



IUScholarWorks at Indiana University South Bend

# Krylov SSP Integrating Factor Runge-Kutta WENO Methods

Chen, Shanqin

To cite this article: Chen, Shanqin. "Krylov SSP integrating factor Runge-Kutta WENO methods." *Mathematics* 9:13 June 24, 2021.

This document has been made available through IUScholarWorks repository, a service of the Indiana University Libraries. Copyrights on documents in IUScholarWorks are held by their respective rights holder(s). Contact [iusw@indiana.edu](mailto:iusw@indiana.edu) for more information.

Article

# Krylov SSP Integrating Factor Runge–Kutta WENO Methods

Shanqin Chen

Department of Mathematical Sciences, Indiana University South Bend, South Bend, IN 46615, USA; chen39@iusb.edu

**Abstract:** Weighted essentially non-oscillatory (WENO) methods are especially efficient for numerically solving nonlinear hyperbolic equations. In order to achieve strong stability and large time-steps, strong stability preserving (SSP) integrating factor (IF) methods were designed in the literature, but the methods there were only for one-dimensional (1D) problems that have a stiff linear component and a non-stiff nonlinear component. In this paper, we extend WENO methods with large time-stepping SSP integrating factor Runge–Kutta time discretization to solve general nonlinear two-dimensional (2D) problems by a splitting method. How to evaluate the matrix exponential operator efficiently is a tremendous challenge when we apply IF temporal discretization for PDEs on high spatial dimensions. In this work, the matrix exponential computation is approximated through the Krylov subspace projection method. Numerical examples are shown to demonstrate the accuracy and large time-step size of the present method.

**Keywords:** strong stability preserving; integrating factor; Runge–Kutta; weighted essentially non-oscillatory methods; Krylov subspace approximation



**Citation:** Chen, S. Krylov SSP Integrating Factor Runge–Kutta WENO Methods. *Mathematics* **2021**, *9*, 1483. <https://doi.org/10.3390/math9131483>

Academic Editor: Athanasios Tzavaras

Received: 12 May 2021  
Accepted: 18 June 2021  
Published: 24 June 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the author. 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 (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

We consider the two-dimensional general form of the nonlinear hyperbolic partial differential equation (PDE)

$$u_t + f(u)_x + g(u)_y = 0, \quad (1)$$

where  $u$  is the unknown function,  $f$  is the flux function in  $x$  spatial direction, and  $g$  is the flux function in  $y$  spatial direction. Nonlinear hyperbolic PDEs (1) are common mathematical models in applications from science and engineering. To numerically solve this time-dependent problem (1), we first need to choose an efficient nonlinearly stable spatial discretization. The solutions to hyperbolic PDEs may have discontinuities, sharp gradients, or discontinuous derivatives, and so forth. A straightforward high-order scheme for such kinds of problems will generate instability called the Gibbs phenomena. Thus, we desire schemes that not only have a uniformly high order of accuracy in smooth regions of the solution, but can also resolve singularities in an accurate and essentially nonoscillatory (ENO) fashion. Weighted essentially nonoscillatory (WENO) schemes are a popular class of such schemes. WENO schemes [1,2] were designed based on the successful ENO schemes in [3–5] with additional advantages. For problems with physically complicated solution structures, such as turbulence systems in fluid dynamics, WENO methods with very high-order accuracy and refined computational grids can resolve the physical solution with mixed stable and unstable behavior [6]. The essence of the WENO schemes is to reconstruct a weighted combination of some local reconstructions based on different small stencils. The combination coefficients, which are also called nonlinear weights, are determined by the linear weights and the smoothness indicators. The linear weights are usually selected to increase the order of accuracy on each small stencil, and the smoothness indicators measure the smoothness of the reconstructed function in the relevant small stencils. Since they were developed, WENO schemes have been studied and applied extensively for solving problems with both singularities and a complex solution structure. Here are some examples. Fast sweeping WENO schemes and homotopy WENO schemes were designed in [7–12]

to efficiently solve hyperbolic PDEs' steady-state problems. To solve stiff convection-diffusion-reaction PDEs, high-order Krylov implicit integration factor WENO schemes were developed in [13–16]. In [17], sparse grid WENO methods were developed to solve high-dimensional hyperbolic PDEs.

Since WENO schemes were designed to handle problems with both complicated solution structures and discontinuities, they require more operations than many other schemes because of their sophisticated nonlinear properties and high-order accuracy. The computational cost rises greatly when the PDEs are on 2D or higher, or the number of grid points is large. Especially for solving long time simulations or solving steady-state problems, efficient large time-step sizes and high-order temporal discretization methods are crucial. Many innovative high-order temporal schemes were designed, such as the total variation diminishing (TVD) Runge–Kutta (RK) schemes [4,18–20], spectral deferred correction (SDC) methods [21–25], high-order implicit–explicit (IMEX) multistep/RK methods [26–33] and their applications, hybrid methods of SDC and high-order RK schemes [34], SSP multistage two-derivative time-stepping schemes [35], and so forth. All of these numerical methods have their own advantages.

Challenges in designing large time-step size schemes for (1) also come from restrictions on local time-step sizes due to both linear stability and nonlinear stability. Since discontinuous solutions often arise in hyperbolic PDEs, the linear stability theory itself does not guarantee convergence. SSP temporal numerical schemes were developed [4,36] to assure the preservation of the nonlinear stability properties. That is, if the high-order spatial method (for example, a WENO scheme) coupled with forward Euler satisfies the nonlinear stability, then the same spatial discretization coupled with these higher-order time discretization methods will preserve the nonlinear stability. Exponential integrators such as integrating factor methods are time schemes which have large linear stability regions. They can achieve very large time-step sizes or Courant–Friedrichs–Lewy (CFL) numbers for problems with smooth solutions [13–15,37]. However, these schemes do not satisfy the SSP property. Hence, direct applications of these schemes to hyperbolic PDEs with discontinuous solutions may lead to nonlinear instability. Recently, integrating factor Runge–Kutta (IFRK) methods with a SSP property were developed in [38–40] to solve one-dimensional hyperbolic PDEs, which have explicit linear and nonlinear parts. These methods are exponential integrators which satisfy the SSP requirement. They can completely relax the time-step size restriction coming from the linear part of the high-order spatial discretization. However, the question of how to apply the method to multi-dimensional problems, which do not have explicit linear and nonlinear parts, is still open.

In this paper, we extend WENO methods with large time-stepping SSP integrating factor Runge–Kutta time discretization to solve general nonlinear multi-dimensional problems. When applying the integrating factor methods to multi-dimensional problems, evaluating the product of the matrix exponential and a vector efficiently is crucial and challenging. To overcome this difficulty, we utilize the Krylov subspace technique to approximately calculate such products. In [41], Gallopoulos and Saad used and analyzed the Krylov subspace methods to approximate the product of an exponential matrix and a vector. In [42,43], we applied the Krylov subspace method to implicit integration factor methods to solve high-dimensional reaction–diffusion PDEs. To solve hyperbolic PDEs, in this paper, we first use third- or fifth-order WENO schemes for the spatial discretization. In order to apply exponential integrators to the resulting nonlinear ODE system, we split it into linear and nonlinear parts. Then, for the time discretization, we combine the SSP IFRK methods [38] with the Krylov subspace approximation method to solve two-dimensional general nonlinear hyperbolic problems.

The organization of the paper is as follows. WENO methods with the Krylov SSP IFRK methods are derived and formulated in Section 2. The stability region for our schemes are analyzed in Section 3. In Section 4, we present some numerical experiments. Discussions and conclusions are given in Section 5.

## 2. Krylov SSP Integrating Factor Runge–Kutta Methods

Consider two-dimensional hyperbolic conservation laws (1) with appropriate boundary conditions. Since high-order finite difference WENO methods are very efficient in resolving complex solution structures and keeping nonlinear stability for multi-dimensional problems due to its dimension-by-dimension implementation fashion [6], it is natural to apply high-order finite difference WENO schemes in the spatial discretization of (1).

### 2.1. Spatial Discretization: The Third and Fifth-Order WENO Schemes

We use a finite difference scheme to approximate the values of the hyperbolic terms  $f(u)_x + g(u)_y$  at a grid  $(x_i, y_j)$  in a conservative fashion. To be specific, we use a conservative flux difference

$$f(u)_x|_{x=x_i} \approx \frac{1}{\Delta x} (\hat{f}_{i+1/2} - \hat{f}_{i-1/2}), \tag{2}$$

to approximate  $f(u)_x$  at  $(x_i, y_j)$  along the line  $y = y_j$ . Here,

$$\hat{f}_{i+\frac{1}{2}} = \hat{f}_{i+\frac{1}{2}}^+ + \hat{f}_{i+\frac{1}{2}}^-. \tag{3}$$

In the WENO schemes, the following *Lax-Friedrichs flux splitting* is used.

$$f^+(u) = \frac{1}{2}(f(u) + \alpha_x u), \quad f^-(u) = \frac{1}{2}(f(u) - \alpha_x u), \tag{4}$$

where

$$\alpha_x = \max_u |f'(u)|. \tag{5}$$

$f^+(u)$  and  $f^-(u)$  are the positive and negative wind parts, respectively. Note that, for simplicity, we omit the index  $j$  for the  $y$  direction when the approximations for the  $x$  direction are described.

Let's first look at the third-order WENO (WENO3) method formulation. The stencil (in the 2D case) is given on the left of Figure 1. The numerical flux  $\hat{f}_{i+1/2}^+$  depends on a left-biased stencil, namely, the value  $f(u_k)$  at three nodes  $k = i - 1, i, i + 1$ , because the wind is positive; that is,  $f'(u) \geq 0$  for the scalar case. Its formula is given by

$$\hat{f}_{i+1/2}^+ = \omega_0 \left[ \frac{1}{2}f^+(u_i) + \frac{1}{2}f^+(u_{i+1}) \right] + \omega_1 \left[ -\frac{1}{2}f^+(u_{i-1}) + \frac{3}{2}f^+(u_i) \right], \tag{6}$$

where

$$\omega_r = \frac{\alpha_r}{\alpha_1 + \alpha_2}, \quad \alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2}, \quad r = 0, 1. \tag{7}$$

Here,  $d_0 = \frac{2}{3}$  and  $d_1 = \frac{1}{3}$  are called the *linear weights*, and  $\beta_0 = (f^+(u_{i+1}) - f^+(u_i))^2$ ,  $\beta_1 = (f^+(u_i) - f^+(u_{i-1}))^2$  are called the *smoothness indicators*. To prevent a zero denominator from happening, we choose  $\epsilon$  as a tiny positive number. From the Formula (6), we can see that  $\hat{f}_{i+1/2}^+$  is a convex combination of  $\left[ \frac{1}{2}f^+(u_i) + \frac{1}{2}f^+(u_{i+1}) \right]$  and  $\left[ -\frac{1}{2}f^+(u_{i-1}) + \frac{3}{2}f^+(u_i) \right]$ , which are both second-order numerical fluxes and are built on two substencils,  $(x_i, x_{i+1})$  and  $(x_{i-1}, x_i)$ , respectively. The coefficients  $\omega_i$  are determined by  $\beta_i$  in each substencil.

For  $\hat{f}_{i+1/2}^-$ , this is the case when  $f'(u) < 0$ , or a negative wind. A WENO3 scheme can be constructed by using a right-biased stencil with values  $f(u_i), f(u_{i+1}),$  and  $f(u_{i+2})$ .

$$\hat{f}_{i+1/2}^- = \tilde{\omega}_0 \left[ \frac{3}{2}f^-(u_{i+1}) - \frac{1}{2}f^-(u_{i+2}) \right] + \tilde{\omega}_1 \left[ \frac{1}{2}f^-(u_i) + \frac{1}{2}f^-(u_{i+1}) \right], \tag{8}$$

where

$$\tilde{\omega}_r = \frac{\tilde{\alpha}_r}{\tilde{\alpha}_1 + \tilde{\alpha}_2}, \quad \tilde{\alpha}_r = \frac{\tilde{d}_r}{(\epsilon + \tilde{\beta}_r)^2}, \quad r = 0, 1, \tag{9}$$

with  $\tilde{d}_0 = \frac{1}{3}$ ,  $\tilde{d}_1 = \frac{2}{3}$ , and  $\tilde{\beta}_0 = (f^-(u_{i+2}) - f^-(u_{i+1}))^2$ ,  $\tilde{\beta}_1 = (f^-(u_{i+1}) - f^-(u_i))^2$ .

For the fifth-order WENO scheme (WENO5), the stencil is shown on the right of Figure 1. The numerical flux  $\hat{f}_{i+1/2}^+$  depends on a left-biased stencil, namely, the five-point value  $f(u_k), k = i - 2, i - 1, i, i + 1, i + 2$ . Its formula is given by

$$\begin{aligned} \hat{f}_{i+1/2}^+ &= \omega_0 \left[ \frac{1}{3}f^+(u_i) + \frac{5}{6}f^+(u_{i+1}) - \frac{1}{6}f^+(u_{i+2}) \right] \\ &+ \omega_1 \left[ -\frac{1}{6}f^+(u_{i-1}) + \frac{5}{6}f^+(u_i) + \frac{1}{3}f^+(u_{i+1}) \right] \\ &+ \omega_2 \left[ \frac{1}{3}f^+(u_{i-2}) - \frac{7}{6}f^+(u_{i-1}) + \frac{11}{6}f^+(u_i) \right], \end{aligned} \tag{10}$$

where

$$\omega_r = \frac{\alpha_r}{\alpha_0 + \alpha_1 + \alpha_2}, \quad \alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2}, \quad r = 0, 1, 2, \tag{11}$$

with  $d_0 = \frac{3}{10}, d_1 = \frac{3}{5}$ , and  $d_2 = \frac{1}{10}$ , and

$$\beta_0 = \frac{13}{12}(f^+(u_i) - 2f^+(u_{i+1}) + f^+(u_{i+2}))^2 + \frac{1}{4}(3f^+(u_i) - 4f^+(u_{i+1}) + f^+(u_{i+2}))^2,$$

$$\beta_1 = \frac{13}{12}(f^+(u_{i-1}) - 2f^+(u_i) + f^+(u_{i+1}))^2 + \frac{1}{4}(f^+(u_{i-1}) - f^+(u_{i+1}))^2,$$

$$\beta_2 = \frac{13}{12}(f^+(u_{i-2}) - 2f^+(u_{i-1}) + f^+(u_i))^2 + \frac{1}{4}(f^+(u_{i-2}) - 4f^+(u_{i-1}) + 3f^+(u_i))^2.$$

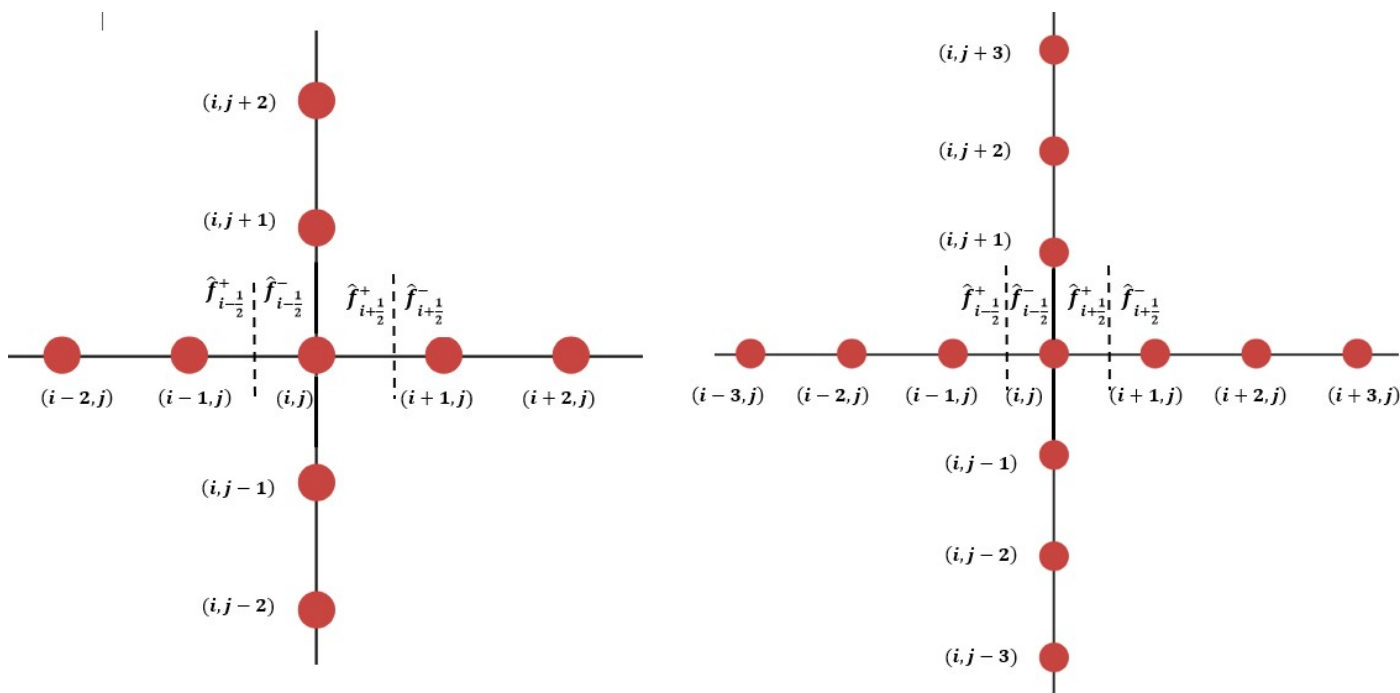


Figure 1. Stencils for the third-order WENO (left) and the fifth-order WENO (right).

From the Formula (10), we can see that this numerical flux  $\hat{f}_{1+1/2}^+$  is a convex combination of three brackets, and each one of them is a third-order numerical flux and is built on three different substencils of three points each. The coefficients  $\omega_i$  are determined by  $\beta_i$  in each substencil.

For the case of negative wind, that is,  $f'(u) < 0$ , a fifth-order WENO approximation to  $\hat{f}_{i+1/2}^-$  can be constructed by using a right-biased stencil with numerical values  $f(u_{i-1}), f(u_i), f(u_{i+1}), f(u_{i+2})$ , and  $f(u_{i+3})$ .

$$\begin{aligned} \hat{f}_{i+1/2}^- &= \tilde{\omega}_0 \left[ \frac{11}{6} f^-(u_{i+1}) - \frac{7}{6} f^-(u_{i+2}) + \frac{1}{3} f^-(u_{i+3}) \right] \\ &+ \tilde{\omega}_1 \left[ \frac{1}{3} f^-(u_i) + \frac{5}{6} f^-(u_{i+1}) - \frac{1}{6} f^-(u_{i+2}) \right] \\ &+ \tilde{\omega}_2 \left[ -\frac{1}{6} f^-(u_{i-1}) + \frac{5}{6} f^-(u_i) + \frac{1}{3} f^-(u_{i+1}) \right], \end{aligned} \tag{12}$$

where

$$\tilde{\omega}_r = \frac{\tilde{\alpha}_r}{\tilde{\alpha}_0 + \tilde{\alpha}_1 + \tilde{\alpha}_2}, \quad \tilde{\alpha}_r = \frac{\tilde{d}_r}{(\epsilon + \tilde{\beta}_r)^2}, \quad r = 0, 1, 2, \tag{13}$$

with  $\tilde{d}_0 = \frac{1}{10}$ ,  $\tilde{d}_1 = \frac{3}{5}$ , and  $\tilde{d}_2 = \frac{3}{10}$ , and

$$\begin{aligned} \tilde{\beta}_0 &= \frac{13}{12} (f^-(u_{i+1}) - 2f^-(u_{i+2}) + f^-(u_{i+3}))^2 + \frac{1}{4} (3f^-(u_{i+1}) - 4f^-(u_{i+2}) + f^-(u_{i+3}))^2, \\ \tilde{\beta}_1 &= \frac{13}{12} (f^-(u_i) - 2f^-(u_{i+1}) + f^-(u_{i+2}))^2 + \frac{1}{4} (f^-(u_i) - f^-(u_{i+2}))^2, \\ \tilde{\beta}_2 &= \frac{13}{12} (f^-(u_{i-1}) - 2f^-(u_i) + f^-(u_{i+1}))^2 + \frac{1}{4} (f^-(u_{i-1}) - 4f^-(u_i) + 3f^-(u_{i+1}))^2. \end{aligned}$$

Combining Equations (3), (6), and (8), we have the numerical flux for the third-order WENO method

$$\begin{aligned} \hat{f}_{i+1/2}(u_{i-1}, u_i, u_{i+1}, u_{i+2}) &= \omega_0 \left[ \frac{1}{2} f^+(u_i) + \frac{1}{2} f^+(u_{i+1}) \right] + \omega_1 \left[ -\frac{1}{2} f^+(u_{i-1}) + \frac{3}{2} f^+(u_i) \right] \\ &+ \tilde{\omega}_0 \left[ \frac{3}{2} f^-(u_{i+1}) - \frac{1}{2} f^-(u_{i+2}) \right] + \tilde{\omega}_1 \left[ \frac{1}{2} f^-(u_i) + \frac{1}{2} f^-(u_{i+1}) \right]. \end{aligned} \tag{14}$$

Combining Equations (3), (10), (12), and using the following notations

$$\begin{aligned} W0_i &= \frac{1}{3} f(u_i) + \frac{5}{6} f(u_{i+1}) - \frac{1}{6} f(u_{i+2}) \\ W1_i &= -\frac{1}{6} f(u_{i-1}) + \frac{5}{6} f(u_i) + \frac{1}{3} f(u_{i+1}) \\ W2_i &= \frac{1}{3} f(u_{i-2}) - \frac{7}{6} f(u_{i-1}) + \frac{11}{6} f(u_i) \\ \tilde{W}0_i &= \frac{11}{6} f(u_{i+1}) - \frac{7}{6} f(u_{i+2}) + \frac{1}{3} f(u_{i+3}), \end{aligned} \tag{15}$$

we have the numerical flux for the fifth-order WENO method

$$\begin{aligned} \hat{f}_{i+1/2}(u_{i-2}, u_{i-1}, u_i, u_{i+1}, u_{i+2}, u_{i+3}) &= \omega_0 W0_i^+ + \omega_1 W1_i^+ + \omega_2 W2_i^+ \\ &+ \tilde{\omega}_0 \tilde{W}0_i^- + \tilde{\omega}_1 W0_i^- + \tilde{\omega}_2 W1_i^-. \end{aligned} \tag{16}$$

Here,  $W0_i^+$  means that  $f^+$  is used for  $f$  in (15), and  $W0_i^-$  means that  $f^-$  is used for  $f$  in (15), and so on.

We apply the similar procedures in the  $y$  direction to approximate the numerical flux of  $g(u)_y$ , then we have a nonlinear system

$$(u_{i,j})_t = -\frac{\hat{f}_{i+1/2,j} - \hat{f}_{i-1/2,j}}{\Delta x} - \frac{\hat{g}_{i,j+1/2} - \hat{g}_{i,j-1/2}}{\Delta y}, \quad i = 1, \dots, n; j = 1, \dots, m, \tag{17}$$

where  $n$  and  $m$  are the numbers of grid points in  $x$  and  $y$  spatial directions, respectively.

### 2.2. Temporal Scheme: SSP Integrating Factor Runge–Kutta Schemes

After the spacial discretizations, a semi-discretized nonlinear ODE system (17) is obtained. We write it in the vector form

$$\vec{U}_t = \vec{F}(\vec{U}). \tag{18}$$

Here,  $\vec{U}$  is the vector of the solution variables, and  $\vec{F}(\vec{U})$  is the approximation for the nonlinear convection terms by high WENO schemes. Each component of  $\vec{F}(\vec{U})$  is a nonlinear function of numerical solution values on approximation stencils of WENO schemes. In order to apply exponential integrators to the nonlinear system (18), we factor out the linear part of the system as was done in [13] for the fully nonlinear convection-diffusion PDEs. Specifically, at every time-step interval  $[t_k, t_{k+1}]$  with time-step size  $\Delta t = t_{k+1} - t_k$ , we perform the splitting:

$$\vec{U}_t = \vec{F}(\vec{U}) = C_k \vec{U} + \vec{F}_N, \tag{19}$$

where  $C_k \vec{U}$  is the linear part with

$$C_k = \frac{\partial \vec{F}}{\partial \vec{U}}(\vec{U}^k) \tag{20}$$

being the Jacobian matrix (or an approximation of the Jacobian matrix if its direct computation is difficult) and  $\vec{U}^k$  being the numerical solution of  $\vec{U}(t_k)$ .  $\vec{F}_N$  is the nonlinear remainder of the system (18) after the splitting

$$\vec{F}_N = \vec{F}(\vec{U}) - C_k \vec{U}.$$

For example, for 1D problems, if  $c_{ij}$  is the  $i$ th row and  $j$ th column element of the Jacobian matrix  $C_k$ , then  $c_{ij} = \frac{1}{\Delta x} \left( \frac{\partial \hat{f}_{i+1/2}}{\partial u_j} - \frac{\partial \hat{f}_{i-1/2}}{\partial u_j} \right)$  based on the 1D version of (17).

For the third-order WENO method, from Equation (14), we have

$$\begin{aligned} \frac{\partial \hat{f}_{i+1/2}}{\partial u_{i-1}} &= \frac{\partial \omega_0}{\partial u_{i-1}} \left( \frac{1}{2} f^+(u_i) + \frac{1}{2} f^+(u_{i+1}) \right) + \frac{\partial \omega_1}{\partial u_{i-1}} \left( -\frac{1}{2} f^+(u_{i-1}) + \frac{3}{2} f^+(u_i) \right) \\ &\quad - \frac{1}{4} \omega_1 (f'(u_{i-1}) + \alpha_x), \end{aligned}$$

$$\begin{aligned} \frac{\partial \hat{f}_{i+1/2}}{\partial u_i} &= \frac{\partial \omega_0}{\partial u_i} \left( \frac{1}{2} f^+(u_i) + \frac{1}{2} f^+(u_{i+1}) \right) + \frac{1}{4} \omega_0 (f'(u_i) + \alpha_x) \\ &\quad + \frac{\partial \omega_1}{\partial u_i} \left( -\frac{1}{2} f^+(u_{i-1}) + \frac{3}{2} f^+(u_i) \right) + \frac{3}{4} \omega_1 (f'(u_i) + \alpha_x) \\ &\quad + \frac{\partial \tilde{\omega}_0}{\partial u_i} \left( \frac{3}{2} f^-(u_{i+1}) - \frac{1}{2} f^-(u_{i+2}) \right) + \frac{\partial \tilde{\omega}_1}{\partial u_i} \left( \frac{1}{2} f^-(u_i) + \frac{1}{2} f^-(u_{i+1}) \right) \\ &\quad + \frac{1}{4} \tilde{\omega}_1 (f'(u_i) - \alpha_x), \end{aligned}$$

$$\begin{aligned} \frac{\partial \hat{f}_{i+1/2}}{\partial u_{i+1}} &= \frac{\partial \omega_0}{\partial u_{i+1}} \left( \frac{1}{2} f^+(u_i) + \frac{1}{2} f^+(u_{i+1}) \right) + \frac{1}{4} \omega_0 (f'(u_{i+1}) + \alpha_x) \\ &\quad + \frac{\partial \omega_1}{\partial u_{i+1}} \left( -\frac{1}{2} f^+(u_{i-1}) + \frac{3}{2} f^+(u_i) \right) \\ &\quad + \frac{\partial \tilde{\omega}_0}{\partial u_{i+1}} \left( \frac{3}{2} f^-(u_{i+1}) - \frac{1}{2} f^-(u_{i+2}) \right) + \frac{3}{4} \tilde{\omega}_0 (f'(u_{i+1}) - \alpha_x) \\ &\quad + \frac{\partial \tilde{\omega}_1}{\partial u_{i+1}} \left( \frac{1}{2} f^-(u_i) + \frac{1}{2} f^-(u_{i+1}) \right) + \frac{1}{4} \tilde{\omega}_1 (f'(u_{i+1}) - \alpha_x), \end{aligned}$$

$$\begin{aligned} \frac{\partial \hat{f}_{i+\frac{1}{2}}}{\partial u_{i+2}} &= \frac{\partial \tilde{\omega}_0}{\partial u_{i+2}} \left( \frac{3}{2} f^-(u_{i+1}) - \frac{1}{2} f^-(u_{i+2}) \right) - \frac{1}{4} \tilde{\omega}_0 (f'(u_{i+2}) - \alpha_x) \\ &\quad + \frac{\partial \tilde{\omega}_1}{\partial u_{i+2}} \left( \frac{1}{2} f^-(u_i) + \frac{1}{2} f^-(u_{i+1}) \right), \end{aligned}$$

where  $\frac{\partial \omega_r}{\partial u_j}$  and  $\frac{\partial \tilde{\omega}_r}{\partial u_j}$ ,  $r = 0, 1$  can be easily obtained based on the definitions of (7) and (9).

For all  $j$  other than  $j = i - 1, i, i + 1$ , and  $i + 2$ , we have  $\frac{\partial \hat{f}_{i+1/2}}{\partial u_j} = 0$ .

For the fifth-order WENO method, from Equation (16) we have

$$\begin{aligned} \frac{\partial \hat{f}_{i+\frac{1}{2}}}{\partial u_{i-2}} &= \frac{\partial \omega_0}{\partial u_{i-2}} W0_i^+ + \frac{\partial \omega_1}{\partial u_{i-2}} W1_i^+ + \frac{\partial \omega_2}{\partial u_{i-2}} W2_i^+ + \frac{1}{6} \omega_2 (f'(u_{i-2}) + \alpha_x), \\ \frac{\partial \hat{f}_{i+\frac{1}{2}}}{\partial u_{i-1}} &= \frac{\partial \omega_0}{\partial u_{i-1}} W0_i^+ + \frac{\partial \omega_1}{\partial u_{i-1}} W1_i^+ - \frac{1}{12} \omega_1 (f'(u_{i-1}) + \alpha_x) + \frac{\partial \omega_2}{\partial u_{i-1}} W2_i^+ \\ &\quad - \frac{7}{12} \omega_2 (f'(u_{i-1}) + \alpha_x) + \frac{\partial \tilde{\omega}_0}{\partial u_{i-1}} \tilde{W}0_i^- + \frac{\partial \tilde{\omega}_1}{\partial u_{i-1}} W0_i^- + \frac{\partial \tilde{\omega}_2}{\partial u_{i-1}} W1_i^- \\ &\quad - \frac{1}{12} \tilde{\omega}_2 (f'(u_{i-1}) + \alpha_x), \\ \frac{\partial \hat{f}_{i+\frac{1}{2}}}{\partial u_i} &= \frac{\partial \omega_0}{\partial u_i} W0_i^+ + \frac{1}{6} \omega_0 (f'(u_i) + \alpha_x) + \frac{\partial \omega_1}{\partial u_i} W1_i^+ + \frac{5}{12} \omega_1 (f'(u_i) + \alpha_x) + \frac{\partial \omega_2}{\partial u_i} W2_i^+ \\ &\quad + \frac{11}{12} \omega_2 (f'(u_i) + \alpha_x) + \frac{\partial \tilde{\omega}_0}{\partial u_i} \tilde{W}0_i^- + \frac{\partial \tilde{\omega}_1}{\partial u_i} W0_i^- + \frac{1}{6} \tilde{\omega}_1 (f'(u_i) - \alpha_x) + \frac{\partial \tilde{\omega}_2}{\partial u_i} W1_i^- \\ &\quad + \frac{5}{12} \tilde{\omega}_2 (f'(u_i) - \alpha_x), \\ \frac{\partial \hat{f}_{i+\frac{1}{2}}}{\partial u_{i+1}} &= \frac{\partial \omega_0}{\partial u_{i+1}} W0_i^+ + \frac{5}{12} \omega_0 (f'(u_{i+1}) + \alpha_x) + \frac{\partial \omega_1}{\partial u_{i+1}} W1_i^+ + \frac{1}{6} \omega_1 (f'(u_{i+1}) + \alpha_x) \\ &\quad + \frac{\partial \omega_2}{\partial u_{i+1}} W2_i^+ + \frac{\partial \tilde{\omega}_0}{\partial u_{i+1}} \tilde{W}0_i^- + \frac{11}{12} \tilde{\omega}_0 (f'(u_{i+1}) - \alpha_x) + \frac{\partial \tilde{\omega}_1}{\partial u_{i+1}} W0_i^- \\ &\quad + \frac{5}{12} \tilde{\omega}_1 (f'(u_{i+1}) - \alpha_x) + \frac{\partial \tilde{\omega}_2}{\partial u_{i+1}} W1_i^- + \frac{1}{6} \tilde{\omega}_2 (f'(u_{i+1}) - \alpha_x), \\ \frac{\partial \hat{f}_{i+\frac{1}{2}}}{\partial u_{i+2}} &= \frac{\partial \omega_0}{\partial u_{i+2}} W0_i^+ - \frac{1}{12} \omega_0 (f'(u_{i+2}) + \alpha_x) + \frac{\partial \omega_1}{\partial u_{i+2}} W1_i^+ + \frac{\partial \omega_2}{\partial u_{i+2}} W2_i^+ \\ &\quad + \frac{\partial \tilde{\omega}_0}{\partial u_{i+2}} \tilde{W}0_i^- - \frac{7}{12} \tilde{\omega}_0 (f'(u_{i+2}) - \alpha_x) + \frac{\partial \tilde{\omega}_1}{\partial u_{i+2}} W0_i^- \\ &\quad - \frac{1}{12} \tilde{\omega}_1 (f'(u_{i+2}) - \alpha_x) + \frac{\partial \tilde{\omega}_2}{\partial u_{i+2}} W1_i^-, \\ \frac{\partial \hat{f}_{i+\frac{1}{2}}}{\partial u_{i+3}} &= \frac{\partial \tilde{\omega}_0}{\partial u_{i+3}} \tilde{W}0_i^- + \frac{1}{6} \tilde{\omega}_0 (f'(u_{i+3}) - \alpha_x) + \frac{\partial \tilde{\omega}_1}{\partial u_{i+3}} W0_i^- + \frac{\partial \tilde{\omega}_2}{\partial u_{i+3}} W1_i^-, \end{aligned}$$

where  $\frac{\partial \omega_r}{\partial u_j}$  and  $\frac{\partial \tilde{\omega}_r}{\partial u_j}$ ,  $r = 0, 1, 2$  can be easily obtained based on the definitions of (11) and (13).

For all  $j$  except  $j = i - 2, i - 1, i, i + 1, i + 2$ , and  $i + 3$ , we have  $\frac{\partial \hat{f}_{i+1/2}}{\partial u_j} = 0$ .

**Remark 1.**  $C_k$  is a  $n \times n$  sparse matrix for a one-dimensional problem. For two-dimensional problems, the dimension-by-dimension procedure is followed. A  $nm \times nm$  sparse matrix  $C_k$  is obtained. However, we only store the nonzero elements and their locations in the matrix  $C_k$  for the Krylov approximation procedure.



Then, for system (19), we apply the integration factor approach in [13,14], with  $e^{-C_k t}$  as the integration factor, and the resulting integration factor equations of the system is

$$\frac{d(e^{-C_k t} \vec{U})}{dt} = e^{-C_k t} \vec{F}_N(\vec{U}). \tag{21}$$

Denoting  $\vec{Y} = e^{-C_k t} \vec{U}$ , then (21) can be written in terms of  $\vec{Y}$  as

$$\frac{d\vec{Y}}{dt} = e^{-C_k t} \vec{F}_N(e^{C_k t} \vec{Y}). \tag{22}$$

The next step is to apply Runge–Kutta methods to (22) to approximate  $\vec{Y}(t)$ ; then the relation  $\vec{U} = e^{C_k t} \vec{Y}$  is used to obtain the exponential integrator time schemes for  $\vec{U}$ . To obtain the SSP exponential integrator, we apply the method in [38]. As pointed out in [38], application of a SSP Runge–Kutta method to derive an exponential integrator time scheme is not enough to ensure the preservation of the SSP property. In order to obtain SSP exponential integrator schemes for  $\vec{U}$  based on SSP Runge–Kutta methods, the system (22) must be solved by SSP Runge–Kutta methods with non-decreasing abscissas. For example, the third-order SSP Runge–Kutta method with non-decreasing abscissas is used to solve (22) and leads to the following third-order SSP exponential integrator:

$$\begin{aligned} \vec{U}^{(1)} &= \frac{1}{2} e^{\frac{2}{3} C_k \Delta t} \left( 2\vec{U}^k + \frac{4}{3} \Delta t \vec{F}_N(\vec{U}^k) \right), \\ \vec{U}^{(2)} &= \frac{2}{3} e^{\frac{2}{3} C_k \Delta t} \vec{U}^k + \frac{1}{3} \left( \vec{U}^{(1)} + \frac{4}{3} \Delta t \vec{F}_N(\vec{U}^{(1)}) \right), \\ \vec{U}^{k+1} &= \frac{59}{128} e^{C_k \Delta t} \vec{U}^k + \frac{15}{128} e^{C_k \Delta t} \left( \vec{U}^k + \frac{4}{3} \Delta t \vec{F}_N(\vec{U}^k) \right) \\ &\quad + \frac{27}{64} e^{\frac{1}{3} C_k \Delta t} \left( \vec{U}^{(2)} + \frac{4}{3} \Delta t \vec{F}_N(\vec{U}^{(2)}) \right). \end{aligned} \tag{23}$$

### 2.3. Integrating Factor Methods Based on Krylov Subspace Approximation

The efficiency of the SSP IF Runge–Kutta methods such as (23) largely relies on the methods to compute the product of the matrix exponential and a vector,  $e^{C_k \Delta t} \vec{U}^k$ . When we use a specific spatial discretization to solve a PDE, a large matrix  $C_k$  such as (20) will be generated. Although  $C_k$  is sparse, the exponential matrix  $e^{C_k \Delta t}$  is not. For one-dimensional problems,  $e^{C_k \Delta t}$  can be directly computed, since the size of the matrix  $C_k$  is manageable. However, for 2D problems, as an illustration, if we use a finite difference method on a  $n \times n$  rectangular mesh, then we will obtain matrices  $C_k$  with size  $n^2 \times n^2$ . Therefore, it cannot be afforded to compute and store  $e^{C_k \Delta t}$  directly with regard to both computational cost and computer storage, due to the huge size of the matrix  $C_k$ . Fortunately, if we take a close look at (23), we will see that we do not need  $e^{C_k \Delta t}$  itself—all we need are the products of  $e^{C_k \Delta t}$  and some vectors in (23). An outstanding option for doing this job is the Krylov subspace approximation, which can provide both accuracy and efficiency.

Now, we implement the Krylov subspace technique to find approximations to the products of some exponential matrices and vectors in the SSP IF Runge–Kutta scheme (23) and derive the new Krylov SSP IF Runge–Kutta methods. First, we review the procedure of the Krylov subspace methods to approximate  $e^{A \Delta t} v$ , following the literature (e.g., [41,44]).

First, we form a Krylov subspace  $K_M$  based on the large sparse matrix  $A$  and the vector  $v$  in the following way

$$K_M = \text{span}\{v, Av, A^2 v, \dots, A^{M-1} v\}. \tag{24}$$

Here,  $M$  is the dimension of the subspace  $K_M$ , and it is much smaller than the size of  $A$ . We denote  $V_M = [v_1, v_2, v_3, \dots, v_M]$  as an orthonormal basis of the Krylov subspace  $K_M$ , which is generated by the Arnoldi algorithm [45] in the steps described below (Algorithm 1).

**Algorithm 1:** Arnoldi algorithm

---

Step 1. Calculate the initial vector:  $v_1 = v / \|v\|_2$ .  
 Step 2. For  $j = 1, 2, \dots, M$  :  
 Step 2.1 Calculate the vector  $w = Av_j$ .  
 Step 2.2 For  $i = 1, 2, \dots, j$ , compute  
     (1) the inner product  $h_{i,j} = (w, v_i)$ .  
     (2)  $w = w - h_{i,j}v_i$ .  
 Step 2.3 Calculate  $h_{j+1,j} = \|w\|_2$ .  
 Step 2.4 If  $h_{j+1,j} = 0$ , then  
     Set  $M = j$ ;  
     Stop and exit;  
 else  
     Calculate  $v_{j+1} = w/h_{j+1,j}$ .

---

The vectors  $h_{i,j}$  produced from the above Arnoldi algorithm consist of an  $M \times M$  upper-Hessenberg matrix, denoted by  $H_M$ . Because the vectors in  $V_M$  are orthogonal, it is easy to obtain

$$H_M = V_M^T A V_M. \quad (25)$$

The above equation shows that  $H_M$  is the projection of  $A$  to the Krylov subspace  $K_M$ , with respect to the basis  $V_M$ . Note that the dimension of  $H_M$  is  $M$ , which is much smaller than that of the matrix  $A$ . What's more, because  $V_M$  is orthonormal, the vector  $V_M V_M^T e^{A\Delta t} v$  is the orthogonal projection of  $e^{A\Delta t} v$  on  $K_M$ , that is,  $V_M V_M^T e^{A\Delta t} v$  is the best approximation to  $e^{A\Delta t} v$  from  $K_M$ . Thus, we have

$$e^{A\Delta t} v \approx V_M V_M^T e^{A\Delta t} v = \|v\|_2 V_M V_M^T e^{A\Delta t} v_1 = \|v\|_2 V_M V_M^T e^{A\Delta t} V_M e_1, \quad (26)$$

where  $e_1$  is the first column of the identity matrix  $I_M$  with size  $M \times M$ . From (25), we obtain the approximation

$$e^{A\Delta t} v \approx \|v\|_2 V_M e^{H_M \Delta t} e_1. \quad (27)$$

Therefore, the large  $e^{A\Delta t}$  matrix exponential computation is substituted by a small  $e^{H_M \Delta t}$  matrix exponential computation. The much smaller  $e^{H_M \Delta t}$  problem can be calculated by a scaling and squaring algorithm with a Padé approximation. The computational cost of this algorithm is only  $O(M^2)$ , see [41,44,46]. After implementing the Krylov subspace approximation (27) to (23), we obtain the Krylov SSP IF Runge–Kutta scheme.

We choose the value of  $M$  to be small, but still large enough to ensure that the truncation errors of the numerical methods (23) are much larger than the errors of Krylov subspace approximations. In this paper, we use  $M = 25$ . From our numerical experiments, we observe that the proposed Krylov SSP IF RK methods have already achieved clear accuracy orders for such a small value of  $M$ . What's more, we do not need to increase the size of  $M$  when we refine the spatial–temporal resolution.

### 3. Linear Stability Analysis

To discuss the linear stability of (23), we consider the test model problem

$$u_t = Lu + Nu, \quad (28)$$

where  $Lu$  is the linear part, and  $Nu$  is the linearization of the nonlinear part  $\vec{F}_N$  in (19). Based on applying the third-order SSP IF Runge–Kutta method (23) to the test problem (28), we plot the stability region boundaries, which are some curves for different values of  $\Delta t L$ . This is done by following the similar analysis for ETD methods in [47,48]. Note that the linear stability analysis here is different from that in [38].

Applying (23) to the Equation (28), we obtain

$$u^{k+1} = e^{\Delta t L} u^k \left\{ \frac{59}{128} + \frac{15}{128} \left(1 + \frac{4}{3} \Delta t N\right) + \frac{9}{64} \left(1 + \frac{4}{3} \Delta t N\right) \left(3 + 2\Delta t N + \frac{8}{9} (\Delta t N)^2\right) \right\}. \quad (29)$$

Let  $\lambda = N\Delta t$ , and substituting  $u^k = e^{ik\theta}$  into the above equation, we have

$$e^{i\theta} = e^{\Delta t L} \left(1 + \lambda + \frac{1}{2} \lambda^2 + \frac{1}{6} \lambda^3\right). \quad (30)$$

We plot the stability regions on the complex plane of  $\lambda$  in Figure 2, for  $\Delta t L = -0.5, -5,$  and  $-10$ . The interior of the closed curves is the stable region. The size of the stability region is an increasing function of  $|\Delta t L|$ . As  $L \rightarrow -\infty$ , the stability region approaches the whole complex plane. This is consistent with the property of the method, because the linear part is integrated exactly. Therefore, the larger the linear part is, the larger the stability region is.

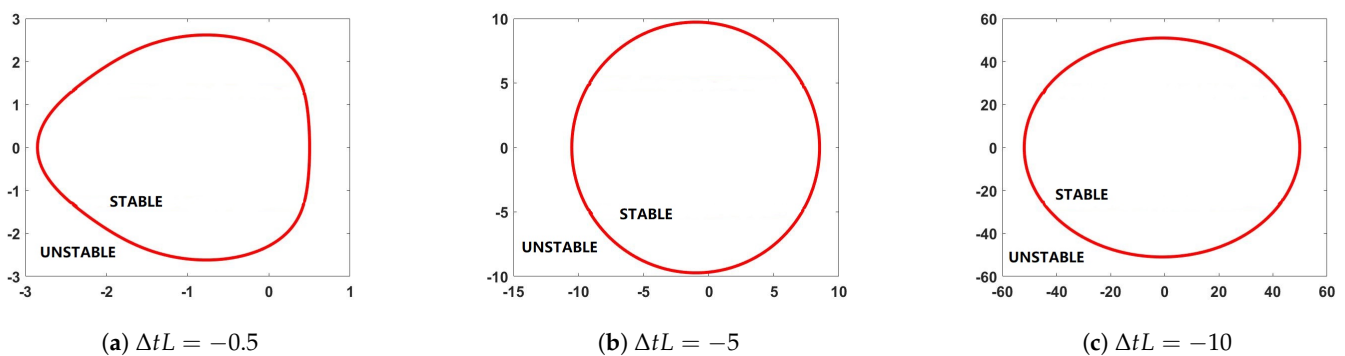


Figure 2. Linear stability regions for the third-order SSP IF Runge-Kutta scheme.

#### 4. Numerical Experiments

In this section, numerical examples are presented to demonstrate the stability, accuracy, and large CFL numbers for the WENO spatial discretizations coupled with the third-order Krylov SSP integrating factor Runge–Kutta method for 1D and 2D scalar hyperbolic equations. Both the linear advection equation and nonlinear Burgers’ equations are tested. For all numerical experiments in this paper, the dimension of the Krylov subspace  $M$  is taken as 25.

**Example 1.** Consider the one-dimensional Burgers’ equation

$$u_t + \left(\frac{1}{2}u^2\right)_x = 0, \quad x \in (-1, 1), \quad t > 0 \quad (31)$$

with periodic boundary conditions. The initial condition is given by  $u(x, 0) = 0.3 + 0.7 \sin \pi x$ . The exact solution can be obtained by using the method of characteristics and Newton’s method, namely,

$$u(x, t) = \frac{x - x_0}{t},$$

where  $x_0$  is the solution to the nonlinear equation

$$x = (0.3 + 0.7 \sin(\pi x_0))t + x_0.$$

The computation is carried up to time  $T = \frac{0.5}{\pi^2}$ , which is before shocks are formed. Here,  $CFL = \alpha_x \frac{\Delta t}{\Delta x}$ , where  $\Delta t$  and  $\Delta x$  are the time-step size and spatial step size, respectively, and  $\alpha_x$  is defined in (5). Different CFL numbers are tried. For  $CFL = 4$  and  $CFL = 10$ , the  $L_\infty$  and  $L_1$  errors and order of accuracy for the WENO3 method coupled with the

proposed third-order Krylov SSP IF Runge–Kutta method are reported in Table 1. We can see that the CFL number can be as large as 10 in the new method while still keeping the third order of accuracy.

**Table 1.** Example 1, numerical errors and accuracy orders for the WENO3 scheme with the third-order SSP IF Runge–Kutta time discretization.  $T = \frac{0.5}{\pi^2}$ .

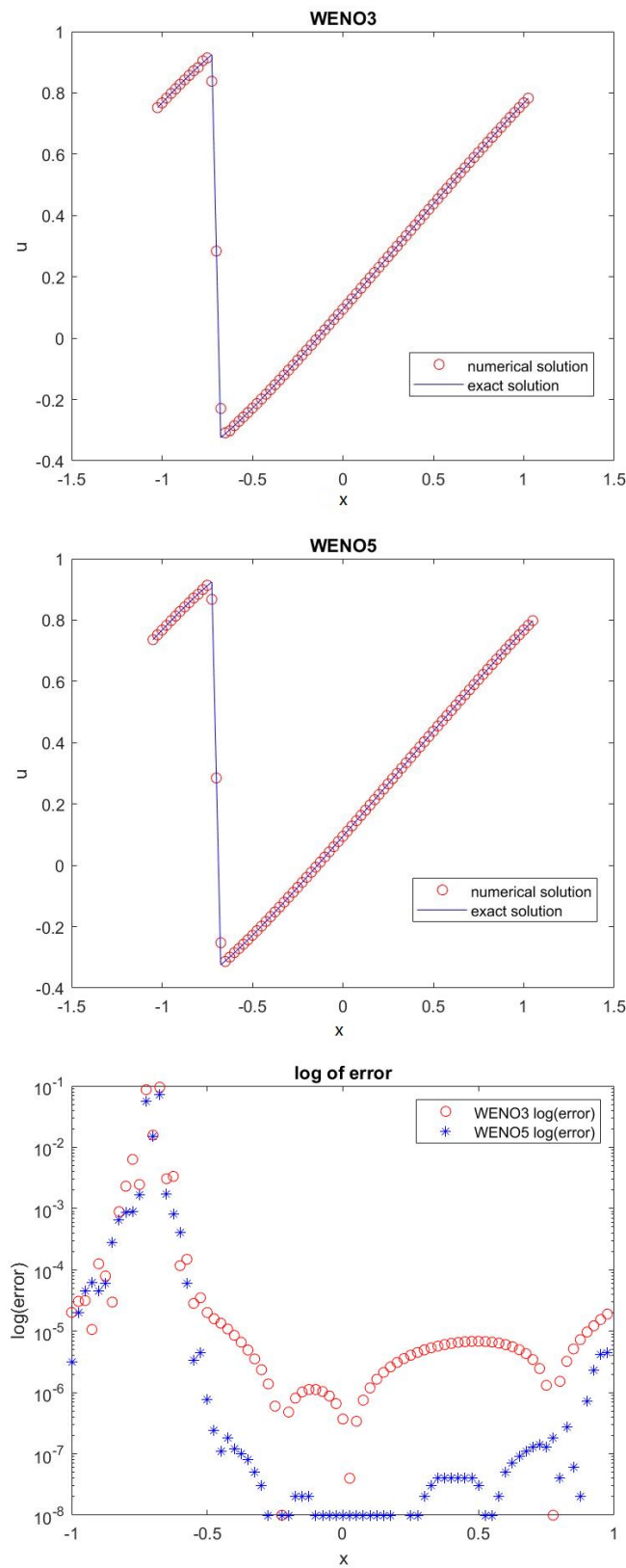
	$n$	$L_\infty$ Error	$L_\infty$ Order	$L_1$ Error	$L_1$ Order
CFL =4	80	$8.4928 \times 10^{-5}$	-	$1.4150 \times 10^{-5}$	-
	160	$1.0082 \times 10^{-5}$	3.074	$2.3096 \times 10^{-6}$	2.615
	320	$1.0759 \times 10^{-6}$	3.228	$2.4674 \times 10^{-7}$	3.227
	640	$1.2844 \times 10^{-7}$	3.066	$2.9298 \times 10^{-8}$	3.074
	1280	$1.5870 \times 10^{-8}$	3.017	$3.6127 \times 10^{-9}$	3.020
	2560	$1.9786 \times 10^{-9}$	3.004	$4.5006 \times 10^{-10}$	3.005
CFL = 10	80	$8.4928 \times 10^{-5}$	-	$1.4150 \times 10^{-5}$	-
	160	$1.0475 \times 10^{-5}$	3.019	$2.4040 \times 10^{-6}$	2.557
	320	$7.9553 \times 10^{-6}$	0.397	$1.9065 \times 10^{-6}$	0.334
	640	$1.3174 \times 10^{-6}$	2.594	$3.1101 \times 10^{-7}$	2.616
	1280	$2.1668 \times 10^{-7}$	2.604	$5.0765 \times 10^{-8}$	2.615
	2560	$2.7342 \times 10^{-8}$	2.986	$6.3857 \times 10^{-9}$	2.991

When we implement the fifth-order WENO with the third-order SSP IF Runge–Kutta method, in order to keep the consistency of truncation errors in the spatial and temporal directions, we are using  $\Delta t = (\Delta x)^{5/3}$ . The results are reported in Table 2. We observe that the numerical accuracy orders are actually higher than five. We also notice that for  $n = 80$ , WENO5 has larger errors than WENO3. This is because WENO schemes are highly nonlinear, and they have asymptotical convergence properties.

**Table 2.** Example 1. Numerical errors and accuracy orders for the WENO5 scheme with the third-order SSP IF Runge–Kutta time discretization.  $T = \frac{0.5}{\pi^2}$ .

$n$	$L_\infty$ Error	$L_\infty$ Order	$L_1$ Error	$L_1$ Order
80	$4.9425 \times 10^{-3}$	-	$4.2436 \times 10^{-4}$	-
160	$5.4341 \times 10^{-7}$	13.151	$5.0832 \times 10^{-8}$	13.027
320	$1.4364 \times 10^{-9}$	8.563	$1.5245 \times 10^{-10}$	8.381
640	$4.5605 \times 10^{-12}$	8.299	$4.9620 \times 10^{-13}$	8.263
1280	$2.1716 \times 10^{-13}$	4.392	$3.8558 \times 10^{-14}$	3.686

We also implement the proposed WENO schemes with SSP IF Runge–Kutta methods till time  $T = 1$ , when shocks are formed. In Figure 3, the exact solution and the numerical solutions of the third-order WENO (top) and the fifth-order WENO (middle) are displayed. It is clearly shown in the Figure that the numerical shock is well-resolved for both WENO methods. We also plot the errors in log scale for both WENO3 and WENO5 at  $T = 1$  (bottom). We can see that WENO5 has much smaller errors than WENO3 in smooth regions at this time.



**Figure 3.** Top and middle: The numerical solutions (symbols) versus the exact solution (solid line) with  $n = 80$  for Example 1.  $T = 1$ . Top: WENO3 with the third-order SSP IF Runge-Kutta method; middle: WENO5 with the third-order SSP IF Runge-Kutta method. Bottom:  $\log_{10}(\text{error})$  for WENO 3 (red circles) and WENO5 (blue asterisks).

**Example 2.** Consider the two-dimensional linear advection equation

$$u_t + u_x + u_y = 0, \quad (x, y) \in (0, 2\pi) \times (0, 2\pi), \quad t > 0, \tag{32}$$

with periodic boundary conditions. The initial condition is given by  $u(x, y, 0) = \sin(x + y)$ . The exact solution is  $u(x, y, t) = \sin(x + y - 2t)$ . For two-dimensional cases, the proposed Krylov method is used. The CFL number in this case is defined by  $CFL = \Delta t \left( \frac{\alpha_x}{\Delta x} + \frac{\alpha_y}{\Delta y} \right)$ , where  $\alpha_x = \max_u |f'(u)|$ , and  $\alpha_y = \max_u |g'(u)|$ .

The computation is carried up to  $T = 1$ . The  $L_\infty$  and  $L_1$  errors and order of accuracy for the third-order WENO method coupled with the third-order Krylov SSP IF RK method are reported in Table 3 for different CFL numbers. We see that the CFL number can be as large as five in the proposed new method, while still keeping the third order of accuracy.

**Table 3.** Example 2. Numerical errors and accuracy orders for the WENO3 scheme with the third-order SSP IF Runge–Kutta time discretization.  $T = 1$ .

	$n \times m$	$L_\infty$ Error	$L_\infty$ Order	$L_1$ Error	$L_1$ Order
CFL=3	40 × 40	$2.3920 \times 10^{-2}$	-	$7.5559 \times 10^{-3}$	-
	80 × 80	$2.3242 \times 10^{-3}$	3.363	$5.8279 \times 10^{-4}$	3.697
	160 × 160	$9.2379 \times 10^{-5}$	4.653	$2.8572 \times 10^{-5}$	4.350
	320 × 320	$3.8538 \times 10^{-6}$	4.583	$1.3139 \times 10^{-6}$	4.443
	640 × 640	$2.3876 \times 10^{-7}$	4.013	$1.0040 \times 10^{-7}$	3.710
	1280 × 1280	$2.2247 \times 10^{-8}$	3.424	$1.2550 \times 10^{-8}$	3.000
CFL = 5	40 × 40	$6.4163 \times 10^{-1}$	-	$1.5473 \times 10^{-1}$	-
	80 × 80	$2.4451 \times 10^{-3}$	8.036	$5.8007 \times 10^{-4}$	8.059
	160 × 160	$9.2652 \times 10^{-5}$	4.722	$2.8558 \times 10^{-5}$	4.344
	320 × 320	$3.8539 \times 10^{-6}$	4.587	$1.3139 \times 10^{-6}$	4.442
	640 × 640	$2.3876 \times 10^{-7}$	4.013	$1.0040 \times 10^{-7}$	3.710
	1280 × 1280	$2.2247 \times 10^{-8}$	3.424	$1.2550 \times 10^{-8}$	3.000

Again, when we implement the fifth-order WENO with the third-order Krylov SSP Runge–Kutta method, in order to keep the consistency of truncation errors in the spatial and temporal directions, we are using

$$\Delta t = \min((\Delta x)^{5/3}, (\Delta y)^{5/3}).$$

The results are reported in Table 4. We observe that the numerical accuracy orders are actually higher than five.

**Table 4.** Example 2. Numerical errors and accuracy orders for the WENO5 scheme with the third-order SSP IF Runge–Kutta time discretization.  $T = 1$ .

$n \times m$	$L_\infty$ Error	$L_\infty$ Order	$L_1$ Error	$L_1$ Order
40 × 40	$4.3511 \times 10^{-5}$	-	$1.0357 \times 10^{-5}$	-
80 × 80	$9.8415 \times 10^{-6}$	2.144	$1.1647 \times 10^{-6}$	3.153
160 × 160	$1.9511 \times 10^{-8}$	8.978	$4.3367 \times 10^{-9}$	8.069
320 × 320	$3.1284 \times 10^{-11}$	9.285	$1.7546 \times 10^{-11}$	7.949
640 × 640	$3.9935 \times 10^{-13}$	6.292	$2.9549 \times 10^{-13}$	5.892
1280 × 1280	$9.5479 \times 10^{-15}$	5.386	$4.6189 \times 10^{-15}$	5.999

**Example 3.** Consider the two-dimensional Burgers’ equation

$$u_t + \left( \frac{1}{2}u^2 \right)_x + \left( \frac{1}{2}u^2 \right)_y = 0, \quad (x, y) \in (-2, 2) \times (-2, 2), \quad t > 0 \tag{33}$$

with periodic boundary conditions. The initial condition is given by  $u(x, y, 0) = 0.3 + 0.7 \sin(\pi(x + y)/2)$ . The exact solution can be obtained by using the method of characteristics and Newton’s method.

This problem has a smooth solution before time  $\frac{0.5}{\pi^2}$ ; after that, shocks will form. We first calculate up to  $T = \frac{0.5}{\pi^2}$  with  $M = 25$  at which  $L_\infty$  and  $L_1$  errors are measured. Numerical errors and order of accuracy for the third-order WENO with the third-order SSP IF Runge–Kutta method are recorded in Table 5. We see that the desired third order of accuracy is achieved with the CFL number as large as five.

**Table 5.** Example 3. Numerical errors and accuracy orders for WENO3 scheme with the third-order SSP IF Runge–Kutta time discretization.  $T = \frac{0.5}{\pi^2}$ .

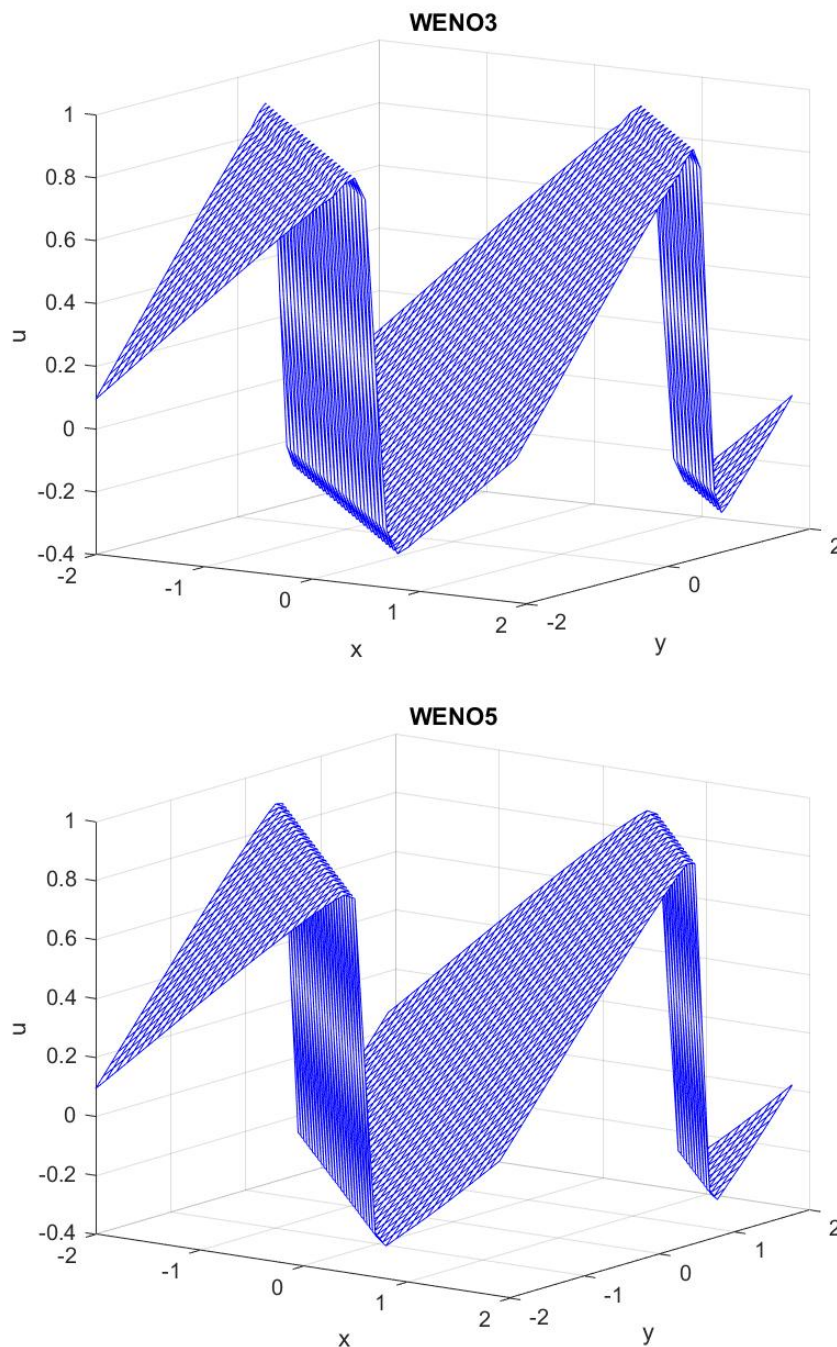
	$n \times m$	$L_\infty$ Error	$L_\infty$ Order	$L_1$ Error	$L_1$ Order
CFL=3	20 × 20	$1.1356 \times 10^{-2}$	-	$2.0771 \times 10^{-3}$	-
	40 × 40	$1.5136 \times 10^{-3}$	2.907	$2.3810 \times 10^{-4}$	3.125
	80 × 80	$8.4928 \times 10^{-5}$	4.156	$1.4304 \times 10^{-5}$	4.057
	160 × 160	$5.1569 \times 10^{-6}$	4.042	$1.1611 \times 10^{-6}$	3.623
	320 × 320	$4.7135 \times 10^{-7}$	3.452	$1.1089 \times 10^{-7}$	3.388
	640 × 640	$5.7582 \times 10^{-8}$	3.033	$1.3480 \times 10^{-8}$	3.040
	1280 × 1280	$7.2197 \times 10^{-9}$	2.996	$1.6847 \times 10^{-9}$	3.000
CFL = 5	20 × 20	$1.1356 \times 10^{-2}$	-	$2.0771 \times 10^{-3}$	-
	40 × 40	$1.5136 \times 10^{-3}$	2.907	$2.3810 \times 10^{-4}$	3.125
	80 × 80	$8.4928 \times 10^{-5}$	4.156	$1.4304 \times 10^{-5}$	4.057
	160 × 160	$1.0475 \times 10^{-5}$	3.019	$2.4046 \times 10^{-6}$	2.573
	320 × 320	$1.4410 \times 10^{-6}$	2.862	$3.3236 \times 10^{-7}$	2.855
	640 × 640	$2.2619 \times 10^{-7}$	2.671	$5.1713 \times 10^{-8}$	2.684
	1280 × 1280	$2.8333 \times 10^{-8}$	2.997	$6.4587 \times 10^{-9}$	3.001

Again, when we implement the fifth-order WENO with the third-order SSP Runge–Kutta method, in order to keep the consistency of truncation errors in the spatial and temporal directions, we use  $\Delta t = \min((\Delta x)^{5/3}, (\Delta y)^{5/3})$ . The results are reported in Table 6. We see that a fifth-order of accuracy is achieved.

**Table 6.** Example 3. Numerical errors and accuracy orders for WENO5 scheme with the third-order SSP IF Runge–Kutta time discretization.  $T = \frac{0.5}{\pi^2}$ .

$n \times m$	$L_\infty$ Error	$L_\infty$ Order	$L_1$ Error	$L_1$ Order
40 × 40	$2.0376 \times 10^{-4}$	-	$1.9799 \times 10^{-5}$	-
80 × 80	$3.7412 \times 10^{-6}$	5.767	$6.1221 \times 10^{-7}$	5.015
160 × 160	$7.3331 \times 10^{-8}$	5.673	$1.6803 \times 10^{-8}$	5.187
310 × 320	$2.4151 \times 10^{-9}$	4.924	$5.4418 \times 10^{-10}$	4.948
640 × 640	$7.3460 \times 10^{-11}$	5.039	$1.6789 \times 10^{-11}$	5.019
1280 × 1280	$2.2919 \times 10^{-12}$	5.002	$5.2834 \times 10^{-13}$	4.990

We also implement the Krylov SSP IF Runge–Kutta WENO methods till time  $T = 1$ , when shocks are formed. In Figure 4, the numerical solutions of the third-order WENO (top) and the fifth-order WENO (bottom) are displayed. We can see that the numerical shocks are resolved very well for both WENO methods.



**Figure 4.** The numerical solutions with  $n \times m = 40 \times 40$  for Example 3.  $T = 1$ . Top: WENO3 with the third-order Krylov SSP IF Runge-Kutta time scheme; bottom: WENO5 with the third-order Krylov SSP IF Runge-Kutta time scheme.

## 5. Discussion and Conclusions

In this paper, we developed a class of efficient high-order numerical methods by extending the work in [38] to solve general nonlinear 2D hyperbolic equations. High-order WENO schemes are incorporated with a large time-stepping third-order SSP integrating factor Runge–Kutta method. The reason we chose SSP RK methods as the time discretization is because they are the most efficient time-method with WENO spatial discretization for solving pure hyperbolic PDEs. A detailed discussion can be found in [20]. It will also be interesting to compare the method proposed here with other methods, such as semi-implicit schemes, which will be our future work.



Due to the large size of matrix exponentials rising from two-dimensional problems, we used the Krylov subspace projection method to efficiently approximate the matrix exponential operator. Numerical examples show that we can use large CFL numbers to achieve the desired accuracy of the methods. Numerical examples also verify that the error of the Krylov subspace approximation does not affect the accuracy orders of the third-order integrating factor SSP Runge–Kutta time discretization or the high-order WENO spatial discretizations, which suggests that the Krylov subspace approximation error is already much smaller than the temporal and spatial truncation errors of the numerical schemes.

The linear stability analysis in the paper is only for the time scheme. Because WENO schemes are highly nonlinear schemes, the complete stability analysis for the fully discretized schemes, including both time and spatial directions, is still an open problem, which will be studied in future research.

In this first paper, we only tested the proposed method for scalar equations. In future work, we will extend the method to solve more complicated hyperbolic PDE systems and higher-dimensional problems.

**Funding:** This research was supported by Indiana University Faculty Research Grant.

**Data Availability Statement:** The data that support the findings of this study are available from the corresponding author upon reasonable request.

**Conflicts of Interest:** The author declares no conflict of interest.

## References

- Jiang, G.; Shu, C.-W. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.* **1996**, *126*, 202–228. [[CrossRef](#)]
- Liu, X.-D.; Osher, S.; Chan, T. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.* **1994**, *115*, 200–212. [[CrossRef](#)]
- Harten, A.; Engquist, B.; Osher, S.; Chakravarthy, S. Uniformly high-order essentially non-oscillatory schemes, III. *J. Comput. Phys.* **1987**, *71*, 231–303. [[CrossRef](#)]
- Shu, C.-W.; Osher, S. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.* **1988**, *77*, 439–471. [[CrossRef](#)]
- Shu, C.-W.; Osher, S. Efficient implementation of essentially non-oscillatory shock capturing schemes, II. *J. Comput. Phys.* **1989**, *83*, 32–78. [[CrossRef](#)]
- Zhang, Y.-T.; Shi, J.; Shu, C.-W.; Zhou, Y. Numerical viscosity and resolution of high-order weighted essentially nonoscillatory schemes for compressible flows with high Reynolds numbers. *Phys. Rev. E* **2003**, *68*, 046709. [[CrossRef](#)] [[PubMed](#)]
- Chen, S. Fixed-point fast sweeping WENO methods for steady-state solution of scalar hyperbolic conservation laws. *Int. J. Numer. Anal. Model.* **2014**, *11*, 117–130.
- Hao, W.; Hauenstein, J.D.; Shu, C.-W.; Somme, A.J.; Xu, Z.; Zhang, Y.-T. A homotopy method based on WENO schemes for solving steady-state problems of hyperbolic conservation laws. *J. Comput. Phys.* **2013**, *250*, 332–346. [[CrossRef](#)]
- Wu, L.; Zhang, Y.-T.; Zhang, S.; Shu, C.-W. high-order fixed-point sweeping WENO methods for steady-state of hyperbolic conservation laws and its convergence study. *Commun. Comput. Phys.* **2016**, *20*, 835–869. [[CrossRef](#)]
- Xiong, T.; Zhang, M.; Zhang, Y.-T.; Shu, C.-W. Fast sweeping fifth-order WENO scheme for static Hamilton-Jacobi equations with accurate boundary treatment. *J. Sci. Comput.* **2010**, *45*, 514–536. [[CrossRef](#)]
- Zhang, Y.-T.; Zhao, H.K.; Chen, S. Fixed-point iterative sweeping methods for static Hamilton-Jacobi equations. *Methods Appl. Anal.* **2006**, *13*, 299–320. [[CrossRef](#)]
- Zhang, Y.-T.; Zhao, H.K.; Qian, J. high-order fast sweeping methods for static Hamilton-Jacobi equation. *J. Sci. Comput.* **2006**, *29*, 25–56. [[CrossRef](#)]
- Jiang, T.; Zhang, Y.-T. Krylov implicit integration factor WENO methods for semilinear and fully nonlinear advection–diffusion–reaction equations. *J. Comput. Phys.* **2013**, *253*, 368–388. [[CrossRef](#)]
- Jiang, T.; Zhang, Y.-T. Krylov single-step implicit integration factor WENO methods for advection–diffusion–reaction equations. *J. Comput. Phys.* **2016**, *311*, 22–44. [[CrossRef](#)]
- Lu, D.; Zhang, Y.-T. Computational complexity study on Krylov integration factor WENO method for high spatial dimension convection–diffusion problems. *J. Sci. Comput.* **2017**, *73*, 980–1027. [[CrossRef](#)]
- Zhao, R.; Zhang, Y.-T.; Chen, S. Krylov implicit integration factor WENO method for SIR model with directed diffusion. *Discrete Contin. Dyn. Syst. Ser. B* **2019**, *v24*, 4983–5001. [[CrossRef](#)]
- Lu, D.; Chen, S.; Zhang, Y.-T. third-order WENO scheme on sparse grids for hyperbolic equations. *Pure Appl. Math. Q.* **2018**, *14*, 57–86. [[CrossRef](#)]
- Gottlieb, S.; Ketcheson, D.I.; Shu, C.-W. *Strong Stability Preserving Runge–Kutta and Multistep Time Discretizations*; World Scientific Press: Singapore, 2011.
- Gottlieb, S.; Shu, C.-W. Total variation diminishing Runge–Kutta schemes. *Math. Comp.* **1998**, *67*, 73–85. [[CrossRef](#)]

20. Gottlieb, S.; Shu, C.-W.; Tadmor, E. Strong stability-preserving high-order time discretization methods. *SIAM Rev.* **2001**, *43*, 89–112. [[CrossRef](#)]
21. Bourlioux, A.; Layton, A.T.; Minion, M.L. High-order multi-implicit spectral deferred correction methods for problems of reactive flow. *J. Comput. Phys.* **2003**, *189*, 651–675. [[CrossRef](#)]
22. Dutt, A.; Greengard, L.; Rokhlin, V. Spectral deferred correction methods for ordinary differential equations. *BIT Numer. Math.* **2000**, *40*, 241–266. [[CrossRef](#)]
23. Huang, J.; Jia, J.; Minion, M. Arbitrary order Krylov deferred correction methods for differential algebraic equations. *J. Comput. Phys.* **2007**, *221*, 739–760. [[CrossRef](#)]
24. Layton, A.T.; Minion, M.L. Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics. *J. Comput. Phys.* **2004**, *194*, 697–715. [[CrossRef](#)]
25. Minion, M.L. Semi-implicit spectral deferred correction methods for ordinary differential equations. *Commun. Math. Sci.* **2003**, *1*, 471–500. [[CrossRef](#)]
26. Ascher, U.M.; Ruuth, S.J.; Spiteri, R.J. Implicit-explicit Runge–Kutta methods for time-dependent partial differential equations. *Appl. Numer. Math.* **1997**, *25*, 151–167. [[CrossRef](#)]
27. Ascher, U.; Ruuth, S.; Wetton, B. Implicit-explicit methods for time-dependent PDE's. *SIAM J. Numer. Anal.* **1995**, *32*, 797–823. [[CrossRef](#)]
28. 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](#)]
29. Kennedy, C.A.; Carpenter, M.H. Additive Runge–Kutta schemes for convection-diffusion-reaction equations. *Appl. Numer. Math.* **2003**, *44*, 139–181. [[CrossRef](#)]
30. Najm, H.N.; Wyckoff, P.S.; Knio, O.M. A semi-implicit numerical scheme for reacting flow. I. Stiff chemistry. *J. Comput. Phys.* **1998**, *143*, 381–402. [[CrossRef](#)]
31. Ruuth, S.J. Implicit-explicit methods for reaction-diffusion problems in pattern formation. *J. Math. Biol.* **1995**, *34*, 148–176. [[CrossRef](#)]
32. Verwer, J.G.; Sommeijer, B.P. An implicit-explicit Runge–Kutta–Chebyshev scheme for diffusion-reaction equations. *SIAM J. Sci. Comput.* **2004**, *25*, 1824–1835. [[CrossRef](#)]
33. Zhong, X. Additive semi-implicit Runge–Kutta methods for computing high-speed nonequilibrium reactive flows. *J. Comput. Phys.* **1996**, *128*, 19–31. [[CrossRef](#)]
34. Christlieb, A.; Ong, B.; Qiu, J.-M. Integral deferred correction methods constructed with high-order Runge–Kutta integrators. *Math. Comput.* **2010**, *79*, 761–783. [[CrossRef](#)]
35. Christlieb, A.; Gottlieb, S.; Grant, Z.; Seal, D.C. Explicit strong stability preserving multistage two-derivative time-stepping schemes. *J. Sci. Comput.* **2016**, *68*, 914–942. [[CrossRef](#)]
36. Shu, C.-W. Total-variation diminishing time discretizations. *SIAM J. Sci. Stat. Comp.* **1988**, *9*, 1073–1084. [[CrossRef](#)]
37. Li, S.-J.; Luo, L.-S.; Wang, Z.J.; Ju, L. An exponential time-integrator scheme for steady and unsteady inviscid flows. *J. Comput. Phys.* **2018**, *365*, 206–225. [[CrossRef](#)]
38. Isherwood, L.; Grant, Z.J.; Gottlieb, S. Strong stability preserving integrating factor Runge–Kutta methods. *SIAM J. Numer. Anal.* **2018**, *56*, 3276–3307. [[CrossRef](#)]
39. Isherwood, L.; Grant, Z.J.; Gottlieb, S. Downwinding for preserving strong stability in explicit integrating factor Runge–Kutta methods. *Pure Appl. Math. Q.* **2018**, *14*, 3–25. [[CrossRef](#)]
40. Isherwood, L.; Grant, Z.J.; Gottlieb, S. Strong stability preserving integrating factor two-step Runge–Kutta methods. *J. Sci. Comput.* **2019**. [[CrossRef](#)]
41. Gallopoulos, E.; Saad, Y. Efficient solution of parabolic equations by Krylov approximation methods. *SIAM J. Sci. Stat. Comput.* **1992**, *13*, 1236–1264. [[CrossRef](#)]
42. Chen, S.; Zhang, Y.T. Krylov implicit integration factor methods for spatial discretization on high-dimensional unstructured meshes: Application to discontinuous Galerkin methods. *J. Comput. Phys.* **2011**, *230*, 4336–4352. [[CrossRef](#)]
43. Liu, Y.; Cheng, Y.; Chen, S.; Zhang, Y.-T. Krylov implicit integration factor discontinuous Galerkin methods on sparse grids for high-dimensional reaction-diffusion equations. *J. Comput. Phys.* **2019**, *388*, 90–102. [[CrossRef](#)]
44. Moler, C.; Van Loan, C. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.* **2003**, *45*, 3–49. [[CrossRef](#)]
45. Trefethen, L.N.; Bau, D. *Numerical Linear Algebra*; SIAM: Philadelphia, PA, USA, 1997.
46. Higham, N.J. The scaling and squaring method for the matrix exponential revisited. *SIAM Rev.* **2009**, *51*, 747–764. [[CrossRef](#)]
47. Beylkin, G.; Keiser, J.M.; Vozovoi, L. A new class of time discretization schemes for the solution of nonlinear PDEs. *J. Comput. Phys.* **1998**, *147*, 362–387. [[CrossRef](#)]
48. Karniadakis, G.E.; Israeli, M.; Orszag, S.A. high-order splitting methods for the incompressible Navier–Stokes equations. *J. Comput. Phys.* **1991**, *97*, 414–443. [[CrossRef](#)]