is the total energy. We consider an isentropic vortex problem with the analytic solution [7] (p. 209):

$$\begin{cases} \rho\,(t, x, y) = \left(1 - \frac{\gamma-1}{16\gamma\pi^2} \exp\left(2 - 2(x-t)^2 - 2y^2\right)\right)^{\frac{1}{\gamma-1}}, \\ u\,(t, x, y) = 1 - \frac{y}{2\pi} \exp\left(1 - (x-t)^2 - y^2\right), \\ v\,(t, x, y) = \frac{x}{2\pi} \exp\left(1 - (x-t)^2 - y^2\right), \\ p\,(t, x, y) = \rho(t, x, y)^{\gamma}. \end{cases} \tag{49}$$

It is used for defining the initial condition and boundary conditions for numerical simulations.

We solve the problem on three meshes, denoted by "Coarse 2D", "Middle 2D", and "Fine 2D". In each mesh, the domain is divided into the surrounding region and the central region. The coarse elements in the surrounding region are made by diagonalizing the squares of same sizes. The central region is a square whose side length is kept two times the leg length for the coarse element. The elements in the central region are refined artificially. The smallest element length is kept one-eighth

of that in the surrounding region, leading to $S = 8$. The ratio between the adjacent element sizes is kept less than 1.5 during the refinement. Since all elements within the mesh are are isosceles right triangles, the skewness (measured by the normalized equiangular method) for the mesh is constant 0.25 and the aspect ratio is 1.41. The biggest leg lengths in Coarse 2D, Middle 2D, and Fine 2D are 0.5, 0.25, and 0.125, respectively.

Here are some remarks concerning the quality of the 2D meshes. Similar to the 1D cases, we believe the impact of the radical size change is minimized by the independence of the local approximations in the DG method. In addition, the radical size jump in element size is common in relevant researches, see, e.g., Figure 8 in [9], Figure 5 in [11] and Figure 1c in [12], for reference.

Figure 4 shows the Coarse 2D mesh and its elements grouping. The grouping strategy for the 2D elements is similar to the 1D case. The relationship between the color and the membership for an element is given in Table 7. The elements groupings for Middle 2D and Fine 2D are similar to that for Coarse 2D.

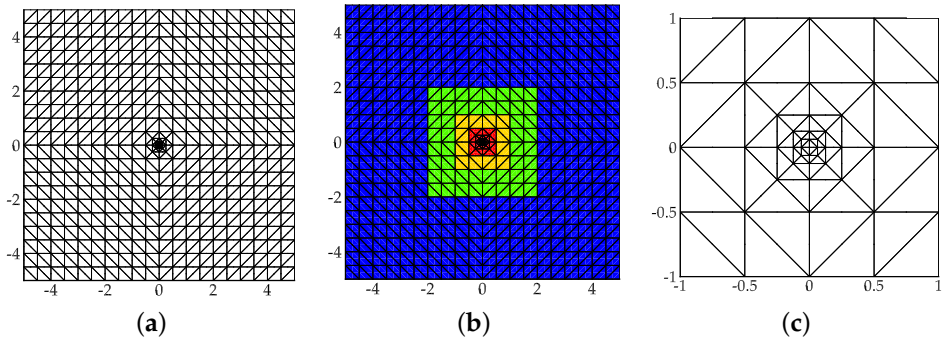


Figure 4. (**a**) Coarse 2D mesh for the isentropic vortex problem. (**b**) Elements coloring for Coarse 2D. (**c**) Zoom in for $\mathcal{T}_{h,im}$ of Coarse 2D.

Table 7. Coloring strategy for 2D meshes.

Group	Color
$\mathcal{T}_{h,im}$	red + yellow
$\mathcal{T}_{h,ex}$	green + blue
$\mathcal{T}_{h,ex1}$	yellow + green +blue
$\mathcal{T}_{h,ex2}$	yellow + green

For each calculation, $N = 2$ is taken for local approximations. The spatial discretization for the problem is similar to that in Section 4.3. For defining time steps, we take $d = 2$, $C_{ex} = 1$ and $C_{im} = 8$. The definition of λ_k is:

$$\lambda_k = \max_{\mathbf{x}\in\partial\Omega_k} \left\{ \left|\mathbf{v}\cdot\mathbf{n}^k\right|, \quad \left|\left|\mathbf{v}\cdot\mathbf{n}^k\right| - c\right|, \quad \left|\mathbf{v}\cdot\mathbf{n}^k\right| + c \right\}, \tag{50}$$

where $\mathbf{v} = (u, v)$ and $c = \sqrt{\gamma p/\rho}$. Each calculation is run up to $t = 3$.

Table 8 shows the L^2 errors of the densities and their order of convergence. Similar to the 1D cases, the optimal $(N+1)$th order for the DG method is achieved. It indicates that the high-order accuracy of the SSP-RK3 scheme or the LDIRK3 scheme is maintained by the hybrid scheme, and the impact of the big size changes in the mesh is negligible. Table 9 compares the efficiency of different time integration schemes. The hybrid scheme performs even better in simulating 2D problems than in 1D problems. For each 2D simulation, the CPU time it took was merely one-third of that for the SSP-RK3 scheme or less than half of that for the SSP-RK3-4 scheme.

Table 8. L^2 errors, density, 2D Euler system.

Mesh	L^2 Error	Order
Coarse 2D	1.32×10^{-3}	
Middle 2D	1.84×10^{-4}	2.84
Fine 2D	2.44×10^{-5}	2.91

Table 9. Comparison of different schemes, 2D Euler system.

Mesh	$\Delta t_{\mathbf{RK3}}$	$\Delta t_{\mathbf{RK3-4}}$	$\Delta t_{\mathbf{hybrid}}$	$T_{\mathbf{CPU,RK3}}/s$	$T_{\mathbf{CPU,RK3-4}}/s$	$T_{\mathbf{CPU,hybrid}}/s$
Coarse 2D	1.1×10^{-3}	2.3×10^{-3}	1.2×10^{-3}	104	78	38
Middle 2D	5.8×10^{-4}	1.2×10^{-3}	6.2×10^{-3}	832	568	262
Fine 2D	2.9×10^{-4}	5.8×10^{-4}	3.1×10^{-3}	7815	6254	2349

4.5. 2D Navier-Stokes System

The problem is governed by:

$$\frac{\partial \mathbf{q}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} = \frac{\partial \mathbf{r}_1}{\partial x} + \frac{\partial \mathbf{r}_2}{\partial y}, \tag{51}$$

where $\mathbf{q}$, $\mathbf{f}$, and $\mathbf{g}$ are defined same to Equation (48) and:

$$\mathbf{r}_1 = \begin{pmatrix} 0 \\ \tau_{11} \\ \tau_{12} \\ \tau_{11}u + \tau_{12}v + \kappa\frac{\partial T}{\partial x} \end{pmatrix}, \quad \mathbf{r}_2 = \begin{pmatrix} 0 \\ \tau_{21} \\ \tau_{22} \\ \tau_{21}u + \tau_{22}v + \kappa\frac{\partial T}{\partial y} \end{pmatrix}, \tag{52}$$

where

$$\tau_{11} = \mu(\tfrac{4}{3}\tfrac{\partial u}{\partial x} - \tfrac{2}{3}\tfrac{\partial v}{\partial y}), \quad \tau_{12} = \tau_{21} = \mu(\tfrac{\partial u}{\partial x} + \tfrac{\partial v}{\partial y}), \quad \tau_{22} = \mu(\tfrac{4}{3}\tfrac{\partial v}{\partial y} - \tfrac{2}{3}\tfrac{\partial u}{\partial x}). \tag{53}$$

In Equations (52) and (53), u and v are the velocity components, μ is the viscosity coefficient, κ is the heat conductivity, and T is the temperature. To derive a dimensionless form of the governing system, we introduce a reference density ρ_{ref}, a reference velocity V_{ref}, and a reference length L_{ref}. L_{ref}/V_{ref} defines a reference time, $\rho_{ref}V_{ref}^2$ defines a reference pressure, and V_{ref}^2/c_v defines a reference temperature with c_v denoting the specific heat at constant volume. Normalizing Equation (51) with the reference quantities and denoting the dimensionless quantities with the apostrophe, we obtain:

$$\frac{\partial \mathbf{q}'}{\partial t'} + \frac{\partial \mathbf{f}'}{\partial x'} + \frac{\partial \mathbf{g}'}{\partial y'} = \frac{\partial \mathbf{r}_1{}'}{\partial x'} + \frac{\partial \mathbf{r}_2{}'}{\partial y'}, \tag{54}$$

where $\mathbf{q}'$, $\mathbf{f}'$, and $\mathbf{g}'$ are defined as:

$$\mathbf{q}' = \begin{pmatrix} \rho' \\ \rho'u' \\ \rho'v' \\ E' \end{pmatrix}, \quad \mathbf{f}' = \begin{pmatrix} \rho'u' \\ \rho'u'^2 + p' \\ \rho'u'v' \\ (E'+p')u' \end{pmatrix}, \quad \mathbf{g}' = \begin{pmatrix} \rho'v' \\ \rho'u'v' \\ \rho'v'^2 + p' \\ (E'+p')v' \end{pmatrix}, \tag{55}$$

and $\mathbf{r}_1{}'$ and $\mathbf{r}_2{}'$ are defined as:

$$\mathbf{r}_1{}' = \begin{pmatrix} 0 \\ \tau_{11}{}' \\ \tau_{12}{}' \\ \tau_{11}{}'u' + \tau_{12}{}'v' + \frac{\gamma}{\mathrm{Re}_{ref}\,\mathrm{Pr}}\frac{\partial T'}{\partial x'} \end{pmatrix}, \quad \mathbf{r}_2{}' = \begin{pmatrix} 0 \\ \tau_{21}{}' \\ \tau_{22}{}' \\ \tau_{21}{}'u' + \tau_{22}{}'v' + \frac{\gamma}{\mathrm{Re}_{ref}\,\mathrm{Pr}}\frac{\partial T'}{\partial y'} \end{pmatrix}, \tag{56}$$

where

$$\tau_{11}{}' = \frac{1}{\mathrm{Re}_{ref}}\left(\frac{4}{3}\frac{\partial u'}{\partial x'} - \frac{2}{3}\frac{\partial v'}{\partial y'}\right), \quad \tau_{12}{}' = \tau_{21}{}' = \frac{1}{\mathrm{Re}_{ref}}\left(\frac{\partial u'}{\partial x'} + \frac{\partial v'}{\partial y'}\right), \quad \tau_{22}{}' = \frac{1}{\mathrm{Re}_{ref}}\left(\frac{4}{3}\frac{\partial v'}{\partial y'} - \frac{2}{3}\frac{\partial u'}{\partial x'}\right). \tag{57}$$

In Equations (56) and (57), $\mathrm{Re}_{ref} = \rho_{ref} V_{ref} L_{ref}/\mu$ is the reference Reynolds number and $\mathrm{Pr} = \gamma c_v \mu/\kappa$ is the Prandtl number. We take $\gamma = 1.4$ and $\mathrm{Pr} = 0.71$ for air. In our applications, $\rho_{ref} = \rho_\infty/\gamma$, $V_{ref} = c_\infty$ and L_{ref} takes the cylinder diameter. The speed of sound is calculated by $c = \sqrt{\gamma p/\rho}$ and the subscript ∞ refers to the state of the free-stream.

Figure 5a shows the mesh for the numerical simulation. It is generated in a way similar to that in Figure 14, [9] and Figure 10, [32]. The diameter for the outer boundary is 40 times as big as that for the cylinder. There are 3840 triangular elements in the mesh. They are achieved by diagonalizing 48×40 structured rectangular elements. The heights of the first five layers of elements adhering to the cylinder are 0.02, 0.03, 0.05, 0.12 and 0.12. In the first layer of elements, we achieve the maximum of the aspect ratio, 3.25, and the maximum of the skewness (measured by the normalized equiangular method), 0.72. The biggest size changes appear at the third and the fourth layers of elements, i.e., $0.12/0.05 = 2.4$. Within the mesh, the maximum and minimum of h'_k are 0.0607 and 0.0168, respectively, resulting in $S \approx 4$. Figure 5b presents the coloring of the elements. The coloring strategy is the same as in Table 8.

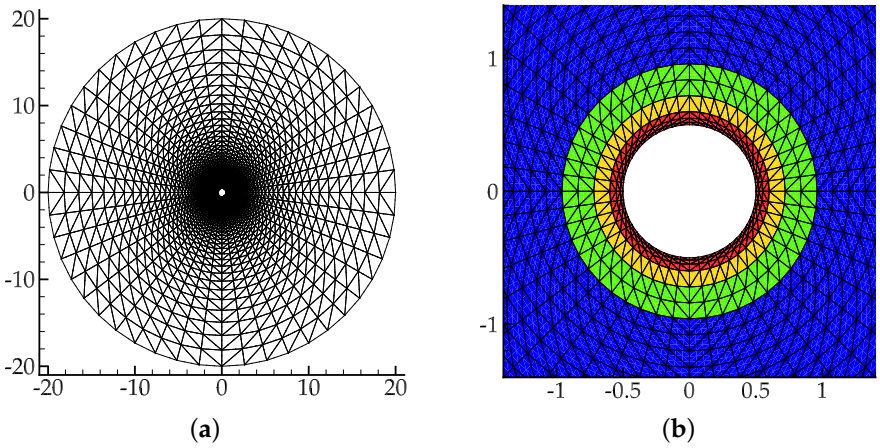


Figure 5. (**a**) Mesh for the flow around cylinder. (**b**) Elements coloring for the mesh.

In our computation, the state for the free-stream is defined by:

$$\mathbf{b}'_\infty = \begin{pmatrix} \rho'_\infty \\ u'_\infty \\ v'_\infty \\ p'_\infty \end{pmatrix} = \begin{pmatrix} \gamma \\ 0.3 \\ 0 \\ 1 \end{pmatrix}, \tag{58}$$

leading to $\mathrm{Ma}_\infty = 0.3$, with the Mach number calculated by $\mathrm{Ma} = \sqrt{u^2 + v^2}/c$. Besides, the Reynolds number of the free-stream is set to $\mathrm{Re}_\infty = 150$. To speed up the simulation, our computation consists of two periods. In the first period, we pursue the steady inviscid solution to the problem. That is, the flow is governed by the 2D Euler system, with $\mathbf{b}'_\infty$ defining the initial condition. During the computation, characteristic boundary conditions [33] and inviscid wall conditions are applied to the outer boundary and the cylinder surface, respectively. In the second period, the governing system for the problem is replaced by Equation (54). The inviscid solution just obtained is used for initializing the flow. Characteristic boundary conditions and viscous wall conditions [8] (p. 484) are applied during the remaining computation.

In our computation, $N = 3$ is taken for local approximations. The spatial discretization for the convective items are the same as that in Section 4.4. The NIPG approach for treating the viscous items can be found in [8] (pp. 486–491). To find the upper limit of C_{em}, we first integrate the DG discretization using the SSP-RK3 scheme, with the time step length defined by:

$$\Delta t' = \frac{1}{2}\frac{C_{ex}}{(2N+1)} \cdot \min_{\Omega_k \in \mathcal{T}_h} \left\{ \frac{h'_k}{\lambda'_k} \right\}, \tag{59}$$

where:

$$\lambda'_k = \max_{x \in \partial\Omega_k} \left\{ \left| \mathbf{v}' \cdot \mathbf{n}^k \right|, \quad \left| \left| \mathbf{v}' \cdot \mathbf{n}^k \right| - c' \right|, \quad \left| \mathbf{v}' \cdot \mathbf{n}^k \right| + c' \right\} + \frac{N}{\mathrm{Re}_{ref}\rho' {h'}_k}. \tag{60}$$

where $\mathbf{v}' = (u', v')$, $c' = \sqrt{\gamma p'/\rho'}$, and h'_k is the dimensionless element size.

Through numerical tests, we found that the maximum of C_{ex} for maintaining the calculation stability is 0.8. Next, the hybrid scheme with $\Delta t' = 0.0025$ is used for the time integration. The fixed time step length is close to one obtained by Equation (32) with $d = 2$, $C_{em} = 0.8$ and $C_{im} = 4$.

According to the direct numerical simulations (DNS) results in [34] and the experimental results referred to therein, the flow is unsteady with $\mathrm{Ma}_\infty = 0.3$ and $\mathrm{Re}_\infty = 150$. Over time, vortices periodically shed from the cylinder, causing fluctuations in the physical quantities in the wake of the cylinder. Figure 6 shows the numerical results at $t' = 408$. The fluctuations in the fields and the bending of the streamlines indicate that the shedding of vortices has been set up well. During the calculation, the lift coefficient C_L of the cylinder is recorded every 50 time steps. The time history for C_L from $t' = 350$ to $t' = 408$ is shown in Figure 7. It can be found that the fluctuation period of C_L is $T'_{shed} = 18$. Table 10 compares our numerical results with that in reference [34]. The matches between the two sets of results illustrate the usefulness of the hybrid scheme.

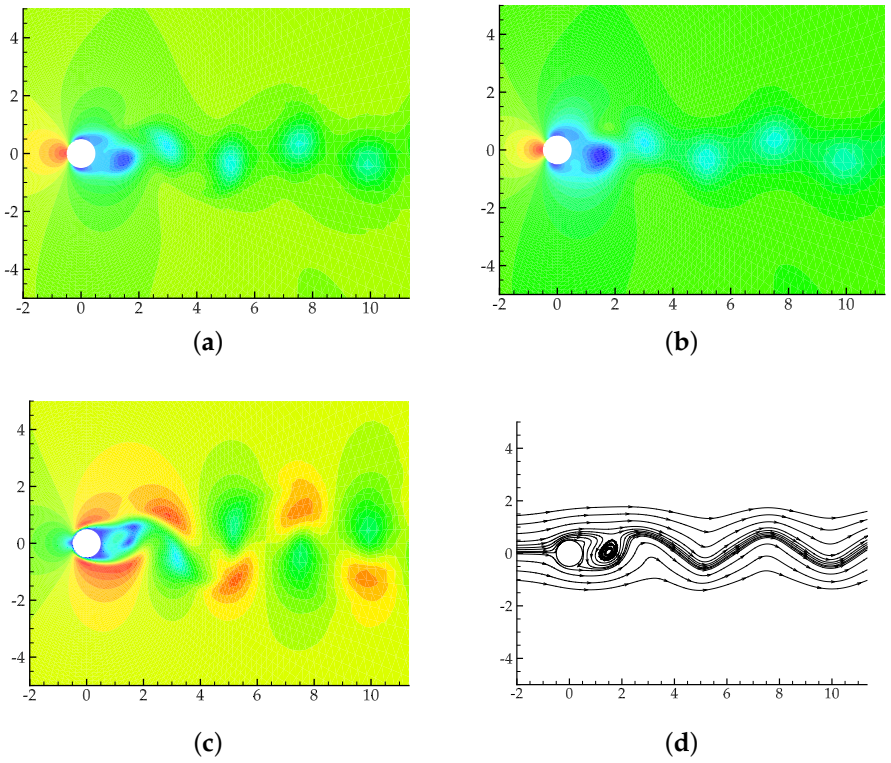


Figure 6. Numerical results for the flow over a cylinder, $t' = 408$. (**a**) Contours for $\rho' - \rho'_\infty$, 20 equally spaced from −0.13 to 0.06. (**b**) Contours for $p' - p'_\infty$, 20 equally spaced from −0.09 to 0.07. (**c**) Contours for Ma, 20 equally spaced from 0 to 0.43. (**d**) Stream lines plot.

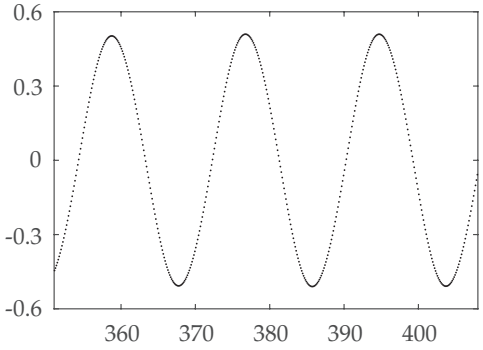


Figure 7. Time history of C_L, t'=350–408, flow around a cylinder.

Table 10. Results comparison, 2D Navier-Stokes system.

Data	Strouhal Number	C_L Amplitude
Numerical	0.185	0.51
Reference	0.183	0.52

Table 11 presents the available dimensionless time step length and the CPU time (in seconds) cost per time step for each scheme. After dividing the CPU time with the time step length for each scheme, we got the required CPU time for the simulation corresponding to unit dimensionless time, i.e., 2985 s for the SSP-RK3 scheme, 1923 s for the SSP-RK3-4 scheme and 1480s for our hybrid scheme. It is clear that the hybrid scheme performs best in terms of efficiency.

Table 11. Comparison of different schemes, 2D Navier-Stokes system.

$\Delta t'_{\mathbf{RK3}}$	$\Delta t'_{\mathbf{RK3-4}}$	$\Delta t'_{\mathbf{hybrid}}$	$\Delta T_{\mathbf{CPU,RK3}}/s$	$\Delta T_{\mathbf{CPU,RK3-4}}/s$	$\Delta T_{\mathbf{CPU,hybrid}}/s$
6.7×10^{-4}	1.3×10^{-3}	2.5×10^{-3}	2.0	2.5	3.7

5. Conclusions

In this work, we developed a hybrid scheme for the temporal integration of the DG discretization for the unsteady convection-dominated problem. The hybrid scheme consists of two popular high-order schemes: the SSP-RK3 scheme and the LDIRK3 scheme. By separately treating the coarse elements and the refined elements with an explicit scheme and an implicit scheme, a time step bigger than that for a purely explicit scheme can be reached at a comparatively low cost. Using the independence of the local approximation for the DG method, we proposed a strategy of grouping the elements that is necessary for maintaining the stability and reducing the computing cost of the hybrid scheme. We tested our scheme by numerically solving several threshold problems. The matches between the numerical solutions and the reference results proved the usefulness of the hybrid scheme. The comparison with the traditional explicit schemes testified the effectiveness of the hybrid scheme on the meshes with certain level of geometry-induced stiffness.

Author Contributions: Conceptualization, methodology, software, validation, formal analysis, investigation, data curation, writing—original draft preparation, writing—review and editing, visualization: L.L.; resources, supervision, funding acquisition: S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 91530325.

Acknowledgments: The first author is sincerely thankful to Fujun Liu from the School of Mathematics and System Science, Beihang University for the advice on the setting for the last numerical test. The valuable suggestions given by the anonymous reviewers are deeply appreciated by the authors.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

A new class of SSP-RK schemes was developed in reference [29]. By allowing the stage numbers bigger than the order numbers, the available CFL numbers for the new SSP-RK schemes are larger than those for the traditional SSP-RK schemes. A four-stage, third-order accurate scheme of this class is denoted by SSP-RK3-4. Its Butcher tableau is given as:

$$\left(\begin{array}{c|c} c_j & a_{ij} \\ \hline & b_j \end{array}\right)_{\mathrm{SSP-RK3-4}} = \left(\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 & 0 \\ 1 & 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/6 & 1/6 & 1/6 & 0 \\ \hline & 1/6 & 1/6 & 1/6 & 1/2 \end{array}\right). \tag{A1}$$

Similar to that in Section 3.4, we deduce the increment function for the scheme, i.e.,

$$R(z)_{\mathrm{SSP-RK3-4}} = 1 + z + \frac{1}{2}z^2 + \frac{1}{6}z^3 + \frac{1}{48}z^4, \tag{A2}$$

and plot the stability region in Figure A1. For more information of the SSP-RK3-4 scheme, the reader is referred to reference [29].

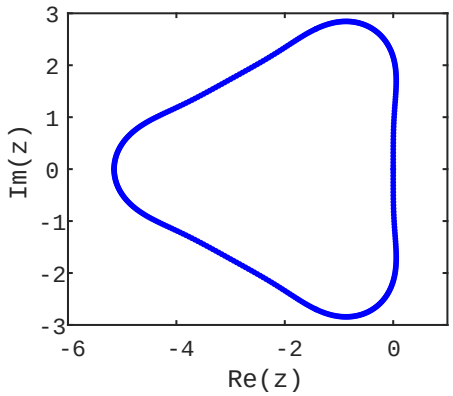


Figure A1. Absolute stability region for the SSP-RK3-4 scheme, interior of the blue line.

References

1. Reed, W.; Hill, T. *Triangular Mesh Methods for the Neutron Transport Equation*; Technical Report LA-UR-73-479; Los Alamos Scientific Laboratory: Los Alamos, NM, USA, 1973.
2. Cockburn, B.; Shu, C.-W. The Runge-Kutta local projection P1-discontinuous Galerkin finite element method for scalar conservation laws. *Math. Model. Numer. Anal.* **1991**, *25*, 337–361. [CrossRef]
3. Cockburn, B.; Shu, C.-W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II: General framework. *Math. Comput.* **1989**, *52*, 411–435. [CrossRef]
4. Cockburn, B.; Lin, S.; Shu, C.-W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. III: One-dimensional systems. *J. Comput. Phys.* **1989**, *84*, 90–113. [CrossRef]
5. Cockburn, B.; Hou, S.; Shu, C.-W. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV: The multidimensional case. *Math. Comput.* **1990**, *54*, 545–581. [CrossRef]
6. Cockburn, B.; Shu, C.-W. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *J. Comput. Phys.* **1998**, *141*, 199–224. [CrossRef]
7. Hesthaven, J.S.; Warburton, T. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*, 1st ed.; Springer: New York, NY, USA, 2008.
8. Dolejší, V.; Feistauer, M. *Discontinuous Galerkin Method: Analysis and Applications to Compressible Flow*, 1st ed.; Springer: Cham, Switzerland, 2015.
9. Kanevsky, A.; Carpenter, M.H.; Gottlieb, D.; Hesthaven, J.S. Application of implicit-explicit high order Runge-Kutta methods to discontinuous-Galerkin schemes. *J. Comput. Phys.* **2007**, *225*, 1753–1781. [CrossRef]

10. Kennedy, C.A.; Carpenter, M.H. Additive Runge–Kutta schemes for convection–diffusion–reaction equations. *Appl. Numer. Math.* **2003**, *44*, 139–181. [CrossRef]
11. Dolean, V.; Fahs, H.; Fezoui, L.; Lanteri, S. Locally implicit discontinuous Galerkin method for time domain electromagnetics. *J. Comput. Phys.* **2010**, *229*, 512–526. [CrossRef]
12. Hochbruck, M.; Sturm, A. Error analysis of a second-order locally implicit method for linear Maxwell equations. *SIAM J. Numer. Anal.* **2016**, *54*, 3167–3191. [CrossRef]
13. Descombes, S.; Lanteri, S.; Moya, L. Temporal convergence analysis of a locally implicit discontinuous Galerkin time domain method for electromagnetic wave propagation in dispersive media. *J. Comput. App. Math.* **2017**, *316*, 122–132. [CrossRef]
14. Gottlieb, S; Shu, C.-W.; Tadmor, E. Strong stability-preserving high-order time discretization methods. *SIAM Rev.* **2001**, *43*, 89–112. [CrossRef]
15. Gottlieb, S; Shu, C.-W. Total variation diminishing Runge-Kutta schemes. *Math. Comput.* **1998**, *67*, 73–85. [CrossRef]
16. Alexander, R. Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s. *SIAM J. Numer. Anal.* **1977**, *14*, 1006–1021. [CrossRef]
17. Cameron, I.T. Solution of differential-algebraic systems using diagonally implicit Runge-Kutta methods. *IMA J. Numer. Anal.* **1983**, *3*, 273–289. [CrossRef]
18. Skvortsov, L.M. Diagonally implicit Runge–Kutta methods for stiff problems. *Comput. Math. Math. Phys.* **2006**, *46*, 2110–2123. [CrossRef]
19. Li, L.; Wu, S. Construction of Jacobian matrix for solving convection-diffusion problem with interior penalty method. In Proceedings of the 2018 International Conference on Computer Information Science and Application Technology, Daqing, China, 7–9 December 2018. [CrossRef]
20. Qiu, J.; Khoo, C.K; Shu. C.-W. A numerical study for the performance of the Runge-Kutta discontinuous Galerkin method based on different numerical fluxes. *J. Comput. Phys.* **2006**, *212*, 540–565. [CrossRef]
21. Bassi, F.; Rebay, S. A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations. *J. Comput. Phys.* **1997**, *131*, 267–279. [CrossRef]
22. Cockburn, B.; Shu, C.-W. The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM J. Numer. Anal.* **1998**, *35*, 2440–2463. [CrossRef]
23. Peraire, J.; Persson, P.-O. The compact discontinuous Galerkin (CDG) method for elliptic problems. *SIAM J. Sci. Comput.* **2008**, *30*, 1806–1824. [CrossRef]
24. Cheng, J.; Yang, X.; Liu, X.; Liu, T.; Luo, H. A direct discontinuous Galerkin method for the compressible Navier-Stokes equations on arbitrary grids. *J. Comput. Phys.* **2016**, *327*, 484–502. [CrossRef]
25. Persson, P.-O. Shock capturing for high-order discontinuous Galerkin simulation of transient flow problems. In Proceedings of the 21th AIAA Computational Fluid Dynamics Conference, San Diego, CA, USA, 24–27 June 2013. [CrossRef]
26. Dumbser, M.; Zanotti, O.; Loubère, R.; Diot, S. A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *J. Comput. Phys.* **2014**, *278*, 47–75. [CrossRef]
27. Segawa, H.; Luo, H.; Nourgaliev, R. A diagonally implicit Runge-Kutta method for the discontinuous Galerkin solutions of the Navier-Stokes equation. In Proceedings of the 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Orlando, FL, USA, 4–7 January 2011. [CrossRef]
28. Persson, P.-O.; Peraire, J. Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations. *SIAM J. Sci. Comput.* **2008**, *30*, 2709–2733. [CrossRef]
29. Spiteri, R.J.; Ruuth, S.J. A new class of optimal high-order strong-stability-preserving time discretization methods. *SIAM J. Numer. Anal.* **2002**, *40*, 469–491. [CrossRef]
30. Zhang, F; Cheng, J.; Liu, T. A direct discontinuous Galerkin method for the incompressible Navier-Stokes equations on arbitrary grids. *J. Comput. Phys.* **2019**, *380*, 269–294. [CrossRef]
31. Dolejší, V.; Feistauer, M. A semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow. *J. Comput. Phys.* **2004**, *198*, 727–746. [CrossRef]
32. Barter, G.E.; Darmofal, D.L. Shock capturing with PDE-based artificial viscosity for DGFEM: Part I. Formulation. *J. Comput. Phys.* **2010**, *229*, 1810–1827. [CrossRef]

33. Abarbanel, S.S.; Don, W.-S.; Gottlieb, D.; Rudy, D.H.; Townsend, J.C. Secondary frequencies in the wake of a circular cylinder with vortex shedding. *J. Fluid Mech.* **1991**, *225*, 557–574. [CrossRef]
34. Inoue, O.; Hatakeyama, N. Sound generation by a two-dimensional circular cylinder in a uniform flow. *J. Fluid Mech.* **2002**, *471*, 285–314. [CrossRef]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).