Journal of Computational Physics 311 (2016) 22–44

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Krylov single-step implicit integration factor WENO methods for advection–diffusion–reaction equations

Tian Jiang, Yong-Tao Zhang *

Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556, USA

ARTICLE INFO

Article history:
Received 3 June 2015
Received in revised form 15 December 2015
Accepted 16 January 2016
Available online 21 January 2016

Keywords:
Advection–diffusion–reaction equations
Single-step methods
Implicit integration factor methods
Krylov subspace approximation
Weighted essentially non-oscillatory schemes

ABSTRACT

Implicit integration factor (IIF) methods were developed in the literature for solving time-dependent stiff partial differential equations (PDEs). Recently, IIF methods were combined with weighted essentially non-oscillatory (WENO) schemes in Jiang and Zhang (2013) [19] to efficiently solve stiff nonlinear advection–diffusion–reaction equations. The methods can be designed for arbitrary order of accuracy. The stiffness of the system is resolved well and the methods are stable by using time step sizes which are just determined by the non-stiff hyperbolic part of the system. To efficiently calculate large matrix exponentials, Krylov subspace approximation is directly applied to the implicit integration factor (IIF) methods. So far, the IIF methods developed in the literature are multistep methods. In this paper, we develop Krylov single-step IIF-WENO methods for solving stiff advection–diffusion–reaction equations. The methods are designed carefully to avoid generating positive exponentials in the matrix exponentials, which is necessary for the stability of the schemes. We analyze the stability and truncation errors of the single-step IIF schemes. Numerical examples of both scalar equations and systems are shown to demonstrate the accuracy, efficiency and robustness of the new methods.

1. Introduction

A central question in numerically solving time dependent partial differential equations (PDEs) is how to design efficient and high order temporal numerical schemes. In the literature, a lot of state-of-the-art high order time-stepping methods were developed. For example, the total variation diminishing (TVD) Runge–Kutta (RK) schemes [32,33,12,13]; spectral deferred correction (SDC) methods [4,8,17,24,27]; high order implicit–explicit (IMEX) multistep/RK methods [1,21,23,36,38]; hybrid methods of SDC and high order RK schemes [6]; etc.

Integration factor (IF) methods are a class of "exactly linear part" time discretization methods for the solution of nonlinear partial differential equations (PDEs) with the linear highest spatial derivatives. This class of methods performs the time evolution of the stiff linear operator via evaluation of an exponential function of the corresponding matrix. Hence the integration factor type time discretization can remove both the stability constraint and time direction numerical errors from the high order derivatives [3,7,26,22].

In [29], a class of efficient implicit integration factor (IIF) methods were developed for solving systems with both stiff linear and nonlinear terms. A novel property of the methods is that the implicit terms are free of the exponential operation of the linear terms. Hence when the methods are applied to PDEs with stiff nonlinear reactions (e.g. the reaction–diffusion

* Corresponding author. Tel.: +1 574 631 6079.
E-mail addresses: jtian29@gmail.com (T. Jiang), yzhang10@nd.edu (Y.-T. Zhang).

http://dx.doi.org/10.1016/j.jcp.2016.01.021
0021-9991/

systems arising from mathematical models in computational developmental biology), the exact evaluation of the linear part is decoupled from the implicit treatment of the nonlinear reaction terms. As a result, the size of the nonlinear system arising from the implicit treatment is independent of the number of spatial grid points; it only depends on the number of the original PDEs. This distinguishes IIF methods [29] from implicit exponential time differencing (ETD) methods in [3]. The methods can have high order accuracy (arbitrary order in principle) for stiff reaction–diffusion systems with linear diffusion terms, and large stability region due to the implicit nature of the schemes. To deal with the difficulty in implementing integration factor type method for high dimensional problems, we developed the compact IIF methods [30] on rectangular meshes, and Krylov IIF methods [5] on general unstructured meshes for complex domains. The compact IIF methods were extended to curvilinear coordinates, such as polar and spherical coordinates in [25].

Recently in [19], we developed Krylov IIF weighted essentially non-oscillatory (IIF-WENO) methods for solving nonlinear advection–diffusion–reaction (ADR) equations. The ADR equations in three spatial dimensions have the following general form

$$u_t + f(u)_x + g(u)_y + h(u)_z = \nabla \cdot (\mathbf{D}\nabla u) + r(u), \tag{1}$$

where u is the unknown, f, g and h are advection fluxes in three spatial dimensions respectively, $\mathbf{D}$ is the diffusion coefficient matrix, and r is the reaction term. The IIF-WENO methods can be designed for arbitrary order of accuracy. The stiffness of the system is resolved well and the methods are stable by using time step sizes which are just determined by the non-stiff hyperbolic part of the system. Large time step size computations are obtained. Krylov subspace approximations are applied in efficiently dealing with large matrix exponential computations in high spatial dimension problems.

The IIF methods developed in the literature so far are multistep methods. It is an open problem how to develop single-step IIF methods for solving stiff problems. While single-step explicit integration factor (EIF) methods can be designed straightforwardly as shown in this paper, the single-step IIF methods need to be designed carefully to preserve the "local implicit" property of the original IIF schemes in [29]. Namely, the implicit terms are free of the exponential operation. As a result, the implicit nonlinear system is decoupled for each spatial grid point. The size of the implicit nonlinear system at every spatial grid point only depends on the number of the original PDEs. This "local implicit" property provides a key factor for the computational efficiency of the IIF schemes, since no coupled large nonlinear system needs to be solved at every time step. The small size implicit nonlinear system can be efficiently solved by a fixed-point iteration method [29] or a Newton iteration method [5].

In this paper, we develop Krylov single-step IIF-WENO methods for solving stiff advection–diffusion–reaction equations. The methods are designed carefully to avoid generating positive exponentials in the matrix exponentials, which is necessary for the stability of the schemes. "Local implicit" property of the original IIF schemes and all other properties of Krylov multistep IIF-WENO methods in [19] are preserved. The rest of the paper is organized as following. In Section 2, we derive and formulate the Krylov single-step IIF-WENO methods for ADR equations. In Section 3, the truncation error and linear stability analysis are performed. Numerical experiments are presented in Section 4. Discussions and conclusions are given in Section 5.

2. High order Krylov single-step IIF-WENO methods

In this section we describe the new methods in details. The single-step integration factor (SIF) methods are derived. The novel part of the methods is in their temporal discretization. However for the self-contained description of the methods, the spatial discretizations are given at first.

2.1. Spatial discretization

In this paper, we use the third order finite difference WENO scheme with Lax–Friedrichs flux splitting [18] to discretize the nonlinear advection terms. WENO schemes are robust in dealing with nonlinear advection under different situations. We will give a brief sketch of the algorithms here. For the advection terms $f(u)_x + g(u)_y + h(u)_z$, the conservative finite-difference schemes we use approximate the point values at a uniform (or smoothly varying) grid (x_i, y_j, z_k) in a conservative fashion. Namely, the derivative $f(u)_x$ at (x_i, y_j, z_k) is approximated along the line $y = y_j, z = z_k$ by a conservative flux difference

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

where for the third order WENO scheme the numerical flux $\hat{f}_{i+1/2}$ depends on the three-point values $f(u_l)$ (here for the simplicity of notations, we use u_l to denote the value of the numerical solution u at the point x_l along the line $y = y_j, z = z_k$ with the understanding that the value could be different for different y and z coordinates), $l = i-1, i, i+1$, when the wind is positive (i.e., when $f'(u) \geq 0$ for the scalar case, or when the corresponding eigenvalue is positive for the system case with a local characteristic decomposition). This numerical flux $\hat{f}_{i+1/2}$ is written as a convex combination of two second order numerical fluxes based on two different substencils of two points each, and the combination coefficients depend on a "smoothness indicator" measuring the smoothness of the solution in each substencil. The detailed formulae is

$$\hat{f}_{i+1/2} = w_0\left[\frac{1}{2}f(u_i) + \frac{1}{2}f(u_{i+1})\right] + w_1\left[-\frac{1}{2}f(u_{i-1}) + \frac{3}{2}f(u_i)\right], \tag{3}$$

where

$$w_r = \frac{\alpha_r}{\alpha_1 + \alpha_2}, \qquad \alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2}, \qquad r = 0, 1. \tag{4}$$

$d_0 = 2/3, d_1 = 1/3$ 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". ϵ is a small positive number chosen to avoid the denominator becoming 0. We take $\epsilon = 10^{-3}$ in this paper.

When the wind is negative (i.e., when $f'(u) < 0$), right-biased stencil with numerical values $f(u_i)$, $f(u_{i+1})$ and $f(u_{i+2})$ are used to construct a third order WENO approximation to the numerical flux $\hat{f}_{i+1/2}$. The formulae for negative and positive wind cases are symmetric with respect to the point $x_{i+1/2}$. For the general case of $f(u)$, we perform the "Lax–Friedrichs flux splitting"

$$f^+(u) = \frac{1}{2}(f(u) + \alpha u), \qquad f^-(u) = \frac{1}{2}(f(u) - \alpha u), \tag{5}$$

where $\alpha = \max_u |f'(u)|$. $f^+(u)$ is the positive wind part, and $f^-(u)$ is the negative wind part. Corresponding WENO approximations are applied to find numerical fluxes $\hat{f}^+_{i+1/2}$ and $\hat{f}^-_{i+1/2}$ respectively. Similar procedures are applied to the other directions for $g(u)_y$ and $h(u)_z$. See [18,34] for more details. For diffusion terms, the second or fourth order central finite difference approximations (depending on the order of accuracy of time discretization) are used.

2.2. Krylov single-step IIF methods

2.2.1. Single-step explicit integration factor methods

We first present single-step explicit integration factor (SEIF) methods. The SEIF methods can be used to resolve the stiffness arising from the diffusion terms in advection–diffusion equations or advection–diffusion–reaction equations with nonstiff reaction terms. The SEIF method is straightforward to be derived and has been developed in the literature, for example as shown in [7] and its references, or in [20]. After spatial discretizations of an ADR equation (1), the following semi-discretized ODE system is obtained

$$\frac{d\vec{U}}{dt} = C\vec{U} + \vec{F}(\vec{U}), \tag{6}$$

where $C\vec{U}$ is the approximation for the diffusion terms by the second or fourth order finite difference schemes, and the nonlinear term $\vec{F}(\vec{U})$ includes the approximation for the nonlinear advection terms by the third order finite difference WENO scheme and nonstiff reaction terms. To derive SEIF methods, we multiply both sides of (6) by an integration factor e^{-Ct} to obtain

$$e^{-Ct}\frac{d\vec{U}}{dt} = e^{-Ct}(C\vec{U} + \vec{F}(\vec{U})), \tag{7}$$

and rewrite (7) as

$$\frac{d(e^{-Ct}\vec{U})}{dt} = e^{-Ct}\vec{F}(\vec{U}). \tag{8}$$

Denote $Y = e^{-Ct}\vec{U}$, then (8) can be written in term of Y as

$$Y_t = e^{-Ct}\vec{F}(e^{Ct}Y). \tag{9}$$

The next step we apply Runge–Kutta methods, for example the Heun's second order method or the Kutta's third order method to (9) to approximate $Y(t)$ instead of $\vec{U}(t)$. At last the relation $\vec{U} = e^{Ct}Y$ is used to get following schemes (**SEIF2** and **SEIF3**) for $\vec{U}$. The second order single-step explicit integration factor (SEIF2) scheme is

$$\begin{cases} \vec{U}^{(1)} = e^{C\Delta t}\left(\vec{U}_n + \Delta t\vec{F}(\vec{U}_n)\right), \\ \vec{U}_{n+1} = e^{C\Delta t}\left(\vec{U}_n + \dfrac{\Delta t}{2}\vec{F}(\vec{U}_n)\right) + \dfrac{\Delta t}{2}\vec{F}(\vec{U}^{(1)}). \end{cases} \tag{10}$$

And the third order single-step explicit integration factor (SEIF3) scheme is

$$\begin{cases} \vec{U}^{(1)} = e^{C\Delta t/2}\left(\vec{U}_n + \dfrac{\Delta t}{2}\vec{F}(\vec{U}_n)\right), \\ \vec{U}^{(2)} = e^{C\Delta t}\left(\vec{U}_n - \Delta t\vec{F}(\vec{U}_n)\right) + 2\Delta t e^{C\Delta t/2}\vec{F}(\vec{U}^{(1)}), \\ \vec{U}_{n+1} = e^{C\Delta t}\left(\vec{U}_n + \dfrac{\Delta t}{6}\vec{F}(\vec{U}_n)\right) + \dfrac{2\Delta t}{3}e^{C\Delta t/2}F(\vec{U}^{(1)}) + \dfrac{\Delta t}{6}\vec{F}(\vec{U}^{(2)}). \end{cases} \tag{11}$$

In order to efficiently calculate the product of a large matrix exponential and a vector, Krylov subspace approximation is used, as that in the Krylov multistep IIF-WENO schemes for ADR systems in [19]. The details are presented in the following section 2.2.3.

2.2.2. Single-step implicit integration factor methods

In this section we develop new single-step implicit integration factor (SIIF) methods. For advection–diffusion–reaction equations arising from biological and physical applications, they often have stiff nonlinear reaction terms. It is necessary to treat nonlinear stiff reaction terms implicitly in order to achieve large time step sizes for efficient time evolution. However, unlike the SEIF methods, it is not straightforward to derive SIIF methods. There are two difficulties here. One is that the approach used in the last section for SEIF methods, i.e., directly applying implicit Runge–Kutta methods to the transformed equation (9), often leads to positive exponentials in the matrix exponentials. Our numerical experiments show that the computations are unstable with these positive exponentials. The second question is how to preserve the local implicit property of the original multistep IIF schemes in [29]. Namely, the implicit terms are free of the exponential operation. As a result, the implicit nonlinear system is decoupled for each spatial grid point. The size of the implicit nonlinear system at every spatial grid point only depends on the number of the original PDEs. Motivated by the ideas in [7] for developing Runge–Kutta type ETD schemes, we derive single-step implicit integration factor (SIIF) methods here.

We start from the following semi-discretized ODE system after spatial discretizations of the ADR system (1)

$$\vec{U}_t = C\vec{U} + \vec{F}(\vec{U}) + \vec{R}(\vec{U}), \tag{12}$$

where $C\vec{U}$ is from the diffusion terms, $\vec{F}(\vec{U})$ is from the advection terms and $\vec{R}(\vec{U})$ is from the reaction terms. Notice that different from the form (6), here we separate the nonlinear advection terms and the nonlinear reaction terms since they are treated differently in the implicit IF schemes.

To construct SIIF methods for (12), we multiply it by the integration factor e^{-Ct} and integrate over one time step from t_n to $t_{n+1} = t_n + \Delta t$ to obtain

$$\vec{U}(t_{n+1}) = e^{C\Delta t}\vec{U}(t_n) + e^{C\Delta t}\int_0^{\Delta t} e^{-C\tau}\left(\vec{F}(\vec{U}(t_n+\tau)) + \vec{R}(\vec{U}(t_n+\tau))\right)d\tau. \tag{13}$$

Note that the time step size Δt can vary at every time step, so it is actually Δt_n. For the simplicity of notations, we use Δt here. The linear term $C\vec{U}$ has been integrated exactly in the time direction, hence the stiffness associated with the linear operator is removed. Two of the nonlinear terms in (13) have different properties. The nonlinear reaction term $\vec{R}(\vec{U})$ is usually stiff but local, while the nonlinear term $\vec{F}(\vec{U})$ derived from WENO approximations to the advection term is nonstiff but couples numerical values at grid points of the stencil. Hence we use different methods to treat them and avoid solving a large coupled nonlinear system. The advection term is treated explicitly while implicit approach is used for the reaction term.

Denote the numerical approximation to $\vec{U}(t_n)$ by $\vec{U}_n$, and let $\vec{G}_1(\tau) \triangleq e^{-C\tau}\vec{F}(\vec{U}(t_n+\tau))$ and $\vec{G}_2(\tau) \triangleq e^{-C\tau}\vec{R}(\vec{U}(t_n+\tau))$. If we use the constants $\vec{G}_1(0)$ to approximate $\vec{G}_1(\tau)$ and $\vec{G}_2(\Delta t)$ to approximate $\vec{G}_2(\tau)$ inside the integral of (13) as following

$$\vec{U}(t_{n+1}) \approx e^{C\Delta t}\vec{U}(t_n) + e^{C\Delta t}\int_0^{\Delta t} e^{-C\cdot 0}\vec{F}(\vec{U}(t_n+0)) + e^{-C\Delta t}\vec{R}(\vec{U}(t_n+\Delta t))d\tau, \tag{14}$$

then the first order single-step IIF scheme (**SIIF1**) is obtained

$$\vec{U}_{n+1} = e^{C\Delta t}\vec{U}_n + \Delta t e^{C\Delta t}\vec{F}(\vec{U}_n) + \Delta t\vec{R}(\vec{U}_{n+1}). \tag{15}$$

Note that the implicit term $\vec{R}(\vec{U}_{n+1})$ is free of matrix exponential operation. To construct a second order single-step IIF (SIIF2) scheme, analogous to the Runge–Kutta methods we introduce a middle stage $\vec{H}_n$ which is given by the first order explicit integration factor (EIF1) scheme

$$\vec{H}_n = e^{C\Delta t}\vec{U}_n + \Delta t e^{C\Delta t}\vec{F}(\vec{U}_n) + \Delta t e^{C\Delta t}\vec{R}(\vec{U}_n). \tag{16}$$

$\vec{H}_n$ is a first order approximation to $\vec{U}(t_{n+1})$. A linear polynomial interpolation is constructed to approximate the integrand in (13) as following. $\vec{U}_n$ and $\vec{H}_n$ are used as interpolation values of $\vec{U}(t)$ at the points t_n and $t_n + \Delta t$ for $\vec{G}_1(\tau)$ which involves the advection term. $\vec{U}_n$ and $\vec{U}_{n+1}$ are used as interpolation values of $\vec{U}(t)$ at the points t_n and $t_n + \Delta t$ for $\vec{G}_2(\tau)$ which involves the stiff reaction term. Hence we have

$$\vec{G}_1(\tau) + \vec{G}_2(\tau) \approx (\vec{F}+\vec{R})(\vec{U}_n) + \frac{\tau}{\Delta t}\left[e^{-C\Delta t}(\vec{F}(\vec{H}_n) + \vec{R}(\vec{U}_{n+1})) - (\vec{F}+\vec{R})(\vec{U}_n)\right]. \tag{17}$$

Substituting (17) to (13), we obtain the **SIIF2** scheme

$$\vec{U}_{n+1} = e^{C\Delta t}\vec{U}_n + \frac{\Delta t}{2}\left[\vec{F}(\vec{H}_n) + \vec{R}(\vec{U}_{n+1}) + e^{C\Delta t}(\vec{F}(\vec{U}_n) + \vec{R}(\vec{U}_n))\right]. \tag{18}$$

Again, the implicit term $\vec{R}(\vec{U}_{n+1})$ is free of matrix exponential operation.

A third order single-step IIF (SIIF3) scheme can be constructed in a similar way. As the procedure that the third order Runge–Kutta ETD method was derived in [7] analogous to the classical third-order Runge–Kutta method, first explicit methods are used to approximate the numerical values of $\vec{U}(t)$ at the points $t_n + \Delta t/2$ and $t_n + \Delta t$:

$$\vec{A}_n = e^{C\Delta t/2}\vec{U}_n + \frac{1}{2}\Delta t e^{C\Delta t/2}\left(\vec{F}(\vec{U}_n) + \vec{R}(\vec{U}_n)\right), \tag{19}$$

$$\vec{B}_n = e^{C\Delta t}\vec{U}_n + 2\Delta t e^{C\Delta t/2}(\vec{F}(\vec{A}_n) + \vec{R}(\vec{A}_n)) - \Delta t e^{C\Delta t}\left(\vec{F}(\vec{U}_n) + \vec{R}(\vec{U}_n)\right). \tag{20}$$

Then the SIIF2 scheme (18) is used to calculate a middle stage value $\vec{A}_{nr}$, an approximation to numerical value of $\vec{U}(t)$ at $t_n + \Delta t/2$,

$$\vec{A}_{nr} = e^{C\Delta t/2}\vec{U}_n + \frac{\Delta t}{4}\left[\vec{F}(\vec{A}_n) + \vec{R}(\vec{A}_{nr}) + e^{C\Delta t/2}\left(\vec{F}(\vec{U}_n) + \vec{R}(\vec{U}_n)\right)\right]. \tag{21}$$

Finally a quadratic interpolation is constructed to approximate the integrand in (13). $\vec{U}_n$, $\vec{A}_n$ and $\vec{B}_n$ are used as interpolation values of $\vec{U}(t)$ at the points t_n, $t_n + \Delta t/2$ and $t_n + \Delta t$ for $\vec{G}_1(\tau)$ which involves the advection term, while $\vec{U}_n$, $\vec{A}_{nr}$ and $\vec{U}_{n+1}$ are used as interpolation values of $\vec{U}(t)$ at the points t_n, $t_n + \Delta t/2$ and $t_n + \Delta t$ for $\vec{G}_2(\tau)$ which involves the stiff reaction term. Then we obtain the following **SIIF3** scheme

$$\begin{aligned}\vec{U}_{n+1} = e^{C\Delta t}\vec{U}_n + \frac{\Delta t}{6}\Big[&e^{C\Delta t}(\vec{F}(\vec{U}_n) + \vec{R}(\vec{U}_n)) + 4e^{C\Delta t/2}(\vec{F}(\vec{A}_n) + \vec{R}(\vec{A}_{nr})) \\ &+ (\vec{F}(\vec{B}_n) + \vec{R}(\vec{U}_{n+1}))\Big].\end{aligned} \tag{22}$$

The implicit terms in (22) and (21) are free of matrix exponential operations. Hence the local implicit property of the original multistep IIF schemes have been preserved here. Again in order to perform the computation for high spatial dimensional problems, we use Krylov subspace method described in the next section to approximate the product of a matrix exponential and a vector.

Remark. It is interesting to observe that if the reaction term $\vec{R}$ is set to be 0 in the single-step IIF schemes SIIF2 and SIIF3, the schemes reduce to the explicit IF schemes SEIF2 and SEIF3.

Remark. By taking the advection term $\vec{F}(\vec{U}) = 0$ in the SIIF2 and SIIF3 schemes, we obtain the single-step IIF schemes for solving stiff reaction–diffusion equations. The **SIIF2** scheme for reaction–diffusion equations is

$$U_{n+1} = e^{C\Delta t}U_n + \Delta t[\frac{1}{2}R(U_{n+1}) + \frac{1}{2}e^{C\Delta t}R(U_n)]. \tag{23}$$

This scheme is exactly the second order IIF scheme for stiff reaction–diffusion equations derived in [29]. We would like to emphasize that the general IIF schemes in our previous work including [29] are multistep schemes, except that the second order one happens to be a single-step scheme for reaction–diffusion equations without advection terms. The approach to derive the multistep schemes in our previous work is totally different from that in this paper for the single-step schemes. As that we can see, the single-step methods and multistep methods are different for advection–diffusion–reaction equations (even for the second order schemes), and they are also different in the third order scheme case even for relatively simple reaction–diffusion equations. The third order single-step IIF scheme **SIIF3** for stiff reaction–diffusion equations is as following:

$$\begin{aligned}A_{nr} &= e^{C\Delta t/2}U_n + \Delta t[\frac{1}{4}R(A_{nr}) + \frac{1}{4}e^{C\Delta t/2}R(U_n)], \\ U_{n+1} &= e^{C\Delta t}U_n + \frac{\Delta t}{6}e^{C\Delta t}R(U_n) + \frac{2}{3}\Delta t e^{C\Delta t/2}R(A_{nr}) + \frac{\Delta t}{6}R(U_{n+1}).\end{aligned} \tag{24}$$

The scheme is a single-step method and it is different from the third order multistep IIF scheme for stiff reaction diffusion equations derived in [29].

2.2.3. Krylov subspace approximations for matrix exponentials

Notice that we do *not* need the full exponential matrices such as $e^{C\Delta t}$ itself, but only the products of the exponential matrices and some vectors in the developed integration factor schemes. The Krylov subspace approximations to the matrix exponential operator is an excellent choice in terms of both accuracy and efficiency. Follow the literature (e.g. [10,28]), we describe the Krylov subspace methods to approximate $e^{C\Delta t}v$ as following.

The large sparse matrix C is projected to the Krylov subspace

$$K_M = \text{span}\{v, Cv, C^2v, \cdots, C^{M-1}v\}. \tag{25}$$

The dimension M of the Krylov subspace is **much** smaller than the dimension N of the large sparse matrix C. For most of numerical computations in this paper, we take $M = 25$ for different N, and accurate results are obtained in the numerical experiments. The well-known Arnoldi algorithm [35] generates an orthonormal basis $V_M = [v_1, v_2, v_3, \cdots, v_M]$ of the Krylov subspace K_M, and an $M \times M$ upper Hessenberg matrix H_M. And this the very small Hessenberg matrix H_M represents the projection of the large sparse matrix C to the Krylov subspace K_M, with respect to the basis V_M. Since the columns of V_M are orthonormal, we have the approximation

$$e^{C\Delta t}v \simeq \beta V_M e^{H_M \Delta t} e_1, \tag{26}$$

where $\beta = \|v\|_2$, and e_1 denotes the first column of the $M \times M$ identity matrix I_M. Thus the large $e^{C\Delta t}$ matrix exponential problem is replaced with a much smaller $e^{H_M\Delta t}$ problem. The small matrix exponential $e^{H_M\Delta t}$ will be computed using a scaling and squaring algorithm with a Padé approximation with only a small computational cost, see [15,28,10]. Then the Krylov approximations are directly applied in computations of the products of matrix exponentials and vectors in the single-step integration factor schemes here.

Remark. We would like to emphasize that in the implementation of the methods, we do *not* store matrices C, because only multiplications of matrices C with a vector are needed in the methods, and they correspond to certain finite difference operations. This implementation technique has been done for the multistep schemes in [19].

Remark. We would like to emphasize that in our Krylov single-step IIF-WENO schemes for ADR systems, the "local implicit" property of the original IIF schemes in [29] is preserved well. Namely, the implicit terms are free of matrix exponential operations. As a result, the implicit nonlinear system is decoupled for each spatial grid point. The size of the implicit nonlinear system at every spatial grid point only depends on the number of the original PDEs. This "local implicit" property provides a key factor for the computational efficiency of our Krylov single-step IIF-WENO schemes. The small size implicit nonlinear systems can be efficiently solved by either a fixed-point iteration method or a Newton iteration method. Also the "local implicit" property reduces the amount of information exchange of different regions of the domain, hence offers some advantages when the methods are parallelized.

Remark. For a one dimensional linear diffusion equation with constant diffusion coefficient and periodic boundary condition, a second order central difference approximation to the diffusion term du_{xx} leads to a differential matrix C with three nonzero elements $d/h^2, -2d/h^2, d/h^2$ in each row. Eigenvalues of the matrix C are

$$\lambda_k = -\frac{4d}{h^2}\sin^2\left(\frac{(k-1)\pi}{N}\right), \qquad k = 1, 2, \cdots, N.$$

Here N is the number of grid points. By the analysis theorems provided in [10,16], an error estimation of the Krylov subspace approximation (26) for this case is

$$\|e^{C\Delta t}v - \beta V_M e^{H_M\Delta t}e_1\|_2 \le 10\beta e^{-M^2/(5\rho\Delta t)}, \tag{27}$$

where $M = 25$ is the dimension of the Krylov subspace, and eigenvalues of the matrix C are in the interval $[-4\rho, 0]$. Hence $\rho = d/h^2$ for this case. We can see that with a fixed Courant number $\rho\Delta t$ and $M = 25$, the negative power in the exponential term of the error (27) leads to small approximation errors.

Remark. As that discussed in [16], the convergence to $e^{C\Delta t}v$ by Krylov subspace approximation is more efficient than that of corresponding Krylov methods for the solution of linear system $(I - \Delta t A)x = v$, which arises in the implicit schemes for diffusion equations or other PDEs with high order derivatives. Since there are many fast linear system solvers which can be used to accelerate computations of implicit schemes, it will be one of our future work to perform systematical comparisons of different methods.

3. Linear analysis

Similar as that in [29,19] for multistep methods, we perform linear analysis for the single-step integration factor methods derived in the last section.

3.1. Linear analysis of SEIF schemes

3.1.1. Truncation error

We focus on analyzing the truncation errors of the SEIF2 scheme (10) and the SEIF3 scheme (11), i.e., the local temporal truncation errors. Consider the following linear semi-discretization system

$$\frac{d\mathbf{u}}{dt} = C\mathbf{u} + F\mathbf{u}, \tag{28}$$

where C and F are matrices derived from linear spatial discretizations of a linear ADR system, and $\mathbf{u}$ is the vector of unknown numerical values. First, we apply the SEIF2 scheme (10) to the system (28) to obtain $\mathbf{u}_{n+1}$ in terms of $\mathbf{u}_n$:

$$\mathbf{u}_{n+1} = e^{C\Delta t}(I + \frac{F}{2}\Delta t)\mathbf{u}_n + \frac{F}{2}\Delta t e^{C\Delta t}(I + \Delta t F)\mathbf{u}_n. \tag{29}$$

To derive the local truncation error, we substitute the exact solution of (28) into the right hand side of the equation (29) and use Taylor expansion. Denoting the exact solution of (28) by $\mathbf{u}(t)$, we replace $\mathbf{u}_n$ by the exact solution value $\mathbf{u}(t_n)$ in (29) and obtain

$$\mathbf{u}_{n+1} = (I + (C+F)\Delta t + (\frac{C^2}{2} + \frac{F^2}{2} + \frac{CF}{2} + \frac{FC}{2})\Delta t^2 + \ldots)\mathbf{u}(t_n). \tag{30}$$

Hence, the local truncation error of the SEIF2 method (10) is

$$\left(I + (C+F)\Delta t + \frac{(C+F)^2}{2}\Delta t^2 + \ldots\right)\mathbf{u}(t_n) - e^{(C+F)\Delta t}\mathbf{u}(t_n) = O(\Delta t^3)\mathbf{u}(t_n). \tag{31}$$

Similarly for the third order scheme SEIF3 (11), we apply it to the system (28) and obtain

$$\begin{aligned}\mathbf{u}_{n+1} = {} & e^{C\Delta t}(I + \frac{\Delta t}{6}F)\mathbf{u}_n + \frac{\Delta t}{6}Fe^{C\Delta t}(I - \Delta t F)\mathbf{u}_n \\ & + \frac{\Delta t}{3}e^{C\Delta t/2}Fe^{C\Delta t/2}(2I + \Delta t F)\mathbf{u}_n + \frac{F}{3}\Delta t^2 e^{C\Delta t/2}Fe^{C\Delta t/2}(I + \frac{\Delta t}{2}F)\mathbf{u}_n.\end{aligned} \tag{32}$$

Using Taylor expansion and replacing $\mathbf{u}_n$ by the exact solution value $\mathbf{u}(t_n)$, we have

$$\mathbf{u}_{n+1} = \left(I + (C+F)\Delta t + \frac{(C+F)^2}{2}\Delta t^2 + \frac{(C+F)^3}{6}\Delta t^3 + \ldots\right)\mathbf{u}(t_n). \tag{33}$$

Hence, the local truncation error of the SEIF3 method (11) is

$$\begin{aligned}&\left(I + (C+F)\Delta t + \frac{(C+F)^2}{2}\Delta t^2 + \frac{(C+F)^3}{6}\Delta t^3 + \ldots\right)\mathbf{u}(t_n) \\ &\quad - e^{(C+F)\Delta t}\mathbf{u}(t_n) = O(\Delta t^4)\mathbf{u}(t_n).\end{aligned} \tag{34}$$

3.1.2. Linear stability

To analyze the linear stability of the SEIF methods, we consider the following scalar linear test equation

$$u_t = -du + au, \qquad \text{with } d > 0. \tag{35}$$

Similar to the stability analysis approaches in [29,19], we will show boundaries of the stability regions in the complex plane for $a\Delta t$, a family of curves for different values of $d\Delta t$, for the second and third order methods. In the context of solving ADR equations with nonstiff reaction terms, a and d actually represent spatial discretizations for the nonstiff advection–reaction terms and the stiff diffusion term respectively.

Applying the second order SEIF2 scheme (10) to the equation (35), then substituting $u_n = e^{in\theta}$ into the resulting equation, we obtain

$$e^{i\theta} = e^{-d\Delta t}(1 + \lambda + \frac{\lambda^2}{2}), \tag{36}$$

where $\lambda = a\Delta t$ has a real part λ_r and imaginary part λ_i. (36) is solved for λ_r and λ_i, and then the boundaries of the stability regions are obtained by varying θ from 0 to 2π. Stability regions are shown in the complex plane of $a\Delta t$ for different values of $d\Delta t$. We choose four different values of $d\Delta t$, $d\Delta t = 1.0$, $d\Delta t = 2.0$, $d\Delta t = 10.0$ and $d\Delta t = 20.0$ as examples. Analysis of the amplification factors of the scheme shows that the special point $\lambda = (0, 0)$ is always included in the stable regions for the SEIF2 scheme. See Fig. 1 for the stability regions. The stable regions are inside the stability boundaries. From Fig. 1 we see that the stable regions become bigger with the increase of the value of $d\Delta t$. This shows that the diffusion term tends to stabilize the SEIF2 scheme.

Next, we analyze the third order scheme SEIF3 (11). Using the same approach, we apply the SEIF3 scheme (11) to the equation (35), then substitute $u_n = e^{in\theta}$ into the resulting equation and obtain the equation for λ (i.e., $a\Delta t$):

$$e^{i\theta} = e^{-d\Delta t}(1 + \lambda + \frac{\lambda^2}{2} + \frac{\lambda^3}{6}). \tag{37}$$

The equation (37) is solved numerically for λ_r and λ_i with specific values of θ and $d\Delta t$, and then the boundaries of the stability regions are obtained by varying θ from 0 to 2π. Stability regions are plotted in the complex plane of $a\Delta t$ for

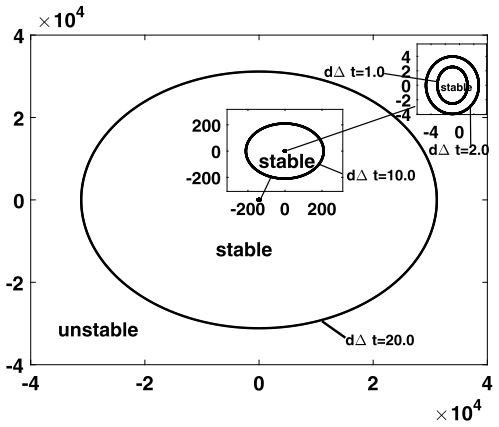

Fig. 1. Linear stability regions of the SEIF2 scheme (10) for different values of $d\Delta t$. Range of $d\Delta t$: $1.0 \sim 20.0$. Due to significant differences of stability regions for different $d\Delta t$, "zoomed in" pictures (the ones in boxes) are presented.

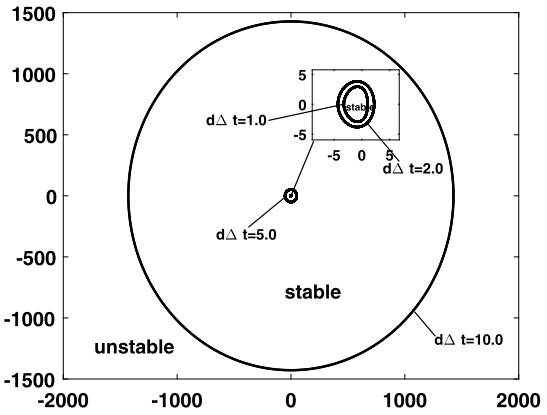

Fig. 2. Linear stability regions of the SEIF3 scheme (11) for different values of $d\Delta t$. Range of $d\Delta t$: $1.0 \sim 10.0$. Due to significant differences of stability regions for different $d\Delta t$, "zoomed in" pictures (the ones in the box) are presented.

different values of $d\Delta t$, $d\Delta t = 1.0$, $d\Delta t = 2.0$, $d\Delta t = 5.0$ and $d\Delta t = 10.0$. Since the point $\lambda = (0, 0)$ is always in the stable regions for the SEIF3 scheme, the interior of the regions bounded by the curves in Fig. 2 are stable regions. From Fig. 2 we see that the stable regions become larger with the increase of the value of $d\Delta t$. Again, this shows that the diffusion term tends to stabilize the scheme.

3.2. Linear analysis of SIIF schemes

3.2.1. Truncation error

In this section we verify the truncation errors of implicit schemes SIIF2 and SIIF3. Consider the following linear semi-discretization system

$$\frac{d\mathbf{u}}{dt} = C\mathbf{u} + F\mathbf{u} + R\mathbf{u}, \tag{38}$$

where C, F and R are matrices derived from linear spatial discretizations of diffusion, advection and reaction terms of a linear ADR system respectively, and $\mathbf{u}$ is the vector of unknown numerical values. First, we apply the SIIF2 scheme (18) to the system (38) to obtain $\mathbf{u}_{n+1}$ in terms of $\mathbf{u}_n$:

$$\mathbf{u}_{n+1} = e^{C\Delta t}(I + \frac{F}{2}\Delta t + \frac{R}{2}\Delta t)\mathbf{u}_n + \frac{\Delta t}{2}Fe^{C\Delta t}(I + F\Delta t + R\Delta t)\mathbf{u}_n + \frac{\Delta t}{2}R\mathbf{u}_{n+1}. \tag{39}$$

Substituting the exact solution of (38) into the right hand side of the equation (39) and using Taylor expansion, we obtain

$$\begin{aligned}\mathbf{u}_{n+1} &= \left(I - \frac{R}{2}\Delta t\right)^{-1}\left[e^{C\Delta t}\left(I + \frac{F}{2}\Delta t + \frac{R}{2}\Delta t\right) + \frac{F}{2}\Delta t e^{C\Delta t}(I + F\Delta t + R\Delta t)\right]\mathbf{u}(t_n)\\ &= \left[I + (C + F + R)\Delta t + \frac{(C + F + R)^2}{2}\Delta t^2 + \ldots\right]\mathbf{u}(t_n).\end{aligned} \tag{40}$$

Hence the local truncation error of the SIIF2 method (18) is

$$\mathbf{u}_{n+1} - e^{(C+F+R)\Delta t}\mathbf{u}(t_n) = (O(\Delta t^3))\mathbf{u}(t_n). \tag{41}$$

Similarly for the third order scheme SIIF3 (22), we apply it to the system (38) to obtain

$$a_n = e^{C\Delta t/2}\left(I + \frac{\Delta t}{2}(F+R)\right)\mathbf{u}_n, \tag{42}$$

$$b_n = e^{C\Delta t}\left(I - \Delta t(F+R)\right)\mathbf{u}_n + 2\Delta t e^{C\Delta t/2}(F+R)(a_n), \tag{43}$$

$$a_{nr} = \left(I - \frac{R}{4}\Delta t\right)^{-1}\left[e^{C\Delta t/2}\left(\mathbf{u}_n + \frac{\Delta t}{4}(F+R)\mathbf{u}_n\right) + \frac{\Delta t}{4}F(a_n)\right]. \tag{44}$$

Then we substitute a_n, b_n and a_{nr} into (22) and obtain $\mathbf{u}_{n+1}$:

$$\mathbf{u}_{n+1} = \left(I - \frac{R}{6}\Delta t\right)^{-1}\left[e^{C\Delta t}(I + \frac{\Delta t}{6}(F+R))\mathbf{u}_n + \frac{2}{3}\Delta t e^{C\Delta t/2}(F(a_n) + R(a_{nr})) + \frac{\Delta t}{6}F(b_n)\right]. \tag{45}$$

Again, substituting the exact solution of (38) into the right hand side of the equation (45) and using Taylor expansion, we find that the local truncation error of the SIIF3 scheme (22) is

$$\begin{aligned}\mathbf{u}_{n+1} - e^{(C+F+R)\Delta t}\mathbf{u}(t_n) &= \left(\frac{F^4\Delta t^4}{24} + \frac{5F^3R\Delta t^4}{36} + \frac{7F^2R^2\Delta t^4}{48} + \frac{FR^3\Delta t^4}{24} - \frac{R^4\Delta t^4}{144} + O(\Delta t^5)\right)\mathbf{u}(t_n)\\ &= O(\Delta t^4)\mathbf{u}(t_n).\end{aligned} \tag{46}$$

3.2.2. Linear stability

In order to analyze the linear stability of the SIIF methods for ADR equations, we consider the following scalar linear test equation

$$u_t = au - du + ru, \qquad \text{with } r \in \mathcal{C}, \text{ and } a, d \in \mathcal{R}, \ d > 0. \tag{47}$$

We will show boundaries of the stability regions in the complex plane for $r\Delta t$, a family of curves for different values of $d\Delta t$ and $a\Delta t$. In the context of solving advection–diffusion–reaction equation, a and d actually represent spatial discretizations for the advection term and the diffusion term respectively. We first analyze the second order scheme. Applying the SIIF2 scheme (18) to the equation (47) and substituting $u_n = e^{in\theta}$ into the resulting equation, we obtain

$$(1 - \frac{\lambda}{2})e^{i\theta} = e^{-d\Delta t}(1 + a\Delta t + 0.5a^2\Delta t^2 + 0.5a\lambda\Delta t + 0.5\lambda) \tag{48}$$

where $\lambda = r\Delta t$ has a real part λ_r and an imaginary part λ_i. λ_r and λ_i are calculated numerically from the equation (48) for certain $a\Delta t$ and $d\Delta t$. Stability regions in the complex plane of $r\Delta t$ for different values of $d\Delta t$ under a fixed value of $a\Delta t = 1.0$ are shown in Fig. 3. Based on analyzing the amplification factors of the scheme, it is found that the stable regions always include the special point $\lambda = (-1, 0)$ for the cases shown in Fig. 3. As a result, regions inside the stability boundary curves shown in Fig. 3 are unstable regions. In Fig. 3, we observe that when $a\Delta t$ is fixed, with the increase of the value of $d\Delta t$, the stable regions become larger. Again, the diffusion term tends to stabilize the scheme. Next, we fix the value $d\Delta t = 1.0$ and choose different values for $a\Delta t$: $a\Delta t = -5.0$, $a\Delta t = -1.0$, $a\Delta t = 1.0$ and $a\Delta t = 5.0$ as examples to show the stability regions in Fig. 4. Based on analysis of the amplification factors of the scheme for the specific cases of $a\Delta t = -5.0$ and 5.0, we find that the special point $\lambda = (0, 0)$ is in the unstable regions. So regions inside the stability boundary curves are stable regions for such cases. When $a\Delta t = -1.0$ and 1.0, the point $\lambda = (0, 0)$ is in the stable regions. So regions inside the stability boundary curves are unstable regions for these cases. In conclusion, for a fixed $d\Delta t$ with the decrease of $|a\Delta t|$ which corresponds to advection terms, the stability regions become larger.

Similar analysis has been done for the third order scheme. Applying the SIIF3 scheme (22) to the equation (47) and substituting $u_n = e^{in\theta}$ into the resulting equation, we obtain

$$\begin{aligned}(1 - \frac{\lambda}{6})e^{i\theta} = e^{-d\Delta t}[&1 + a\Delta t + \frac{1}{2}(a\Delta t)^2 + \frac{1}{6}(a\Delta t)^3 + \frac{1}{6}\lambda + \frac{1}{2}\lambda a\Delta t + \frac{1}{3}\lambda(a\Delta t)^2\\ &+ \frac{1}{6}\lambda^2 a\Delta t + \frac{\frac{2}{3}\lambda + \frac{1}{3}a\Delta t\lambda + \frac{1}{12}(a\Delta t)^2\lambda + \frac{1}{12}a\Delta t\lambda^2 + \frac{1}{6}\lambda^2}{1 - 0.25\lambda}],\end{aligned} \tag{49}$$

where $\lambda = r\Delta t$ has a real part λ_r and an imaginary part λ_i. λ_r and λ_i are calculated numerically from the equation (49) for certain $a\Delta t$ and $d\Delta t$ using MATLAB. Stability regions in the complex plane of $r\Delta t$ for different values of $d\Delta t$ under a fixed value of $a\Delta t = 1.0$ are shown in Fig. 5. Based on analyzing the amplification factors of the scheme for special values of λ, it is found that the stable regions always include the point $\lambda = (-6, 0)$ for the cases shown in Fig. 5. As a result, stable and

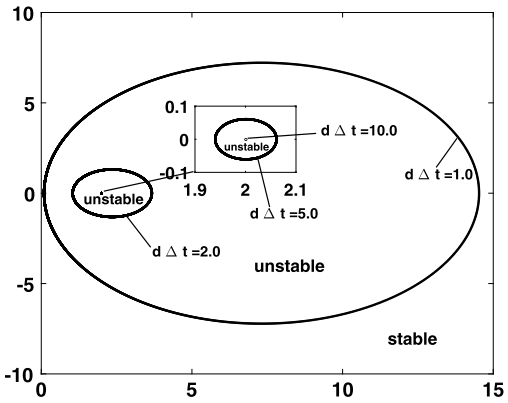

Fig. 3. Linear stability regions of SIIF2 (18) for different values of $d\Delta t$ under a fixed value of $a\Delta t = 1.0$. Range of $d\Delta t : 1.0 \sim 10.0$. Due to significant differences of stability regions for different $d\Delta t$, "zoomed in" pictures (the ones in the box) are presented.

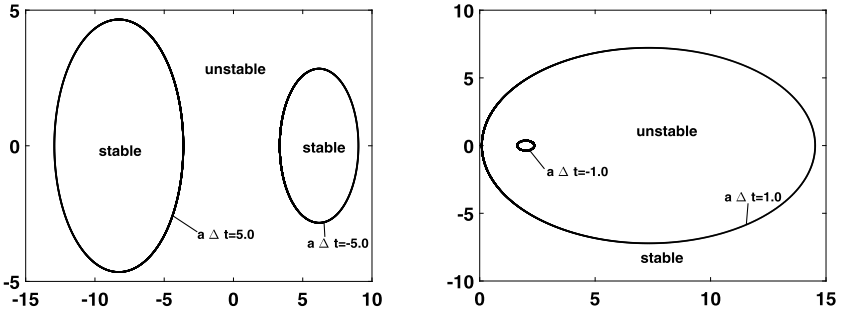

Fig. 4. Linear stability regions of SIIF2 scheme (18) for different values of $a\Delta t$ under a fixed value of $d\Delta t = 1.0$. Range of $a\Delta t : -5.0 \sim 5.0$. Left: $a\Delta t = -5.0$ and 5.0. Regions inside the stability boundary curves are stable regions for such cases. Right: $a\Delta t = -1.0$ and 1.0. Regions outside the stability boundary curves are stable regions for such cases.

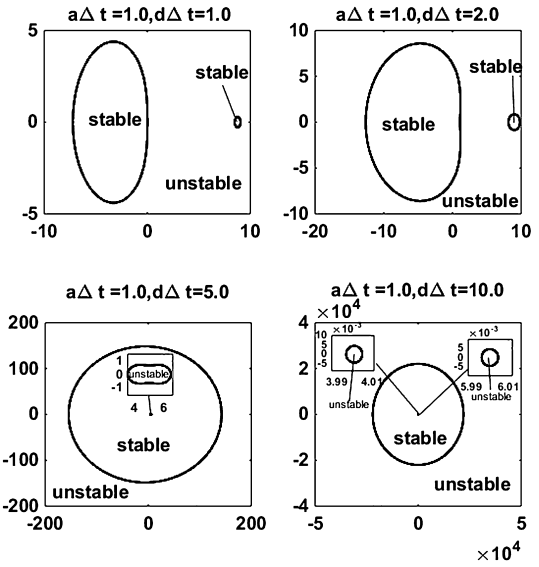

Fig. 5. Linear stability regions of SIIF3 (22) for different values of $d\Delta t$ under a fixed value of $a\Delta t = 1.0$. Range of $d\Delta t : 1.0 \sim 10.0$.

unstable regions are identified as shown in Fig. 5. It is interesting to see there exist some isolated and very small unstable or stable regions in the complex plane. When $a\Delta t$ is fixed, we observe that with the increase of the value of $d\Delta t$, the stable regions become larger. Similar to SIIF2, the diffusion term tends to stabilize the scheme. Next, we fix the value $d\Delta t = 1.0$ and choose different values for $a\Delta t$: $a\Delta t = -5.0$, $a\Delta t = -1.0$, $a\Delta t = 1.0$ and $a\Delta t = 5.0$ as examples to show the stability regions in Fig. 6. Since the special point $\lambda = (4, 0)$ is in the unstable regions for these cases by analysis of the amplification

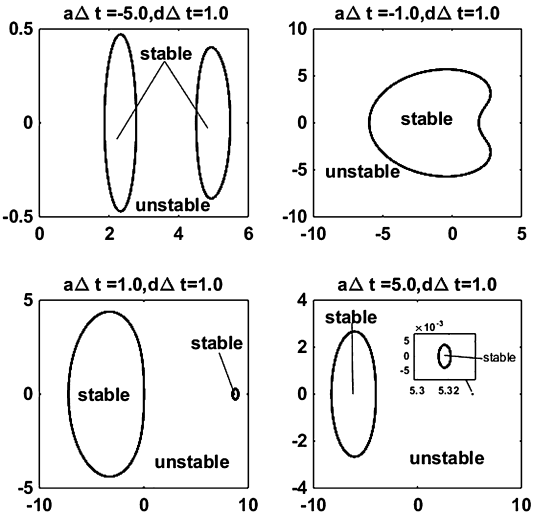

Fig. 6. Linear stability regions of SIIF3 scheme (22) for different values of $a\Delta t$ under a fixed value of $d\Delta t = 1.0$. Range of $a\Delta t : -5.0 \sim 5.0$.

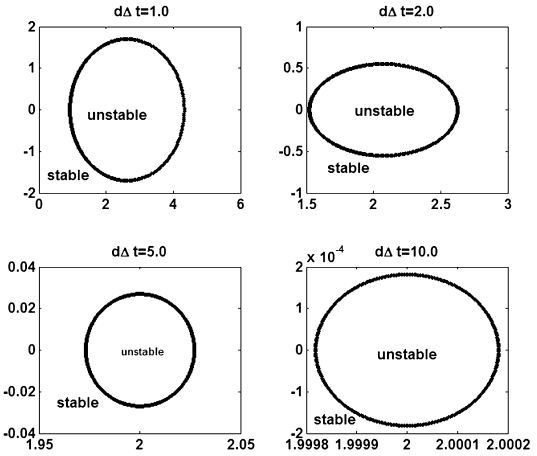

Fig. 7. Linear stability regions of the SIIF2 scheme (23) for different values of $d\Delta t$. Range of $d\Delta t : 1.0 \sim 10.0$.

factors of the scheme, stable and unstable regions are determined by including $\lambda = (4, 0)$ or not in regions separated by stability boundary curves in Fig. 6. In Fig. 6, we find that for a fixed $d\Delta t$, with the decrease of $|a\Delta t|$, the stability regions become larger. Stable regions shrink significantly when advection becomes strong and dominates. When we compare the stability regions of SIIF2 and SIIF3, we observe that in general, SIIF2 is more stable than SIIF3.

Remark. We present the special cases, i.e., the linear stability analysis of the SIIF methods (23) and (24) for reaction–diffusion equations here. Consider the following scalar linear test equation

$$u_t = -du + ru, \qquad \text{with } r \in \mathcal{C}, \text{and } d \in \mathcal{R}, d > 0. \tag{50}$$

We will show boundaries of the stability regions in the complex plane for $r\Delta t$, a family of curves for different values of $d\Delta t$. In the context of solving reaction–diffusion equation, d actually represents spatial discretizations for the diffusion. We first analyze the second order scheme. Applying the SIIF2 scheme (23) to the equation (50) and substituting $u_n = e^{in\theta}$ into the resulting equation, we obtain

$$(1 - \frac{\lambda}{2})e^{i\theta} = e^{-d\Delta t}(1 + 0.5\lambda), \tag{51}$$

where $\lambda = r\Delta t$ has a real part λ_r and an imaginary part λ_i. λ_r and λ_i are calculated numerically from the equation (51) for certain $d\Delta t$. Stability regions in the complex plane of $r\Delta t$ for different values of $d\Delta t$ are shown in Fig. 7. Based on analyzing the amplification factors of the scheme, the stable regions always include the special point $\lambda = (0, 0)$ for the cases shown in Fig. 7. As a result, regions inside the stability boundary curves shown in Fig. 7 are unstable regions. In Fig. 7, we observe that with the increase of the value of $d\Delta t$, the stable regions become larger. Hence the diffusion term tends to stabilize the

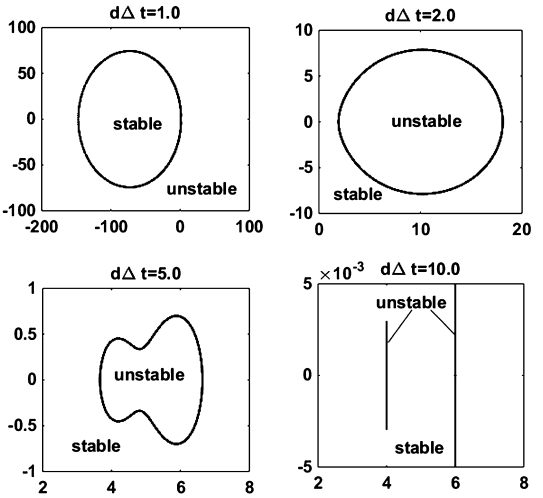

Fig. 8. Linear stability regions of the SIIF3 scheme (24) for different values of $d\Delta t$. Range of $d\Delta t : 1.0 \sim 10.0$.

scheme as that in the previous schemes discussed in this section. Similar analysis has been done for the third order scheme. Applying the SIIF3 scheme (24) to the equation (50) and substituting $u_n = e^{in\theta}$ into the resulting equation, we obtain

$$(1 - \frac{\lambda}{6})e^{i\theta} = e^{-d\Delta t}\left(1 + \frac{1}{6}\lambda + \frac{\frac{2}{3}\lambda + \frac{1}{6}\lambda^2}{1 - 0.25\lambda}\right), \tag{52}$$

where $\lambda = r\Delta t$ has a real part λ_r and an imaginary part λ_i. λ_r and λ_i are calculated numerically from the equation (52) for certain $d\Delta t$ using MATLAB. Stability regions in the complex plane of $r\Delta t$ for different values of $d\Delta t$ are shown in Fig. 8. Based on analyzing the amplification factors of the scheme for special values of λ, the stable regions always include the point $\lambda = (0, 0)$ for the cases shown in Fig. 8. Stable and unstable regions are identified as shown in Fig. 8. When $d\Delta t = 10.0$, the unstable regions which look like two segments are actually two very thin enclosed regions. We observe that with the increase of the value of $d\Delta t$, the stable regions become larger. Similar to SIIF2, the diffusion term tends to stabilize the scheme. When we compare the stability regions of SIIF2 and SIIF3, we observe that in general, SIIF2 is more stable than SIIF3.

4. Numerical experiments

In this section, we present numerical examples to show the stability, accuracy and efficiency of the Krylov single-step integration factor WENO methods for solving advection–diffusion–reaction PDEs. Different aspects of the methods are tested using various examples to show the ability of the methods in dealing with high spatial dimensions, stiff reactions, and convection-dominated cases. Comparisons of single-step and multistep methods are performed. The Krylov subspace dimension M is taken to be $M = 25$ in all examples unless otherwise indicated for some tests. From numerical experiments we can observe that large time step sizes are achieved in numerical computations of these advection–diffusion–reaction equations. Here large time step size means that the time step size is mainly restricted by the CFL condition constraint of the nonstiff advection term (the hyperbolic part), which is treated explicitly. Hence the time step size $\Delta t = O(\Delta x)$ as that in solving a pure hyperbolic PDE. The second order central scheme is used for the diffusion terms when second order integration factor methods are applied, and the fourth order central scheme for the diffusion terms is coupled with third order integration factor schemes. All of the numerical simulations in this paper are performed on a 2.3 GHz, 2 GB RAM Linux workstation.

Example 1 *(Tests on high spatial dimensions).* A major challenge in using integration factor type methods is how to efficiently compute large matrix exponentials for problems with high spatial dimensions. Krylov subspace approximations are applied in the schemes developed in this paper, following early work on this issue [5,19]. We test the performance of our schemes in dealing with problems with two and three spatial dimensions in the first example.

The two spatial dimension equation is

$$\begin{cases} u_t + (\frac{1}{2}u^2)_x + (\frac{1}{2}u^2)_y = u_{xx} + u_{yy} + 2u + \cos(x + y + t)(1 + 2\sin(x + y + t)), \\ u(x, y, 0) = \sin(x + y), \qquad 0 \le x, y \le 2\pi, \end{cases} \tag{53}$$

with periodic boundary conditions, and the exact solution is $u(x, y, t) = \sin(x + y + t)$. We first test the single step implicit integration factor methods. The computation is carried up to $T = 1.0$ at which the L^1 and L^∞ errors are measured. We study the performance of the new single-step IIF methods by comparing it with Krylov multistep IIF (MIIF) methods developed in

Table 1
Example 1, the two-dimensional equation (53). CPU time, error, and order of accuracy of Krylov SIIF2, Krylov MIIF2 and RK2 methods with the third order WENO spatial discretization for the advection term. Final time $T = 1.0$. N is the number of grid points in each spatial direction.

N	Krylov SIIF2, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	2.58E−2	–	1.61E−2	–	1.10
40	7.00E−3	1.88	4.00E−3	2.01	4.38
80	2.10E−3	1.74	1.20E−3	1.74	20.87
160	5.73E−4	1.87	3.23E−4	1.89	177.87
320	1.47E−4	1.96	8.22E−5	1.97	1370.01
N	Krylov MIIF2, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	5.22E−2	–	2.39E−2	–	1.19
40	2.05E−2	1.35	9.80E−3	1.29	3.12
80	6.10E−3	1.75	3.00E−3	1.71	12.71
160	1.80E−3	1.76	8.84E−4	1.76	74.46
320	4.55E−4	1.98	2.31E−4	1.94	616.64
N	RK2, $\Delta t = 0.1\Delta x^2$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	6.75E−3	–	3.22E−3	–	1.60
40	4.33E−3	0.64	1.89E−3	0.77	20.91
80	1.27E−3	1.77	5.65E−4	1.74	334.64
160	3.32E−4	1.94	1.49E−4	1.92	5302.80
320	8.50E−5	1.97	3.80E−5	1.97	85 627.05

Table 2
Example 1, the two-dimensional equation (53). CPU time, error, and order of accuracy of Krylov SIIF3, Krylov MIIF3 and RK3 methods with the third order WENO spatial discretization for the advection term. Final time $T = 1.0$. N is the number of grid points in each spatial direction.

N	Krylov SIIF3, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	1.09E−2	–	5.88E−3	–	7.61
40	8.89E−4	3.62	4.75E−4	3.63	27.37
80	8.80E−5	3.34	4.80E−5	3.31	130.48
160	1.02E−5	3.11	5.60E−6	3.10	1143.38
320	1.26E−6	3.02	6.91E−7	3.02	7140.60
N	Krylov MIIF3, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	4.13E−2	–	2.25E−2	–	0.94
40	1.11E−2	1.90	5.40E−3	2.06	3.68
80	2.10E−3	2.40	9.66E−4	2.48	18.91
160	3.23E−4	2.70	1.49E−4	2.70	130.29
320	4.43E−5	2.87	2.03E−5	2.88	1047.70
N	RK3, $\Delta t = 0.2\Delta x^2$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	2.11E−2	–	1.17E−2	–	2.24
40	1.63E−3	3.69	9.00E−4	3.70	31.91
80	1.21E−4	3.75	6.70E−5	3.75	497.20
160	1.10E−5	3.46	6.00E−6	3.48	8055.46
320	1.30E−6	3.08	7.20E−7	3.06	127 839.7

[19] and other single-step methods such as Runge–Kutta (RK) methods. CPU times, numerical errors, and orders of accuracy of the second order schemes including the Krylov SIIF2, the Krylov MIIF2 and RK2 methods with the third order WENO spatial discretization for the advection terms are reported in Table 1. Table 2 reports the results for the same equation using the third order schemes including the Krylov SIIF3, the Krylov MIIF3 and RK3 methods. Desired accuracy orders for the new Krylov SIIF2 and Krylov SIIF3 schemes, and other existing methods are obtained as shown in Table 1 and Table 2. For our new schemes Krylov SIIF2 and SIIF3, and the existing multistep methods Krylov MIIF2 and MIIF3, the time step sizes can reach $\Delta t = O(\Delta x)$ due to large stability regions of these methods. This is consistent with our goal of achieving a large time step size proportional to the spatial grid size for a stable and accurate computation of a parabolic PDE. However, for the regular Runge–Kutta methods, $\Delta t = O(\Delta x)^2$ is needed for the stability of the computations. We use the largest time step sizes which can result in stable computations by Runge–Kutta methods for examples in this paper. From the data in the tables, we can observe that in this example the Krylov SIIF methods need more CPU time than the Krylov MIIF methods on a

Table 3
Example 1, the two-dimensional equation (53). CPU time, error, and order of accuracy of Krylov SEIF2, Krylov MEIF2, and RK2 methods with the third order WENO spatial discretization for the advection term. Final time $T = 1.0$. N is the number of grid points in each spatial direction.

N	Krylov SEIF2, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	5.04E−2	–	2.43E−2	–	0.24
40	8.80E−3	2.52	4.20E−3	2.53	1.40
80	2.20E−3	2.00	9.06E−4	2.22	10.70
160	5.54E−4	1.99	2.26E−4	2.00	82.33
320	1.39E−4	1.99	5.65E−5	2.00	669.37
N	Krylov MEIF2, $\Delta t = 0.4\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	7.05E−2	–	3.96E−2	–	0.97
40	1.87E−2	1.91	9.80E−3	2.01	6.09
80	4.90E−3	1.93	2.50E−3	1.97	41.54
160	1.30E−3	1.91	6.21E−4	2.01	196.04
320	3.17E−4	2.04	1.57E−4	1.98	1573.4
N	RK2, $\Delta t = 0.1\Delta x^2$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	6.75E−3	–	3.22E−3	–	1.60
40	4.33E−3	0.64	1.89E−3	0.77	20.91
80	1.27E−3	1.77	5.65E−4	1.74	334.64
160	3.32E−4	1.94	1.49E−4	1.92	5302.80
320	8.50E−5	1.97	3.80E−5	1.97	85 627.05

Table 4
Example 1, the two-dimensional equation (53). CPU time, error, and order of accuracy of Krylov SEIF3, Krylov MEIF3, and RK3 methods with the third order WENO spatial discretization for the advection term. Final time $T = 1.0$. N is the number of grid points in each spatial direction.

N	Krylov SEIF3, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	1.26E−2	–	6.70E−3	–	0.55
40	1.00E−3	3.66	5.80E−4	3.53	2.72
80	1.08E−4	3.21	6.13E−5	3.24	17.93
160	1.26E−5	3.10	7.32E−6	3.07	130.99
320	1.56E−6	3.01	9.05E−7	3.02	1087.13
N	Krylov MEIF3, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	2.70E−2	–	1.58E−2	–	1.12
40	4.20E−3	2.68	2.40E−3	2.72	7.52
80	5.39E−4	2.96	2.85E−4	3.07	25.88
160	7.15E−5	2.91	3.58E−5	2.99	148.40
320	9.32E−6	2.94	4.61E−6	2.96	1113.10
N	RK3, $\Delta t = 0.2\Delta x^2$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	2.11E−2	–	1.17E−2	–	2.24
40	1.63E−3	3.69	9.00E−4	3.70	31.91
80	1.21E−4	3.75	6.70E−5	3.75	497.20
160	1.10E−5	3.46	6.00E−6	3.48	8055.46
320	1.30E−6	3.08	7.20E−7	3.06	127 839.7

same spatial mesh. This is due to the fact that single-step integration factor methods perform more matrix vector operations than multistep ones. However, the Krylov SIIF methods have smaller numerical errors than the Krylov MIIF methods on a same spatial mesh. So at the end, in general the Krylov SIIF methods spend less CPU time than the Krylov MIIF methods to reach similar level numerical errors in this example. And both methods are much more efficient than Runge–Kutta methods.

The reaction term in this example is not stiff. We also use this example to test Krylov explicit IF methods including Krylov SEIF methods and classical multistep explicit IF (MEIF) methods. MEIF schemes are commonly used in the literature (e.g. see [7] and its references). Here we apply Krylov approach to compute the matrix exponentials in the classical MEIF schemes. CPU time, error, and order of accuracy of the SEIF2, MEIF2 and RK2 methods with the third order WENO spatial discretization for the advection terms are reported in Table 3 for Equation (53). Table 4 includes the results for the same equation using third order schemes including SEIF3, MEIF3 and RK3 methods. We observe that the desired accuracy orders are obtained for all schemes. Again, similar conclusions as the implicit schemes on efficiency of different methods can be

Table 5
Example 1, the three-dimensional equation (54). CPU time, error, and order of accuracy of Krylov SEIF2, Krylov MEIF2, and RK2 methods with the third order WENO spatial discretization for the advection term. Final time $T = 1.0$. N is the number of grid points in each spatial direction.

N	Krylov SEIF2, $\Delta t = 0.4\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
10	2.49E−1	–	1.23E−1	–	2.18
20	5.84E−2	2.09	2.86E−2	2.10	22.77
40	1.26E−2	2.21	5.61E−3	2.35	135.40
80	3.14E−3	2.00	1.36E−3	2.04	1281.54
160	7.99E−4	1.97	3.45E−4	1.98	27 534.16
N	Krylov MEIF2, $\Delta t = 0.4\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
10	3.80E−1	–	2.30E−1	–	2.96
20	1.50E−1	1.34	8.85E−2	1.38	12.17
40	4.51E−2	1.73	2.40E−2	1.88	91.51
80	1.18E−2	1.93	6.10E−3	1.98	1255.70
160	3.00E−3	1.98	1.50E−3	2.02	20 955.00
N	RK2, $\Delta t = 0.15\Delta x^2$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
10	8.59E−2	–	4.77E−2	–	0.95
20	9.00E−3	3.25	4.37E−3	3.45	21.19
40	5.76E−3	0.64	2.58E−3	0.76	646.58
80	1.70E−3	1.76	7.71E−4	1.74	20 478.39
160	4.46E−4	1.93	2.04E−4	1.92	672 881.48

Table 6
Example 1, the three-dimensional equation (54). CPU time, error, and order of accuracy of Krylov SEIF3, Krylov MEIF3, and RK3 methods with the third order WENO spatial discretization for the advection term. Final time $T = 1.0$. N is the number of grid points in each spatial direction.

N	Krylov SEIF3, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
10	2.00E−1	–	1.30E−1	–	6.46
20	3.62E−2	2.47	2.03E−2	2.68	25.74
40	3.20E−3	3.50	1.80E−3	3.50	155.80
80	2.81E−4	3.51	1.64E−4	3.46	2322.20
160	3.00E−5	3.23	1.80E−5	3.19	50 275.00
N	Krylov MEIF3, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
10	1.40E−1	–	8.61E−2	–	2.41
20	6.61E−2	1.08	3.78E−2	1.19	7.52
40	1.31E−2	2.34	6.90E−3	2.45	71.85
80	2.00E−3	2.71	1.01E−3	2.77	1064.60
160	2.73E−4	2.87	1.37E−4	2.88	18 834.00
N	RK3, $\Delta t = 0.1\Delta x^2$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
10	1.90E−1	–	1.00E−1	–	1.62
20	2.89E−2	2.72	1.61E−2	2.63	44.45
40	2.30E−3	3.65	1.25E−3	3.69	1449.13
80	1.71E−4	3.75	9.30E−5	3.75	45681.23
160	1.60E−5	3.42	9.00E−6	3.37	1 456 538.89

drawn. In this example the Krylov SEIF methods generally spend less CPU time than the Krylov MEIF methods to reach similar level numerical errors. And both methods are much more efficient than Runge–Kutta methods.

Next we test the single-step integration factor methods for the three spatial dimension equation

$$\begin{cases} u_t + (\frac{1}{2}u^2)_x + (\frac{1}{2}u^2)_y + (\frac{1}{2}u^2)_z = u_{xx} + u_{yy} + u_{zz} + 3u + \cos(x+y+z+t)(1+3\sin(x+y+z+t)), \\ u(x,y,z,0) = \sin(x+y+z), \qquad 0 \le x, y, z \le 2\pi, \end{cases} \tag{54}$$

with periodic boundary conditions, and the exact solution is $u(x, y, z, t) = \sin(x+y+z+t)$. First explicit methods are tested. The computation results are reported in Table 5 and Table 6. We observe that the desired accuracy orders are obtained for all schemes. Again, in general the Krylov SEIF methods spend less CPU time than the Krylov MEIF methods to reach similar level numerical errors. However, in this case, we can also see that although Runge–Kutta methods needs much smaller time step sizes and more CPU times than Krylov integration factor methods on a same spatial mesh, their errors are smaller than

Table 7
Example 1, the three-dimensional equation (54). CPU time, error, and order of accuracy of Krylov SIIF2, Krylov MIIF2, and RK2 methods with the third order WENO spatial discretization for the advection term. Final time $T = 1.0$. N is the number of grid points in each spatial direction.

N	Krylov SIIF2, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
10	3.23E−1	–	1.59E−1	–	1.71
20	7.31E−2	2.14	4.44E−2	1.84	9.84
40	2.18E−2	1.75	1.28E−2	1.79	107.55
80	5.77E−3	1.92	3.35E−3	1.93	1218.37
160	1.47E−3	1.97	8.51E−4	1.98	18 215.77
N	Krylov MIIF2, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
10	5.96E−1	–	1.88E−1	–	2.74
20	2.63E−1	1.18	1.05E−1	0.84	8.44
40	1.26E−1	1.06	4.45E−2	1.24	59.88
80	5.31E−2	1.25	1.55E−2	1.52	551.77
160	2.03E−2	1.39	4.32E−3	1.84	8456.35
N	RK2, $\Delta t = 0.15\Delta x^2$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
10	8.59E−2	–	4.77E−2	–	0.95
20	9.00E−3	3.25	4.37E−3	3.45	21.19
40	5.76E−3	0.64	2.58E−3	0.76	646.58
80	1.70E−3	1.76	7.71E−4	1.74	20 478.39
160	4.46E−4	1.93	2.04E−4	1.92	672 881.48

Table 8
Example 1, the three-dimensional equation (54). CPU time, error, and order of accuracy of Krylov SIIF3, Krylov MIIF3, and RK3 methods with the third order WENO spatial discretization for the advection term. Final time $T = 1.0$. N is the number of grid points in each spatial direction.

N	Krylov SIIF3, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
10	1.78E−1	–	1.04E−1	–	2.28
20	2.67E−2	2.74	1.54E−2	2.76	11.27
40	1.96E−3	3.77	1.06E−3	3.86	215.80
80	1.38E−4	3.83	6.65E−5	3.99	3119.01
160	1.38E−5	3.32	6.24E−6	3.41	41 095.87
N	Krylov MIIF3, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
10	3.42E−1	–	1.68E−1	–	2.25
20	9.52E−2	1.85	4.48E−2	1.91	8.32
40	3.57E−2	1.42	1.57E−2	1.51	82.66
80	7.14E−3	2.32	3.21E−3	2.29	1279.88
160	1.19E−3	2.59	5.30E−4	2.60	28 864.92
N	RK3, $\Delta t = 0.1\Delta x^2$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
10	1.90E−1	–	1.00E−1	–	1.62
20	2.89E−2	2.72	1.61E−2	2.63	44.45
40	2.30E−3	3.65	1.25E−3	3.69	1449.13
80	1.71E−4	3.75	9.30E−5	3.75	45 681.23
160	1.60E−5	3.42	9.00E−6	3.37	1 456 538.89

Krylov integration factor methods. This is due to the computational costs of matrix–vector operations in Krylov integration factor methods, and also relatively large errors in integration factor type schemes [7]. Implicit schemes are tested and results are reported in Table 7 and Table 8. We obtain similar observations as the explicit methods except that the third order Krylov single-step IIF method shows very accurate results. The Krylov multistep IIF methods show larger numerical errors in this example than that in [19]. This is because the time step size we use here is much larger than that used in [19], for the purpose of comparing multistep and single-step schemes' performance under a same time step size.

This example shows that the Krylov approach can efficiently deal with the large matrix exponential challenge in single-step IIF schemes for high spatial dimension problems, similar as what we observe in multistep schemes [5,19].

Example 2 *(Stiff system I).* We consider an advection–diffusion–reaction system with stiff reaction terms on two-dimensional domain $\Omega = (0, 2\pi)^2$. The system was used to test different IIF schemes in [29,30,37,19]. It has the following form

Table 9
Example 2. CPU time, error, and order of accuracy of Krylov MIIF2, Krylov SIIF2, and RK2 methods with the third order WENO spatial discretization for the advection term. Final time $T = 1.0$. N is the number of grid points in each spatial direction.

N	Krylov MIIF2, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	3.39E−1	–	9.79E−2	–	1.62
40	7.23E−2	2.23	2.28E−2	2.10	4.50
80	1.54E−2	2.23	5.40E−3	2.08	18.24
160	4.00E−3	1.93	1.40E−3	1.95	136.94
N	Krylov SIIF2, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	2.49E−1	–	7.94E−2	–	3.09
40	6.10E−2	2.03	1.19E−2	2.74	10.61
80	1.01E−2	2.59	2.10E−3	2.50	68.41
160	1.50E−3	2.75	3.97E−4	2.40	485.26
N	RK2, $\Delta t = 0.05\Delta x^2$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	1.40E−1	–	5.58E−2	–	10.57
40	2.08E−2	2.75	9.63E−3	2.53	157.35
80	4.74E−3	2.13	2.81E−3	1.78	2623.07
160	1.27E−3	1.90	9.08E−4	1.63	39 789.26

Table 10
Example 2. CPU time, error, and order of accuracy of Krylov MIIF3, Krylov SIIF3, and RK3 methods with the third order WENO spatial discretization for the advection term. Final time $T = 1.0$. N is the number of grid points in each spatial direction.

N	Krylov MIIF3, $\Delta t = 0.6\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	1.45E−1	–	2.72E−2	–	2.24
40	3.03E−2	2.21	6.65E−3	2.03	5.87
80	4.25E−3	2.83	6.33E−4	3.39	24.72
160	4.08E−4	3.38	7.52E−5	3.07	189.60
N	Krylov SIIF3, $\Delta t = 0.3\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	2.28E−1	–	6.61E−2	–	4.96
40	4.25E−2	2.42	7.90E−3	3.06	17.98
80	5.40E−3	2.98	9.68E−4	3.03	112.38
160	5.26E−4	3.36	9.97E−5	3.28	760.99
N	RK3, $\Delta t = 0.05\Delta x^2$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	2.30E−1	–	1.00E−1	–	15.12
40	4.32E−2	2.41	1.59E−2	2.65	235.26
80	5.21E−3	3.05	1.94E−3	3.03	3820.41
160	4.89E−4	3.41	1.96E−4	3.31	59 435.53

$$\begin{cases} u_t + (a/2)(u_x + u_y) = (d/2)(u_{xx} + u_{yy}) - bu + v, \\ v_t + (a/2)(v_x + v_y) = (d/2)(v_{xx} + v_{yy}) - cv, \end{cases} \tag{55}$$

with periodic boundary conditions. For the initial condition

$$u|_{t=0} = 2\cos(x + y), \qquad v|_{t=0} = (b - c)\cos(x + y),$$

the system has the following exact solution

$$\begin{cases} u(x, y, t) = (e^{-(b+d)t} + e^{-(c+d)t})\cos(x + y - at), \\ v(x, y, t) = (b - c)e^{-(c+d)t}\cos(x + y - at). \end{cases} \tag{56}$$

The parameters are chosen as $a = c = d = 1$ and $b = 100$ to give stiff reaction terms. The final time $T = 1.0$. We compare the performance of Krylov SIIF, Krylov MIIF and Runge–Kutta methods with the third order WENO spatial discretization for the advection terms in Table 9 and Table 10. From the computation results, we can see that the designed second or third order accuracy is obtained for second order or third order methods. Both Krylov MIIF and Krylov SIIF methods can have large time step sizes for this stiff problem, due to their implicit treatment for the stiff reaction terms. They take the time step size $\Delta t = O(\Delta x)$ as that for a pure hyperbolic PDE. For the regular Runge–Kutta methods, $\Delta t = O(\Delta x)^2$ is needed for the stability of the computations. And due to the stiff reaction terms, RK methods need very small time step sizes such as

Table 11
Example 2. CPU time, error, and order of accuracy of Krylov MIIF2, Krylov SIIF2, Krylov MIIF3, Krylov SIIF3, and the 5th order Radau IIA method for this problem with $a = 0$, $c = 1$, $b = 100$, $d = 1$ (reaction dominated). Final time $T = 1.0$. N is the number of grid points in each spatial direction.

N	Krylov MIIF2, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	7.08E−2	–	4.47E−2	–	0.08
40	1.92E−2	1.88	1.22E−2	1.87	0.33
80	5.01E−3	1.94	3.23E−3	1.92	3.38
160	1.31E−3	1.94	8.07E−4	2.00	16.5
N	Krylov SIIF2, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	8.38E−2	–	3.84E−2	–	0.39
40	2.08E−2	2.01	6.66E−3	2.53	1.14
80	5.19E−3	2.00	1.67E−3	2.00	10.03
160	1.29E−3	2.01	4.15E−4	2.01	62.77
N	Krylov MIIF3, $\Delta t = 0.6\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	1.38E−2	–	8.71E−3	–	0.11
40	2.35E−3	2.55	1.49E−3	2.55	0.31
80	3.79E−4	2.63	2.41E−4	2.63	2.34
160	5.36E−5	2.82	3.41E−5	2.82	13.44
N	Krylov SIIF3, $\Delta t = 0.3\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	9.50E−3	–	3.48E−3	–	1.94
40	1.00E−4	6.57	3.22E−5	6.76	8.55
80	6.88E−6	3.86	2.21E−6	3.86	65.20
160	5.06E−7	3.77	1.63E−7	3.76	461.69
N	5th order Radau IIA (Gauss–Seidel linear solver), $\Delta t = 0.25\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	1.44E−3	–	1.00E−3	–	12.94
40	9.04E−5	3.99	6.04E−5	4.05	130.92
80	5.66E−6	4.00	3.69E−6	4.03	2149.34
160	3.54E−7	4.00	2.28E−7	4.02	34603.50

$\Delta t = 0.05(\Delta x)^2$. From the results, we can observe that much less CPU time is needed by using Krylov SIIF or Krylov MIIF methods than RK methods to reach a similar level numerical error.

A nice property of IIF methods is that solving large coupled systems is avoided. This is different from a fully implicit scheme. For stiff nonlinear PDE problems (or the spatial discretization scheme is nonlinear such as WENO schemes), fully implicit methods have very large stability regions but typically require the solution of large nonlinear coupled system of equations, so advanced, efficient and robust methods for solving large nonlinear algebraic systems need to be used. To study the efficiency of IIF methods, we implement the fifth order Runge–Kutta method based 'Radau IIA' method [14,2,9]. Here we focus on testing the stiff part by taking $a = 0$ in the system. Hence the system has no advection part and the nonlinear WENO scheme is not applied. When the fully implicit Radau IIA time discretization is applied, we obtain coupled $N^2 \times N^2$ linear systems at every time step. A fourth order central difference approximation is applied for discretizing the diffusion terms with the fifth order Radau IIA time discretization. The Gauss–Seidel iterative method is used to solve the coupled sparse $N^2 \times N^2$ linear systems. We report the numerical results of Krylov MIIF2, Krylov SIIF2, Krylov MIIF3, Krylov SIIF3 and the 5th order Radau IIA in Table 11 and Table 12. Table 11 is for the reaction dominated case ($c = 1, b = 100, d = 1$), and Table 12 is for the diffusion dominated case ($c = 0.01, b = 0.1, d = 1$). We can see that IIF methods can achieve smaller numerical errors in the diffusion dominated case than that in the reaction dominated case. Both IIF methods and the 5th order Radau IIA achieve large time step sizes $\Delta t = O(\Delta x)$. The fourth order of convergence order reported in the tables for the Radau IIA method is due to the fourth order central difference approximation to the diffusion terms. We compare the numerical errors of third order IIF methods (Krylov MIIF3 and Krylov SIIF3) and the Radau IIA scheme since both the third order IIF methods and the 5th order Radau IIA use the same spatial discretization schemes (the fourth order central difference). For the reaction dominated case, we can see that on the same mesh, the Radau IIA scheme is more accurate than the Krylov MIIF3 scheme, and it has comparable numerical errors as the Krylov SIIF3 scheme. However for the diffusion dominated case, the Krylov MIIF3 and Krylov SIIF3 schemes have smaller errors than the Radau IIA. This shows that different numerical methods have their own advantages for different problems. In terms of CPU times, different from IIF methods, fully implicit schemes such as Radau IIA depend on the methods used for solving the large linear systems at every time step. Here the CPU times for Radau IIA are due to the Gauss–Seidel iterative linear solver. Certainly more advanced methods for solving large sparse linear systems can significantly accelerate the computations of fully implicit schemes.

Table 12
Example 2. CPU time, error, and order of accuracy of Krylov MIIF2, Krylov SIIF2, Krylov MIIF3, Krylov SIIF3, and the 5th order Radau IIA method for this problem with $a = 0$, $c = 0.01$, $b = 0.1$, $d = 1$ (diffusion dominated). Final time $T = 1.0$. N is the number of grid points in each spatial direction.

N	Krylov MIIF2, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	2.27E−4	–	1.44E−4	–	0.14
40	6.21E−5	1.87	3.95E−5	1.87	0.36
80	1.62E−5	1.94	1.03E−5	1.94	2.83
160	4.13E−6	1.97	2.63E−6	1.97	16.06
N	Krylov SIIF2, $\Delta t = 0.5\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	5.74E−3	–	1.90E−3	–	0.34
40	1.43E−3	2.01	4.77E−4	1.99	1.39
80	3.58E−4	2.00	1.19E−4	2.00	9.75
160	8.96E−5	2.00	2.98E−5	2.00	63.02
N	Krylov MIIF3, $\Delta t = 0.6\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	1.85E−5	–	1.17E−5	–	0.16
40	2.94E−6	2.65	1.87E−6	2.65	0.25
80	2.40E−7	3.61	1.53E−7	3.61	2.28
160	1.68E−8	3.84	1.07E−8	3.84	13.45
N	Krylov SIIF3, $\Delta t = 0.3\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	7.48E−5	–	2.47E−5	–	2.58
40	4.71E−6	3.99	1.56E−6	3.98	12.05
80	2.95E−7	4.00	9.81E−8	3.99	97.38
160	1.84E−8	4.00	6.14E−9	4.00	691.97
N	5th order Radau IIA (Gauss–Seidel linear solver), $\Delta t = 0.25\Delta x$				
	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
20	7.48E−5	–	5.21E−5	–	10.50
40	4.71E−6	3.99	3.14E−6	4.05	112.53
80	2.95E−7	4.00	1.92E−7	4.03	1596.52
160	1.85E−8	4.00	1.19E−8	4.01	20 538.91

Example 3 *(Stiff system II – Schnakenberg model).* The Schnakenberg system [31] has been used to model the spatial distribution of a morphogen, e.g., the distribution of calcium in the hairs of the whorl in *Acetabularia* [11]. It is also a classical example for the testing of numerical methods for solving reaction–diffusion models in mathematical biology. The Schnakenberg system with an advection term has the form

$$\begin{cases} \dfrac{\partial C_a}{\partial t} + \dfrac{\partial C_a}{\partial x} + \dfrac{\partial C_a}{\partial y} = D_1 \nabla^2 C_a + \kappa (a - C_a + C_a^2 C_i), \\ \dfrac{\partial C_i}{\partial t} + \dfrac{\partial C_i}{\partial x} + \dfrac{\partial C_i}{\partial y} = D_2 \nabla^2 C_i + \kappa (b - C_a^2 C_i), \end{cases} \tag{57}$$

where C_a and C_i denote concentrations of activator and inhibitor respectively, D_1 and D_2 are diffusion coefficients, κ, a and b are rate constants of the biochemical reactions. We take the initial conditions as

$$\begin{cases} C_a(x, y, 0) = a + b + 10^{-3} e^{-100((x-\frac{1}{3})^2 + (y-\frac{1}{2})^2)}, \\ C_i(x, y, 0) = \dfrac{b}{(a+b)^2}, \end{cases} \tag{58}$$

and the boundary conditions are taken as periodic boundary conditions. The parameter values are $\kappa = 100, a = 0.1305, b = 0.7695, D_1 = 0.05, D_2 = 1$. This is a stiff Turing system [39]. We simulate the system on the unit square domain $\Omega = (0, 1)^2$ using both the new Krylov single-step and Krylov multistep implicit integration factor WENO methods. Following the discussions in [19] for this problem, we use a smaller Krylov subspace dimension $M = 10$ and the following results show that correct convergence orders can be obtained. To study the performance and convergence of the methods for this system, we list in Table 13 the CPU time, error, and order of accuracy for simulations of the Schnakenberg model, on a fixed spatial resolution of 32×32 mesh. Since there is no explicit form for the exact solution of this problem, the errors listed here are convergence errors. Namely, the error at Δt is measured as a difference between this solution, $C_{a,\Delta t}$, and the solution $C_{a,2\Delta t}$ for time step size $2\Delta t$ at time $T = 1.0$, i.e.,

$$E_{\Delta t} = \| C_{a,\Delta t} - C_{a,2\Delta t} \|.$$

Table 13
Example 3, Schnakenberg model. CPU time, error, and order of accuracy of Krylov single-step and multistep IIF methods with the third order WENO spatial discretization for the advection term. The Krylov subspace dimension $M = 10$. $T = 1.0$.

Krylov SIIF2					
Δt	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
5.00E−4	1.79E−0	–	2.17E−1	–	1784.49
2.50E−4	7.20E−1	1.31	7.31E−2	1.57	3649.67
1.25E−4	2.37E−1	1.60	1.95E−2	1.91	6062.35
6.25E−5	6.26E−2	1.92	5.00E−3	1.96	14 612.74
Krylov MIIF2					
Δt	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
5.00E−4	4.00E−1	–	4.25E−2	–	1302.64
2.50E−4	1.20E−1	1.78	1.34E−2	1.67	2693.62
1.25E−4	3.07E−2	1.93	3.70E−3	1.87	5843.02
6.25E−5	7.80E−3	1.97	9.66E−4	1.94	12 616.27
Krylov SIIF3					
Δt	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
5.00E−4	8.59E−2	–	1.10E−2	–	3975.50
2.50E−4	1.31E−2	2.71	1.60E−3	2.78	8199.55
1.25E−4	1.80E−3	2.86	2.13E−4	2.91	16 238.12
6.25E−5	2.34E−4	2.94	2.72E−5	2.97	27 377.87
Krylov MIIF3					
Δt	L^∞ error	L^∞ order	L^1 error	L^1 order	CPU time (seconds)
5.00E−4	2.40E−1	–	2.68E−2	–	1489.30
2.50E−4	4.95E−2	2.31	5.50E−3	2.28	2993.84
1.25E−4	7.80E−3	2.67	8.45E−4	2.70	6420.82
6.25E−5	1.10E−3	2.87	1.13E−4	2.90	12 680.21

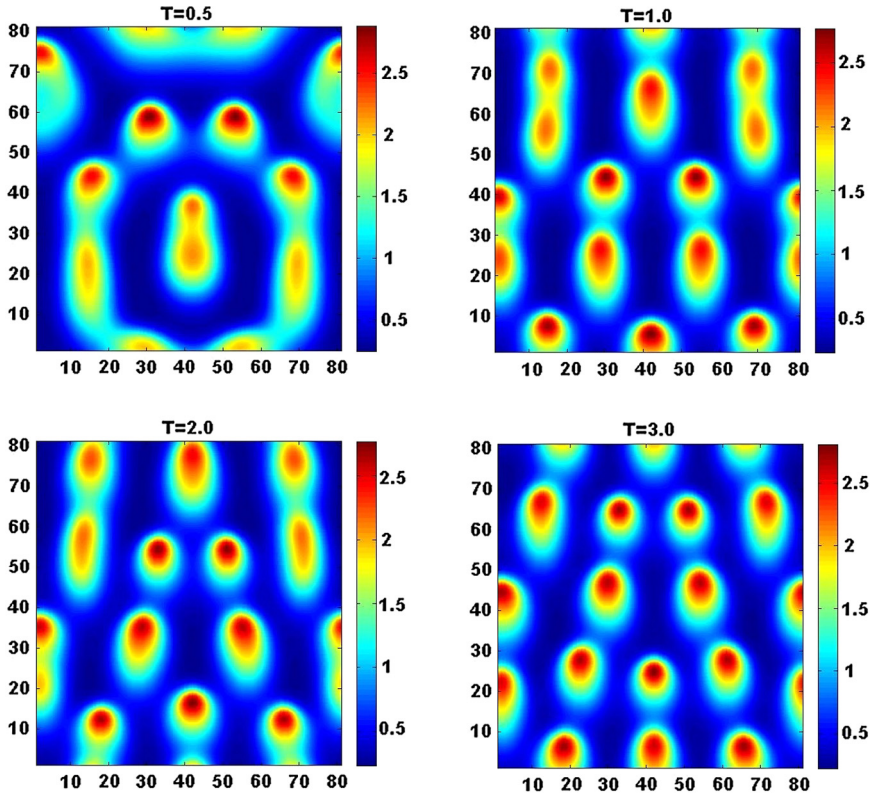

Fig. 9. Numerical solution of the Schnakenberg system by the Krylov SIIF2 WENO method on a 80×80 mesh. Contour plots of time evolution of the concentration of the activator C_a. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

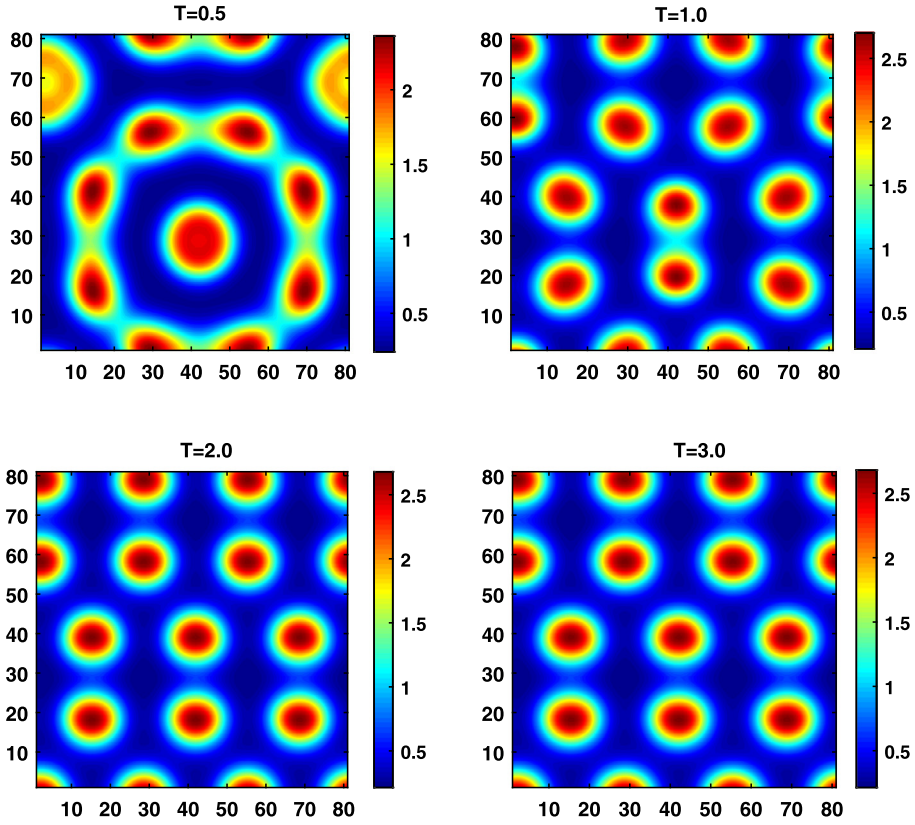

Fig. 10. Numerical solution of the Schnakenberg system by the Krylov SIIF3 WENO method on a 80×80 mesh. Contour plots of time evolution of the concentration of the activator C_a. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

The new Krylov SIIF2/SIIF3 scheme clearly shows a second order/third order of accuracy in time as expected. Krylov multistep methods (MIIF2 and MIIF3) also show correct convergence orders. It is interesting to see that for this example, the MIIF2 scheme converges faster than the SIIF2 scheme while the SIIF3 scheme converges faster than the MIIF3 scheme. The time evolutions of the concentration of activator C_a by single-step methods are shown in Fig. 9 and Fig. 10. Comparing two figures, we notice that with the same N and Δt, the third order scheme (SIIF3) has a better resolution than the second order one (SIIF2). We can also observe that the initial perturbation in (58) is amplified and spreads, leading to a formation of spot-like patterns.

Example 4 *(Nonlinear viscous Burgers' equation).* We consider the two-dimensional nonlinear viscous Burgers' equation

$$\begin{cases} u_t + (\frac{u^2}{2})_x + (\frac{u^2}{2})_y = d\Delta u, \qquad -2 \le x \le 2, \ -2 \le y \le 2, \\ u(x, y, 0) = 0.3 + 0.7 \sin(\frac{\pi}{2}(x + y)), \end{cases} \tag{59}$$

with periodic boundary condition. d is the viscous coefficient. The Krylov SEIF2 WENO scheme is used to solve the PDE to $T = 5/\pi^2$. In this example, we test the performance of the scheme with our Krylov single-step integration factor time discretization technique for convection–diffusion equations without/with the convection dominated property, by taking $d = 1.0$ and $d = 0.01$. The simulation results are reported in Fig. 11. We can see that while the solution is very smooth for $d = 1.0$ (the left pictures), a sharp gradient is developed for the convection dominated case $d = 0.01$ (the right pictures). We can observe that the non-oscillatory property of the WENO scheme is preserved well for the convection dominated problem, under this Krylov single-step integration factor time discretization technique.

5. Conclusions

In [19], multistep Krylov implicit integration factor methods were developed for efficiently solving stiff advection–diffusion–reaction systems. In this paper, we extended the previous work and developed Krylov single-step integration factor WENO methods. For the concern of stability, the new single-step IIF methods are carefully designed to avoid positive

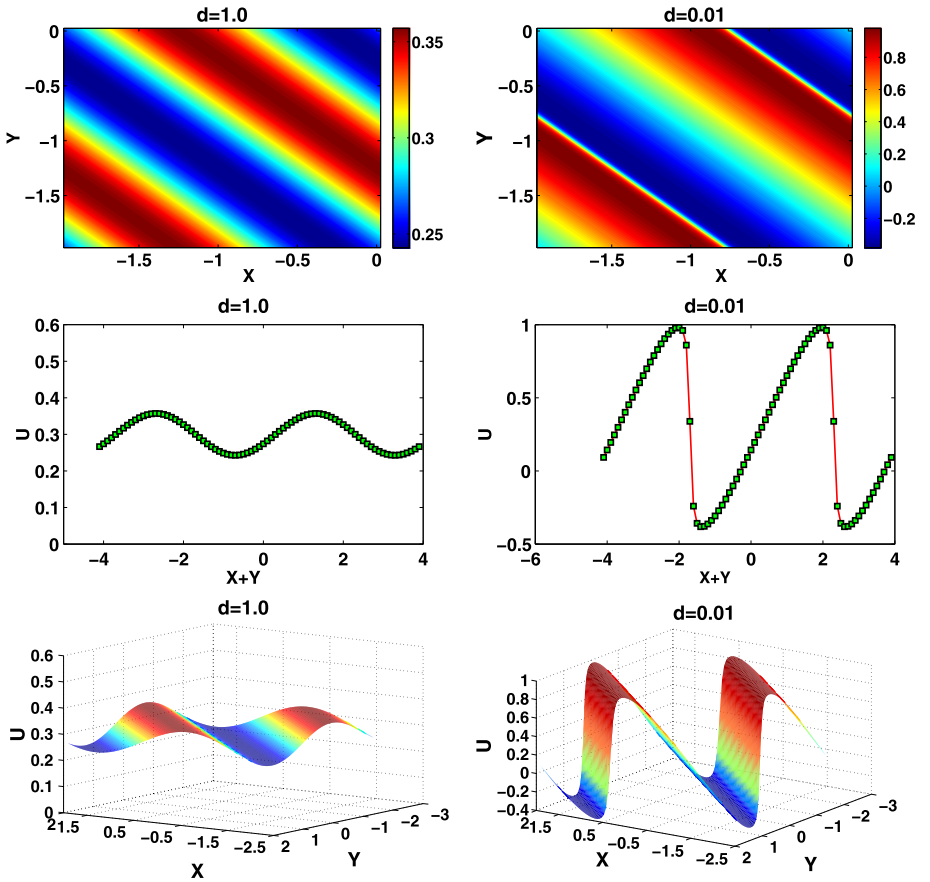

Fig. 11. Example 4, nonlinear viscous Burgers' equation. Simulation on a 80×80 mesh by the Krylov SEIF2 WENO scheme. Time $T = 5/\pi^2$. Left pictures: the viscous coefficient $d = 1.0$; right pictures: the viscous coefficient $d = 0.01$. Top: contour plots; middle: 1D cutting-plot along $x = y$; bottom: 3D surface plots of the solutions. (For interpretation of the colors in this figure, the reader is referred to the web version of this article.)

exponentials. They also preserve the local implicit property of the original multistep IIF schemes. Via numerical experiments and linear stability analysis, we verified that similar to the multistep schemes, large time step size computations are achieved, and the new schemes have large stability regions and are efficient and accurate for simulating nonlinear advection–diffusion–reaction systems. The second and third order single-step IIF schemes are developed for ADR systems in this paper. It will be interesting to consider extending the methodology here to construct and test higher order single-step IIF schemes, which is one of our next projects. Another interesting future work is to theoretically analyze the interaction of different errors in the method, e.g., the Krylov subspace approximation errors in the matrix exponentials and the truncation errors of the schemes.

References

[1] U. Ascher, S. Ruuth, B. Wetton, Implicit–explicit methods for time-dependent PDE's, SIAM J. Numer. Anal. 32 (1995) 797–823.
[2] O. Axelsson, A class of A-stable methods, BIT 9 (1969) 185–199.
[3] G. Beylkin, J.M. Keiser, L. Vozovoi, A new class of time discretization schemes for the solution of nonlinear PDEs, J. Comput. Phys. 147 (1998) 362–387.
[4] A. Bourlioux, A.T. Layton, M.L. Minion, High-order multi-implicit spectral deferred correction methods for problems of reactive flow, J. Comput. Phys. 189 (2003) 651–675.
[5] S. Chen, Y.-T. Zhang, Krylov implicit integration factor methods for spatial discretization on high dimensional unstructured meshes: application to discontinuous Galerkin methods, J. Comput. Phys. 230 (2011) 4336–4352.
[6] A. Christlieb, B. Ong, J.-M. Qiu, Integral deferred correction methods constructed with high order Runge–Kutta integrators, Math. Comput. 79 (2010) 761–783.
[7] S.M. Cox, P.C. Matthews, Exponential time differencing for stiff systems, J. Comput. Phys. 176 (2002) 430–455.
[8] A. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations, BIT 40 (2000) 241–266.
[9] B.L. Ehle, On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems, Report CSRR, Dept. AACS, Univ. of Waterloo, Ontario, Canada, 2010. See also: SIAM J. Math. Anal. 4 (2010) 671–680.
[10] E. Gallopoulos, Y. Saad, Efficient solution of parabolic equations by Krylov approximation methods, SIAM J. Sci. Stat. Comput. 13 (1992) 1236–1264.
[11] B.C. Goodwin, L.E.H. Trainor, Tip and whorl morphogenesis in Acetabularia by calcium-regulated strain fields, J. Theor. Biol. 117 (1985) 79–106.
[12] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge–Kutta schemes, Math. Comput. 67 (1998) 73–85.
[13] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability preserving high order time discretization methods, SIAM Rev. 43 (2001) 89–112.

[14] E. Hairer, G. Wanner, Stiff differential equations solved by Radau methods, J. Comput. Appl. Math. 111 (1999) 93–111.
[15] N.J. Higham, The scaling and squaring method for the matrix exponential revisited, SIAM Rev. 51 (2009) 747–764.
[16] M. Hochbruck, C. Lubich, On Krylov subspace approximations to the matrix exponential operator, SIAM J. Numer. Anal. 34 (1997) 1911–1925.
[17] J. Huang, J. Jia, M. Minion, Arbitrary order Krylov deferred correction methods for differential algebraic equations, J. Comput. Phys. 221 (2007) 739–760.
[18] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[19] T. Jiang, Y.-T. Zhang, Krylov implicit integration factor WENO methods for semilinear and fully nonlinear advection–diffusion–reaction equations, J. Comput. Phys. 253 (2013) 368–388.
[20] C.-Y. Jung, T.B. Nguyen, Semi-analytical time differencing methods for stiff problems, J. Sci. Comput. 63 (2015) 355–373.
[21] A. Kanevsky, M.H. Carpenter, D. Gottlieb, J.S. Hesthaven, Application of implicit–explicit high order Runge–Kutta methods to Discontinuous-Galerkin schemes, J. Comput. Phys. 225 (2007) 1753–1781.
[22] A.-K. Kassam, L.N. Trefethen, Fourth-order time stepping for stiff PDEs, SIAM J. Sci. Comput. 26 (2005) 1214–1233.
[23] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, Appl. Numer. Math. 44 (2003) 139–181.
[24] A.T. Layton, M.L. Minion, Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics, J. Comput. Phys. 194 (2004) 697–715.
[25] X.F. Liu, Q. Nie, Compact integration factor methods for complex domains and adaptive mesh refinement, J. Comput. Phys. 229 (2010) 5692–5706.
[26] Y. Maday, A.T. Patera, E.M. Ronquist, An operator-integration-factor splitting method for time-dependent problems: application to incompressible fluid flow, J. Sci. Comput. 5 (1990) 263–292.
[27] M.L. Minion, Semi-implicit spectral deferred correction methods for ordinary differential equations, Commun. Math. Sci. 1 (2003) 471–500.
[28] C. Moler, C. Van Loan, Nineteen dubious ways to compute the exponential of a matrix twenty-five years later, SIAM Rev. 45 (2003) 3–49.
[29] Q. Nie, Y.-T. Zhang, R. Zhao, Efficient semi-implicit schemes for stiff systems, J. Comput. Phys. 214 (2006) 521–537.
[30] Q. Nie, F. Wan, Y.-T. Zhang, X.-F. Liu, Compact integration factor methods in high spatial dimensions, J. Comput. Phys. 227 (2008) 5238–5255.
[31] J. Schnakenberg, Simple chemical reaction systems with limit cycle behavior, J. Theor. Biol. 81 (1979) 389–400.
[32] C.-W. Shu, TVD time discretizations, SIAM J. Sci. Stat. Comput. 9 (1988) 1073–1084.
[33] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[34] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor, A. Quarteroni (Eds.), Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, in: Lect. Notes Math., vol. 1697, Springer, 1998.
[35] L.N. Trefethen, D. Bau, Numerical Linear Algebra, SIAM, 1997.
[36] J.G. Verwer, B.P. Sommeijer, W. Hundsdorfer, RKC time-stepping for advection–diffusion–reaction problems, J. Comput. Phys. 201 (2004) 61–79.
[37] S. Zhao, J. Ovadia, X. Liu, Y.-T. Zhang, Q. Nie, Operator splitting implicit integration factor methods for stiff reaction–diffusion–advection systems, J. Comput. Phys. 230 (2011) 5996–6009.
[38] X. Zhong, Additive semi-implicit Runge–Kutta methods for computing high-speed nonequilibrium reactive flows, J. Comput. Phys. 128 (1996) 19–31.
[39] J. Zhu, Y.-T. Zhang, S.A. Newman, M. Alber, Application of Discontinuous Galerkin Methods for reaction–diffusion systems in developmental biology, J. Sci. Comput. 40 (2009) 391–418.