

# Krylov implicit integration factor WENO methods for semilinear and fully nonlinear advection–diffusion–reaction equations <sup>☆</sup>



Tian Jiang, Yong-Tao Zhang <sup>\*</sup>

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

## ARTICLE INFO

### Article history:

Received 15 October 2012

Received in revised form 25 June 2013

Accepted 15 July 2013

Available online 23 July 2013

### Keywords:

Advection–diffusion–reaction equations

Implicit integration factor methods

Krylov subspace approximation

High order accuracy

Weighted essentially non-oscillatory schemes

## ABSTRACT

Implicit integration factor (IIF) methods are originally a class of efficient “exactly linear part” time discretization methods for solving time-dependent partial differential equations (PDEs) with linear high order terms and stiff lower order nonlinear terms. For complex systems (e.g. advection–diffusion–reaction (ADR) systems), the highest order derivative term can be nonlinear, and nonlinear nonstiff terms and nonlinear stiff terms are often mixed together. High order weighted essentially non-oscillatory (WENO) methods are often used to discretize the hyperbolic part in ADR systems. There are two open problems on IIF methods for solving ADR systems: (1) how to obtain higher than the second order global time discretization accuracy; (2) how to design IIF methods for solving fully nonlinear PDEs, i.e., the highest order terms are nonlinear. In this paper, we solve these two problems by developing new Krylov IIF-WENO methods to deal with both semilinear and fully 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 nonstiff hyperbolic part of the system. Large time step size computations are obtained. We analyze the stability and truncation errors of the schemes. Numerical examples of both scalar equations and systems in two and three spatial dimensions are shown to demonstrate the accuracy, efficiency and robustness of the methods.

© 2013 Elsevier Inc. All rights reserved.

## 1. Introduction

Efficient and high order temporal numerical schemes are important for the performance of high accuracy numerical simulations. A lot of state-of-the-art high order time-stepping methods were developed. Here we just give a few examples and it is not a complete list. For example, the total variation diminishing (TVD) Runge–Kutta (RK) schemes [38,39,14,15]; spectral deferred correction (SDC) methods [4,11,18,25,30]; high order implicit-explicit (IMEX) multistep/RK methods [2,21,24,43,45]; hybrid methods of SDC and high order RK schemes [9]; 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

<sup>☆</sup> Research was partially supported by NSF grant DMS-0810413.

<sup>\*</sup> Corresponding author. Tel.: +1 574 631 6079; fax: +1 574 631 4822.

E-mail addresses: [tjiang@nd.edu](mailto:tjiang@nd.edu) (T. Jiang), [yztang10@nd.edu](mailto:yztang10@nd.edu) (Y.-T. Zhang).

integration factor type time discretization can remove both the stability constrain and time direction numerical errors from the high order derivatives [3,10,28,22].

In [33], 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 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 [33] 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 [34] on rectangular meshes, and Krylov IIF methods [7] on general unstructured meshes for complex domains. The compact IIF methods were extended to curvilinear coordinates, such as polar and spherical coordinates in [26].

Nonlinear advection–diffusion–reaction (ADR) systems of equations [19] are common mathematical models in applications from biology, chemistry and physics. The fully nonlinear system with three spatial dimensions has the following general form

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

where  $\vec{u}$  is the unknown vector function,  $\vec{f}$ ,  $\vec{g}$  and  $\vec{h}$  are flux vector functions in three spatial dimensions respectively,  $\mathbf{D}(\vec{u})$  is the diffusion matrix and it could be nonlinear, and  $\vec{r}$  is the reaction term. Often the ADR models in applications have nonlinear advection and reaction terms, but a linear diffusion term  $\mathbf{D}\Delta\vec{u}$ , where  $\mathbf{D}$  is the diffusion constant matrix. In such case, the system is semilinear.

For many ADR systems in biology and chemistry, this time-dependent problem (1) could be a mixture of advection-dominated and diffusion-dominated cases. For example, the nonlinear advection terms can dominate at the early time in the system and later the diffusion term becomes dominated [27]. Hence a nonlinear stable discretization suitable for hyperbolic PDEs is needed for the advection terms, for example the weighted essentially non-oscillatory (WENO) schemes [40]. We apply the method of lines (MOL) approach to the system (1). For nonlinear advection terms  $\vec{f}(\vec{u})_x + \vec{g}(\vec{u})_y + \vec{h}(\vec{u})_z$ , the third order finite difference WENO scheme with Lax–Friedrichs flux splitting [20] is used. And the second or fourth order central finite difference scheme (depending on the order of accuracy of IIF time discretizations) is used to discretize the diffusion term. Then we obtain a semi-discretized ODE system

$$\frac{d\vec{U}}{dt} = \vec{F}_d(\vec{U}) + \vec{F}_a(\vec{U}) + \vec{R}(\vec{U}), \tag{2}$$

where  $\vec{U} = (\vec{u}_i)_{1 \leq i \leq N}$ ,  $\vec{F}_d(\vec{U}) = (\hat{F}_{di}(\vec{U}))_{1 \leq i \leq N}$ ,  $\vec{F}_a(\vec{U}) = (\hat{F}_{ai}(\vec{U}))_{1 \leq i \leq N}$ ,  $\vec{R} = (\vec{r}(\vec{u}_i))_{1 \leq i \leq N}$ .  $N$  is the total number of grid points,  $\vec{F}_d(\vec{U})$  is the approximation for the diffusion terms by the second or fourth order finite difference schemes, and its each component  $\hat{F}_{di}$  is a linear or nonlinear function of numerical values on the approximation stencil. If the diffusion term is linear,  $\vec{F}_d(\vec{U}) = C\vec{U}$  where  $C$  is the approximation matrix for the linear diffusion operator  $\mathbf{D}\Delta$  by the central finite difference scheme.  $\vec{F}_a(\vec{U})$  is the approximation for the nonlinear advection terms by the third order finite difference WENO scheme, and its each component  $\hat{F}_{ai}$  is a nonlinear function of several numerical values on the WENO approximation stencil [20].  $\vec{R}(\vec{U})$  is the nonlinear reaction term, and its each component  $\vec{r}(\vec{u}_i)$  is a nonlinear function which only depends on numerical values at one grid point.

In [44], an operator splitting IIF method was developed for semilinear ADR system. There are still two open problems on IIF methods for ADR systems. This first one is how to design high order IIF schemes (3rd order and above) for solving the semi-discretized system (2) derived from the ADR systems, and achieve a large time step size and efficient time evolution computation. There are following two problems needed to be solved in order to design a high order IIF scheme for ADR equations. Firstly, how to deal with the nonlinear term  $\vec{F}_a(\vec{U})$  resulting from the advection? This term comes from the WENO discretization and is usually highly nonlinear. Directly applying the original IIF scheme in [33] to this term will result in a large coupled nonlinear system which is difficult and expensive to solve. To solve this problem, we explicitly approximate the hyperbolic terms and treat it differently from the nonlinear reaction terms in this paper. Then this treatment raises the second problem. Since the time step sizes will change at every time step due to the CFL condition constraint for the hyperbolic term, the matrix exponentials need to be calculated at every time step. How to efficiently compute the matrix exponentials is important. In this paper, this problem is solved by using Krylov subspace approximations to the matrix exponentials as in [7] for solving the reaction–diffusion equations on unstructured meshes.

The second (also more interesting) open problem is how to design IIF methods for solving fully nonlinear PDEs, i.e., the highest order terms are nonlinear. So far, all IIF methods in the literature were designed for semilinear problems. However, fully nonlinear systems often arise in many mathematical models for biological and physical applications. For example, a biologically realistic fully nonlinear ADR system was developed in [1,27] to model chemotactic cell movements for generation of network patterns of vasculogenesis [6]. This model takes into account the finite size of cells in the biological systems and uses an exclusion volume principle that two cells cannot occupy the same space. As a result, the solutions do not blow up

in finite time. Hence it is a more biologically realistic model than the classical Keller–Segel model [23], whose solutions can blow up in finite time due to point-wise cell aggregation [16,17]. A property of the model is that the nonlinear diffusion term becomes dominate along with the time evolution of the system. Another class of examples is radiation diffusion equations in mathematical models for inertial confinement fusion or astrophysics [5,29]. Nonlinear diffusion terms dominate in these systems. Implicit schemes are usually used in numerical simulations of such systems in order to achieve large time step sizes. However traditional implicit schemes need to solve a big coupled nonlinear system at every time step, and often advanced preconditioning techniques have to be developed to improve the convergence of nonlinear algebraic solver and result in very efficient algorithms [35,32]. On the other hand, IIF methods provide another efficient approach suitable for solving diffusion dominated problems, as shown in [8] for the semilinear problems. Here we propose a novel approach for solving fully nonlinear system, i.e., the highest order term in the PDEs are nonlinear. The main idea is to factor out the linear part which mainly contributes to the stiffness of the nonlinear diffusion terms, then we can apply the integration factor approach to remove this stiffness.

The rest of the paper is organized as following. In Section 2, we derive and formulate the Krylov IIF-WENO methods for advection–diffusion–reaction systems. 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 IIF-WENO methods

In this section we describe the new methods in details. Although the novel part of the methods is in their temporal discretization, we first describe the spatial discretization for completeness.

### 2.1. Spatial discretization

In this paper, we use the third order finite difference WENO scheme with Lax–Friedrichs flux splitting [20] to discretize the nonlinear advection terms. 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

$$f(u)_x|_{x=x_i} \approx \frac{1}{\Delta x} (\hat{f}_{i+1/2} - \hat{f}_{i-1/2}), \tag{3}$$

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

$$\hat{f}_{i+1/2} = w_0 \left[ \frac{1}{2} f(u_i) + \frac{1}{2} f(u_{i+1}) \right] + w_1 \left[ -\frac{1}{2} f(u_{i-1}) + \frac{3}{2} f(u_i) \right], \tag{4}$$

where

$$w_r = \frac{\alpha_r}{\alpha_1 + \alpha_2}, \quad \alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2}, \quad r = 0, 1. \tag{5}$$

$d_0 = 2/3, d_1 = 1/3$  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”.  $\epsilon$  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”

$$f^+(u) = \frac{1}{2} (f(u) + \alpha u), \quad f^-(u) = \frac{1}{2} (f(u) - \alpha u), \tag{6}$$

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 [20,40] for more details.

For diffusion terms, central differences are used. For example, a second order approximation to a nonlinear diffusion term  $(k(u)u_x)_x$  at a grid point  $i$  has the form

$$\frac{\partial}{\partial x} \left( k(u) \frac{\partial u}{\partial x} \right)_i \approx \frac{(k(u) \frac{\partial u}{\partial x})_{i+\frac{1}{2}} - (k(u) \frac{\partial u}{\partial x})_{i-\frac{1}{2}}}{\Delta x} = \frac{k(\frac{u_{i+1}+u_i}{2})(u_{i+1} - u_i) - k(\frac{u_i+u_{i-1}}{2})(u_i - u_{i-1})}{(\Delta x)^2}, \tag{7}$$

and a fourth order approximation is

$$\begin{aligned} \frac{\partial}{\partial x} \left( k(u) \frac{\partial u}{\partial x} \right)_i &\approx \frac{-(k(u) \frac{\partial u}{\partial x})_{i+2} + (k(u) \frac{\partial u}{\partial x})_{i-2} + 8(k(u) \frac{\partial u}{\partial x})_{i+1} - 8(k(u) \frac{\partial u}{\partial x})_{i-1}}{12\Delta x} \\ &= \frac{-k(u_{i+2})(-u_{i+4} + u_i + 8u_{i+3} - 8u_{i+1}) + k(u_{i-2})(-u_i + u_{i-4} + 8u_{i-1} - 8u_{i-3})}{144(\Delta x)^2} \\ &\quad + \frac{8k(u_{i+1})(-u_{i+3} + u_{i-1} + 8u_{i+2} - 8u_i) - 8k(u_{i-1})(-u_{i+1} + u_{i-3} + 8u_i - 8u_{i-2})}{144(\Delta x)^2}. \end{aligned} \tag{8}$$

Similar approximations are applied to the y and z directions.

## 2.2. Krylov IIF temporal discretization

### 2.2.1. Linear or nonlinear diffusion term

We construct Krylov IIF methods for (2) by exactly integrating the linear part of the system. For the semilinear system, the linear part is  $\vec{F}_d(\vec{U}) = C\vec{U}$ . Then we can directly multiply (2) by the integration factor  $e^{-Ct}$  and integrate over one time step from  $t_n$  to  $t_{n+1} \equiv t_n + \Delta t_n$  to obtain

$$\vec{U}(t_{n+1}) = e^{C\Delta t_n} \vec{U}(t_n) + e^{C\Delta t_n} \int_0^{\Delta t_n} e^{-C\tau} \vec{F}_a(\vec{U}(t_n + \tau)) d\tau + e^{C\Delta t_n} \int_0^{\Delta t_n} e^{-C\tau} \vec{R}(\vec{U}(t_n + \tau)) d\tau. \tag{9}$$

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 (9) have different properties. The nonlinear reaction term  $\vec{R}(\vec{U})$  is usually stiff but local, while the nonlinear term  $\vec{F}_a(\vec{U})$  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. For the fully nonlinear system, we rewrite  $\vec{F}_d(\vec{U})$  at  $t = t_n$  as

$$\vec{F}_d(\vec{U}) = \vec{F}_d(\vec{U}_n) + C(\vec{U}_n)(\vec{U} - \vec{U}_n) + \vec{E}(\vec{U}), \tag{10}$$

where  $C(\vec{U}_n) = \frac{\partial \vec{F}_d}{\partial \vec{U}}(\vec{U}_n)$  is the Jacobian matrix,  $\vec{U}_n = \vec{U}(t_n)$ , and  $\vec{E}(\vec{U})$  is the remainder  $\vec{E}(\vec{U}) = \vec{F}_d(\vec{U}) - \vec{F}_d(\vec{U}_n) - C(\vec{U}_n)(\vec{U} - \vec{U}_n)$ . Denoting the Jacobian matrix  $C(\vec{U}_n)$  by  $C_n$  and substituting Eq. (10) into Eq. (2), we obtain

$$\frac{d\vec{U}}{dt} = C_n \vec{U} + \vec{F}_d(\vec{U}_n) - C_n \vec{U}_n + \vec{E}(\vec{U}) + \vec{F}_a(\vec{U}) + \vec{R}(\vec{U}). \tag{11}$$

Multiplying Eq. (11) by the integration factor  $e^{-C_n t}$ , we have

$$\frac{d}{dt} (e^{-C_n t} \vec{U}) = e^{-C_n t} (\vec{F}_d(\vec{U}) - C_n \vec{U} + \vec{F}_a(\vec{U}) + \vec{R}(\vec{U})), \tag{12}$$

and integrate it over one time step from  $t_n$  to  $t_{n+1} \equiv t_n + \Delta t_n$  to obtain

$$\vec{U}_{n+1} = e^{C_n \Delta t_n} \vec{U}_n + e^{C_n \Delta t_n} \int_0^{\Delta t_n} e^{-C_n \tau} (\vec{F}_d(\vec{U}(t_n + \tau)) - C_n \vec{U}(t_n + \tau) + \vec{F}_a(\vec{U}(t_n + \tau)) + \vec{R}(\vec{U}(t_n + \tau))) d\tau. \tag{13}$$

**Remark.** Eq. (13) is for a fully nonlinear system, while Eq. (9) is for a semilinear system. Their differences include that the linear part is changing at every time step for a fully nonlinear system, and there are some extra nonlinear terms in Eq. (13). However, the method in the following sections which deals with the matrix exponential and the nonlinear terms in the integrands of Eqs. (13) and (9) follows the same procedure. To simplify the notations, in the following sections we denote  $C_n$  in the fully nonlinear system (13) by  $C$  with the understanding that it is different at different time step;  $\vec{F}$  is used to represent  $\vec{F}_a$  in the semilinear system (9) and to represent  $\vec{F}_d - C_n \vec{U} + \vec{F}_a$  in the fully nonlinear system (13), unless otherwise indicated. Namely, we have

$$\vec{F} = \begin{cases} \vec{F}_a, & \text{for the semilinear system;} \\ \vec{F}_d - C_n \vec{U} + \vec{F}_a, & \text{for the fully nonlinear system.} \end{cases} \tag{14}$$

**Remark.** In order to perform the derivation (10) for the fully nonlinear system,  $\vec{F}_d(\vec{U})$  needs to be differentiable such that its Jacobian matrix exists.

2.2.2. Nonlinear reaction and advection terms

As in the original IIF methods [33], we approximate the nonlinear reaction term  $e^{-C\tau} \bar{R}(\bar{U}(t_n + \tau))$  by an  $(r - 1)$ -th order Lagrange polynomial  $p(\tau)$  with interpolation points at  $t_{n+1}, t_n, \dots, t_{n+2-r}$ . We would like to point out that time step sizes could be non-uniform, which is different from the case in the original IIF methods [33]. This is due to the fact that we have the nonlinear hyperbolic term in the PDE, and it will be treated explicitly (described in Section 2.2.3). Hence the time step sizes are non-uniform due to the CFL condition constraint for the hyperbolic (advection) term. Different time step sizes require the calculation of matrix exponentials at every time step. We use the Krylov subspace method to perform these calculations efficiently (described in Section 2.2.4).

Denote  $\tau_1 = \Delta t_n$ ,  $\tau_0 = 0$ ,  $\tau_i = -\sum_{k=i}^{-1} \Delta t_{n+k}$  for  $i = -1, -2, -3, \dots, 1 - r$ . The interpolation points are represented by  $t_{n+i} = t_n + \tau_i$ ,  $i = 1, 0, -1, \dots, 1 - r$ . Define  $\bar{U}_{n+i}$  as the numerical solution for  $\bar{U}(t_{n+i})$ . The first  $r$  points  $\{t_{n+i}, i = 1, 0, -1, \dots, 2 - r\}$  are used for an implicit approximation of the nonlinear reaction term:

$$e^{C\Delta t_n} \int_0^{\Delta t_n} e^{-C\tau} \bar{R}(\bar{U}(t_n + \tau)) d\tau \approx \sum_{i=2-r}^1 e^{C(\Delta t_n - \tau_i)} \bar{R}(\bar{U}_{n+i}) \int_0^{\Delta t_n} \prod_{j=2-r, j \neq i}^1 \frac{\tau - \tau_j}{\tau_i - \tau_j} d\tau. \tag{15}$$

The nonstiff advection term is highly nonlinear due to the WENO approximations. Different from the nonlinear reaction term, we approximate the nonlinear term  $e^{-C\tau} \bar{F}(\bar{U}(t_n + \tau))$  in (9) and (13) (see (14) for  $\bar{F}$ ) by an  $(r - 1)$ -th order Lagrange polynomial with interpolation points at  $t_n, t_{n-1}, \dots, t_{n+1-r}$ . Hence it is approximated explicitly:

$$e^{C\Delta t_n} \int_0^{\Delta t_n} e^{-C\tau} \bar{F}(\bar{U}(t_n + \tau)) d\tau \approx \sum_{i=1-r}^0 e^{C(\Delta t_n - \tau_i)} \bar{F}(\bar{U}_{n+i}) \int_0^{\Delta t_n} \prod_{j=1-r, j \neq i}^0 \frac{\tau - \tau_j}{\tau_i - \tau_j} d\tau. \tag{16}$$

2.2.3. IIF schemes for ADR systems

Combining Eqs. (9)–(16), we obtain the  $r$ -th order IIF scheme for ADR equations

$$\bar{U}_{n+1} = e^{C\Delta t_n} \bar{U}_n + \Delta t_n \left\{ \alpha_{n+1} \bar{R}(\bar{U}_{n+1}) + \sum_{i=2-r}^0 \alpha_{n+i} e^{C(\Delta t_n - \tau_i)} \bar{R}(\bar{U}_{n+i}) + \sum_{i=1-r}^0 \beta_{n+i} e^{C(\Delta t_n - \tau_i)} \bar{F}(\bar{U}_{n+i}) \right\}, \tag{17}$$

where the coefficients

$$\alpha_{n+i} = \frac{1}{\Delta t_n} \int_0^{\Delta t_n} \prod_{j=2-r, j \neq i}^1 \frac{\tau - \tau_j}{\tau_i - \tau_j} d\tau, \quad i = 1, 0, -1, \dots, 2 - r; \tag{18}$$

$$\beta_{n+i} = \frac{1}{\Delta t_n} \int_0^{\Delta t_n} \prod_{j=1-r, j \neq i}^0 \frac{\tau - \tau_j}{\tau_i - \tau_j} d\tau, \quad i = 0, -1, -2, \dots, 1 - r. \tag{19}$$

Specifically, the second order scheme (IIF2) is of the following form

$$\bar{U}_{n+1} = e^{C\Delta t_n} \bar{U}_n + \Delta t_n \{ \alpha_{n+1} \bar{R}(\bar{U}_{n+1}) + \alpha_n e^{C\Delta t_n} \bar{R}(\bar{U}_n) + \beta_{n-1} e^{C(\Delta t_n + \Delta t_{n-1})} \bar{F}(\bar{U}_{n-1}) + \beta_n e^{C\Delta t_n} \bar{F}(\bar{U}_n) \}, \tag{20}$$

where

$$\alpha_n = \frac{1}{2}, \quad \alpha_{n+1} = \frac{1}{2}, \quad \beta_{n-1} = -\frac{\Delta t_n}{2\Delta t_{n-1}}, \quad \beta_n = \frac{1}{\Delta t_{n-1}} \left( \frac{\Delta t_n}{2} + \Delta t_{n-1} \right).$$

And the third order scheme (IIF3) is

$$\bar{U}_{n+1} = e^{C\Delta t_n} \bar{U}_n + \Delta t_n \{ \alpha_{n+1} \bar{R}(\bar{U}_{n+1}) + \alpha_n e^{C\Delta t_n} \bar{R}(\bar{U}_n) + \alpha_{n-1} e^{C(\Delta t_n + \Delta t_{n-1})} \bar{R}(\bar{U}_{n-1}) + \beta_{n-2} e^{C(\Delta t_n + \Delta t_{n-1} + \Delta t_{n-2})} \bar{F}(\bar{U}_{n-2}) + \beta_{n-1} e^{C(\Delta t_n + \Delta t_{n-1})} \bar{F}(\bar{U}_{n-1}) + \beta_n e^{C\Delta t_n} \bar{F}(\bar{U}_n) \}, \tag{21}$$

where

$$\alpha_{n+1} = \frac{1}{(\Delta t_n + \Delta t_{n-1})} \left( \frac{\Delta t_n}{3} + \frac{\Delta t_{n-1}}{2} \right),$$

$$\alpha_n = \frac{1}{\Delta t_{n-1}} \left( \frac{\Delta t_n}{6} + \frac{\Delta t_{n-1}}{2} \right),$$

$$\alpha_{n-1} = -\frac{\Delta t_n^2}{6\Delta t_{n-1}(\Delta t_{n-1} + \Delta t_n)},$$

$$\begin{aligned} \beta_n &= 1 + \frac{1}{\Delta t_{n-1}(\Delta t_{n-1} + \Delta t_{n-2})} \left[ \frac{\Delta t_n^2}{3} + \frac{\Delta t_n}{2} (2\Delta t_{n-1} + \Delta t_{n-2}) \right], \\ \beta_{n-1} &= -\frac{1}{\Delta t_{n-1}\Delta t_{n-2}} \left[ \frac{\Delta t_n^2}{3} + \frac{\Delta t_n}{2} (\Delta t_{n-1} + \Delta t_{n-2}) \right], \\ \beta_{n-2} &= \frac{1}{\Delta t_{n-2}(\Delta t_{n-1} + \Delta t_{n-2})} \left( \frac{\Delta t_n^2}{3} + \frac{\Delta t_n \Delta t_{n-1}}{2} \right). \end{aligned}$$

2.2.4. Krylov IIF schemes for ADR systems

Time step sizes in IIF schemes for ADR systems ((17), (20), (21)) are non-uniform in general. Hence products of a matrix exponential and a vector need to be performed at every time step. Directly computing and storing exponential matrices for two or three spatial dimensional problems at every time step are not practical for a typical computer. In order to efficiently implement the IIF schemes for ADR systems ((17), (20), (21)), we use the Krylov subspace method [12,31] to approximate the product of a matrix exponential and a vector, as that in the Krylov IIF schemes for solving reaction–diffusion systems [7]. First we briefly describe the Krylov subspace method to approximate the product of a matrix exponential and a vector, e.g.  $e^{C\Delta t}\vec{v}$ . The large sparse matrix  $C$  is projected to the Krylov subspace

$$K_M = \text{span}\{\vec{v}, C\vec{v}, C^2\vec{v}, \dots, C^{M-1}\vec{v}\}. \tag{22}$$

The dimension  $M$  of the Krylov subspace is *much* smaller than the dimension  $N$  of the large sparse matrix  $C$ . In all numerical computations of this paper, we take  $M = 25$  for different  $N$ , and accurate results are obtained as shown in Section 4. The well-known Arnoldi algorithm [42] generates an orthonormal basis  $V_M = [v_1, v_2, v_3, \dots, v_M]$  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

$$e^{C\Delta t}\vec{v} \simeq \gamma V_M e^{H_M \Delta t} e_1, \tag{23}$$

where  $\gamma = \|\vec{v}\|_2$ , 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 computational cost of  $O(M^2)$ , see [12,31,7]. Applying the Krylov subspace approximation (23) to (17), we obtain the  $r$ -th order Krylov IIF scheme for ADR equations

$$\begin{aligned} \vec{U}_{n+1} &= \Delta t_n \alpha_{n+1} \vec{R}(\vec{U}_{n+1}) + \gamma_{0,n} V_{M,0,n} e^{H_{M,0,n} \Delta t_n} e_1 \\ &+ \Delta t_n \left( \beta_{n+1-r} \gamma_{1-r,n} V_{M,1-r,n} e^{H_{M,1-r,n}(\Delta t_n - \tau_{1-r})} e_1 + \sum_{i=2-r}^{-1} \gamma_{i,n} V_{M,i,n} e^{H_{M,i,n}(\Delta t_n - \tau_i)} e_1 \right), \end{aligned} \tag{24}$$

where  $\gamma_{0,n} = \|U_n + \Delta t_n(\alpha_n \vec{R}(\vec{U}_n) + \beta_n \vec{F}(\vec{U}_n))\|_2$ ,  $V_{M,0,n}$  and  $H_{M,0,n}$  are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vector  $U_n + \Delta t_n(\alpha_n \vec{R}(\vec{U}_n) + \beta_n \vec{F}(\vec{U}_n))$ .  $\gamma_{1-r,n} = \|\vec{F}(\vec{U}_{n+1-r})\|_2$ ,  $V_{M,1-r,n}$  and  $H_{M,1-r,n}$  are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vector  $\vec{F}(\vec{U}_{n+1-r})$ .  $\gamma_{i,n} = \|\alpha_{n+i} \vec{R}(\vec{U}_{n+i}) + \beta_{n+i} \vec{F}(\vec{U}_{n+i})\|_2$ ,  $V_{M,i,n}$  and  $H_{M,i,n}$  are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vectors  $\alpha_{n+i} \vec{R}(\vec{U}_{n+i}) + \beta_{n+i} \vec{F}(\vec{U}_{n+i})$ , for  $i = 2 - r, 3 - r, \dots, -1$ . Notice that  $V_{M,0,n}$ ,  $V_{M,1-r,n}$  and  $V_{M,i,n}$ ,  $i = 2 - r, 3 - r, \dots, -1$  are orthonormal bases of different Krylov subspaces for the same matrix  $C$ , which are generated with different initial vectors in the Arnoldi algorithm. The value of  $M$  is taken to be large enough such that the errors of Krylov subspace approximations are much less than the truncation errors of the numerical schemes (17). From our numerical experiments in this paper, we can see that our numerical schemes have already given a clear accuracy order with a very small size  $M = 25$ , and  $M$  does *not* need to be increased when the spatial-temporal resolution is refined. Specifically, the second order Krylov IIF scheme has the following form

$$\vec{U}_{n+1} = \frac{1}{2} \Delta t_n \vec{R}(\vec{U}_{n+1}) + \gamma_{0,n} V_{M,0,n} e^{H_{M,0,n} \Delta t_n} e_1 - \frac{(\Delta t_n)^2}{2\Delta t_{n-1}} (\gamma_{-1,n} V_{M,-1,n} e^{H_{M,-1,n}(\Delta t_n + \Delta t_{n-1})} e_1), \tag{25}$$

where  $\gamma_{0,n} = \|U_n + \Delta t_n(\frac{1}{2} \vec{R}(\vec{U}_n) + \frac{1}{\Delta t_{n-1}}(\frac{\Delta t_n}{2} + \Delta t_{n-1}) \vec{F}(\vec{U}_n))\|_2$ ,  $V_{M,0,n}$  and  $H_{M,0,n}$  are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vector  $U_n + \Delta t_n(\frac{1}{2} \vec{R}(\vec{U}_n) + \frac{1}{\Delta t_{n-1}}(\frac{\Delta t_n}{2} + \Delta t_{n-1}) \vec{F}(\vec{U}_n))$ .  $\gamma_{-1,n} = \|\vec{F}(\vec{U}_{n-1})\|_2$ ,  $V_{M,-1,n}$  and  $H_{M,-1,n}$  are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vector  $\vec{F}(\vec{U}_{n-1})$ . And the third order Krylov IIF scheme has the form

$$\begin{aligned} \vec{U}_{n+1} &= \frac{2\Delta t_n + 3\Delta t_{n-1}}{6(\Delta t_n + \Delta t_{n-1})} \Delta t_n \vec{R}(\vec{U}_{n+1}) + \gamma_{0,n} V_{M,0,n} e^{H_{M,0,n} \Delta t_n} e_1 \\ &+ \Delta t_n \left( \frac{2(\Delta t_n)^2 + 3\Delta t_n \Delta t_{n-1}}{6\Delta t_{n-2}(\Delta t_{n-1} + \Delta t_{n-2})} \gamma_{-2,n} V_{M,-2,n} e^{H_{M,-2,n}(\Delta t_n + \Delta t_{n-1} + \Delta t_{n-2})} e_1 \right. \end{aligned}$$

$$+ \gamma_{-1,n} V_{M,-1,n} e^{H_{M,-1,n}(\Delta t_n + \Delta t_{n-1})} e_1 \Big), \tag{26}$$

where  $\gamma_{0,n} = \|U_n + \Delta t_n(\alpha_n \vec{R}(\vec{U}_n) + \beta_n \vec{F}(\vec{U}_n))\|_2$ ,  $V_{M,0,n}$  and  $H_{M,0,n}$  are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vector  $U_n + \Delta t_n(\alpha_n \vec{R}(\vec{U}_n) + \beta_n \vec{F}(\vec{U}_n))$ .  $\gamma_{-2,n} = \|\vec{F}(\vec{U}_{n-2})\|_2$ ,  $V_{M,-2,n}$  and  $H_{M,-2,n}$  are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vector  $\vec{F}(\vec{U}_{n-2})$ .  $\gamma_{-1,n} = \|\alpha_{n-1} \vec{R}(\vec{U}_{n-1}) + \beta_{n-1} \vec{F}(\vec{U}_{n-1})\|_2$ ,  $V_{M,-1,n}$  and  $H_{M,-1,n}$  are orthonormal basis and upper Hessenberg matrix generated by the Arnoldi algorithm with the initial vectors  $\alpha_{n-1} \vec{R}(\vec{U}_{n-1}) + \beta_{n-1} \vec{F}(\vec{U}_{n-1})$ . See Eq. (21) for values of  $\alpha_n, \beta_n, \alpha_{n-1}, \beta_{n-1}$ .

**Remark.** We would like to emphasize that in the implementation of the methods, we do *not* store matrices  $C$  or  $C_n$ , because only multiplications of matrices  $C$  or  $C_n$  with a vector are needed in the methods, and they correspond to certain finite difference operations.

**Remark.** We would like to emphasize that in our Krylov IIF schemes (24), (25), (26) for ADR systems, the “local implicit” property of the original IIF schemes in [33] is preserved well. 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 linear computational complexity of our high order Krylov IIF schemes, as shown in the numerical experiments section. The small size implicit nonlinear system can be efficiently solved by a fixed-point iteration method [33] or a Newton iteration method [7].

### 3. Linear analysis

Similar to the previous approaches [36,41,33,44], we perform linear analysis for the new IIF schemes (20) and (21).

#### 3.1. Truncation error

In the spatial direction a third order WENO scheme is applied to the advection term, and a second order central difference scheme (when IIF2 is used) or a fourth order central difference scheme (when IIF3 is used) is applied to the diffusion term. Hence the overall spatial discretization is of order two or order three. We focus on analyzing the truncation errors of the IIF2 scheme (20) and the IIF3 scheme (21) for ADR systems, i.e., the local temporal truncation errors. Consider the following linear semi-discretization system

$$\frac{d\mathbf{u}}{dt} = A\mathbf{u} + D\mathbf{u} + R\mathbf{u}, \tag{27}$$

where  $A, D$ , and  $R$  are matrices derived from linear spatial discretizations of advection, diffusion and reaction terms of a linear ADR system respectively, and  $\mathbf{u}$  is the vector of unknown numerical values. First, we apply the IIF2 scheme (20) to the system (27) to obtain  $\mathbf{u}_{n+1}$  in terms of  $\mathbf{u}_n$  and  $\mathbf{u}_{n-1}$ :

$$\mathbf{u}_{n+1} = \left(I - \frac{R}{2}\Delta t\right)^{-1} e^{D\Delta t} \left(I + \frac{R}{2}\Delta t + \frac{3A}{2}\Delta t\right) \mathbf{u}_n - \frac{\Delta t}{2} \left(I - \frac{R}{2}\Delta t\right)^{-1} e^{2D\Delta t} A \mathbf{u}_{n-1}. \tag{28}$$

To derive the local truncation error, we substitute the exact solution of (27) into the right-hand side of Eq. (28) and use Taylor expansion. Denoting the exact solution of (27) by  $\mathbf{u}(t)$ , we replace  $\mathbf{u}_n$  and  $\mathbf{u}_{n-1}$  by the exact solution values  $\mathbf{u}(t_n)$  and  $\mathbf{u}(t_{n-1})$  in (28). Since  $\mathbf{u}(t_{n-1}) = e^{-(A+D+R)\Delta t} \mathbf{u}(t_n)$ , we obtain

$$\begin{aligned} \mathbf{u}_{n+1} &= \left(I + \frac{R}{2}\Delta t + \left(\frac{R}{2}\right)^2 \Delta t^2 + \dots\right) \left(I + D\Delta t + \frac{D^2}{2}\Delta t^2 + \dots\right) \left(I + \frac{R}{2}\Delta t + \frac{3A}{2}\Delta t\right) \mathbf{u}(t_n) \\ &\quad - \frac{\Delta t}{2} \left(I + \frac{R}{2}\Delta t + \dots\right) (I + 2D\Delta t + \dots) A (I + (-A - D - R)\Delta t + \dots) \mathbf{u}(t_n) \\ &= \left(I + \left(R + D + \frac{3A}{2}\right)\Delta t + \left(\frac{R^2}{2} + \frac{D^2}{2} + \frac{RD}{2} + \frac{3}{4}RA + \frac{DR}{2} + \frac{3DA}{2}\right)\Delta t^2 + \dots\right) \mathbf{u}(t_n) \\ &\quad - \left(\frac{A}{2}\Delta t + \left(\frac{RA}{4} + DA - \frac{A^2}{2} - \frac{AD}{2} - \frac{AR}{2}\right)\Delta t^2 + \dots\right) \mathbf{u}(t_n) \\ &= \left(I + (A + D + R)\Delta t + \frac{(A + D + R)^2}{2}\Delta t^2 + \dots\right) \mathbf{u}(t_n). \end{aligned} \tag{29}$$

Hence, the local truncation error of the second order IIF method (20) is

$$\left( I + (A + D + R)\Delta t + \frac{(A + D + R)^2}{2}\Delta t^2 + \dots \right) \mathbf{u}(t_n) - e^{(A+D+R)\Delta t} \mathbf{u}(t_n) = O(\Delta t^3) \mathbf{u}(t_n). \tag{30}$$

Similarly for the third order scheme (21) for ADR systems, we apply it to the system (27) to obtain

$$\begin{aligned} \mathbf{u}_{n+1} &= \left( I - \frac{5R}{12}\Delta t \right)^{-1} e^{D\Delta t} \left( I + \frac{2R}{3}\Delta t + \frac{23A}{12}\Delta t \right) \mathbf{u}_n \\ &\quad - \left( I - \frac{5R}{12}\Delta t \right)^{-1} e^{2D\Delta t} \left( \frac{R}{12}\Delta t + \frac{4A}{3}\Delta t \right) \mathbf{u}_{n-1} + \Delta t \left( I - \frac{5R}{12}\Delta t \right)^{-1} e^{3D\Delta t} \frac{5A}{12} \mathbf{u}_{n-2}. \end{aligned} \tag{31}$$

Using the exact solution of (27), we replace  $\mathbf{u}_{n-1}$  by  $e^{-(A+D+R)\Delta t} \mathbf{u}(t_n)$  and  $\mathbf{u}_{n-2}$  by  $e^{-2(A+D+R)\Delta t} \mathbf{u}(t_n)$ , and use Taylor expansion to obtain

$$\begin{aligned} \mathbf{u}_{n+1} &= \left( I + \frac{5R}{12}\Delta t + \left( \frac{5R}{12} \right)^2 \Delta t^2 + \left( \frac{5R}{12} \right)^3 \Delta t^3 + \dots \right) \left( I + D\Delta t + \frac{D^2}{2}\Delta t^2 + \frac{D^3}{6}\Delta t^3 + \dots \right) \\ &\quad \times \left( I + \frac{2R}{3}\Delta t + \frac{23A}{12}\Delta t \right) \mathbf{u}(t_n) - \left( I + \frac{5R}{12}\Delta t + \left( \frac{5R}{12} \right)^2 \Delta t^2 + \left( \frac{5R}{12} \right)^3 \Delta t^3 + \dots \right) \\ &\quad \times \left( I + 2D\Delta t + \frac{4D^2}{2}\Delta t^2 + \frac{8D^3}{6}\Delta t^3 + \dots \right) \left( \frac{R}{12}\Delta t + \frac{4A}{3}\Delta t \right) \\ &\quad \times \left( I + (-D - A - R)\Delta t + \frac{((-D - A - R)^2)}{2}\Delta t^2 + \dots \right) \mathbf{u}(t_n) \\ &\quad + \left( I + \frac{5R}{12}\Delta t + \left( \frac{5R}{12} \right)^2 \Delta t^2 + \dots \right) \left( I + 3D\Delta t + \frac{9D^2}{2}\Delta t^2 + \frac{27D^3}{6}\Delta t^3 + \dots \right) \frac{5A}{12} \\ &\quad \times \left( I + (-2D - 2A - 2R)\Delta t + \frac{((-2D - 2A - 2R)^2)}{2}\Delta t^2 + \dots \right) \mathbf{u}(t_n) \\ &= \left( I + (A + D + R)\Delta t + \frac{(A + D + R)^2}{2}\Delta t^2 + \frac{(A + D + R)^3}{6}\Delta t^3 + \dots \right) \mathbf{u}(t_n). \end{aligned} \tag{32}$$

Hence, the local truncation error of the third order IIF method (21) is

$$\left( I + (A + D + R)\Delta t + \frac{(A + D + R)^2}{2}\Delta t^2 + \frac{(A + D + R)^3}{6}\Delta t^3 + \dots \right) \mathbf{u}(t_n) - e^{(A+D+R)\Delta t} \mathbf{u}(t_n) = O(\Delta t^4) \mathbf{u}(t_n). \tag{33}$$

### 3.2. Linear stability

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

$$u_t = au - du + ru, \quad \text{with } r \in \mathcal{C}, \text{ and } a, d \in \mathcal{R}, d > 0. \tag{34}$$

Similar to the stability analysis approaches in [33,44], 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  $a\Delta t$ , for the second and third order methods. 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.

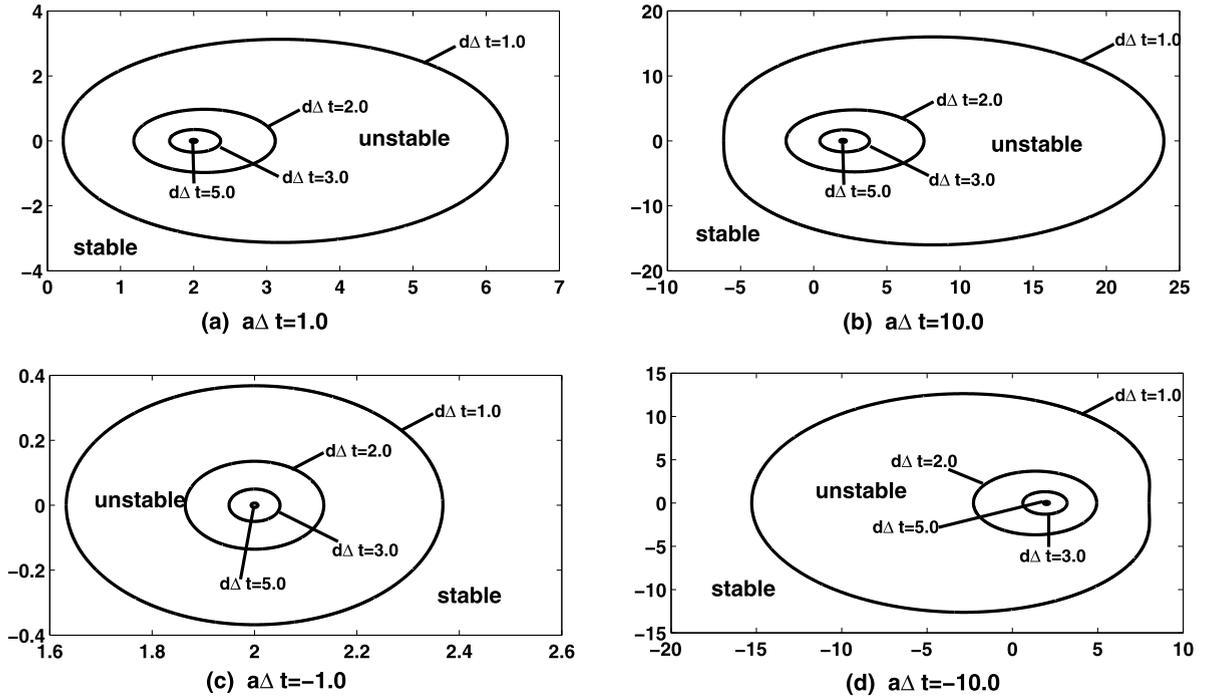
Applying the second order IIF (20) to Eq. (34), then substituting  $u_n = e^{in\theta}$  into the resulting equation, we obtain

$$\left( 1 - \frac{\lambda}{2} \right) e^{2i\theta} = e^{-d\Delta t} \left( 1 + \frac{\lambda}{2} + \frac{3}{2}a\Delta t \right) e^{i\theta} - \frac{a}{2}\Delta t e^{-2d\Delta t}, \tag{35}$$

where  $\lambda = r\Delta t$  has a real part  $\lambda_r$  and imaginary part  $\lambda_i$ . Therefore, the equations for  $\lambda_r$  and  $\lambda_i$  are

$$\begin{cases} \lambda_r = \frac{B_1 C_2 - B_2 C_1}{A_1 B_2 - A_2 B_1}; \\ \lambda_i = \frac{A_1 C_2 - A_2 C_1}{A_2 B_1 - A_1 B_2}, \end{cases} \tag{36}$$

where



**Fig. 1.** Linear stability regions of the second order IIF scheme (20) for different values of  $d\Delta t$  under a fixed value of  $a\Delta t$ . (a)  $a\Delta t = 1.0$ ; (b)  $a\Delta t = 10.0$ ; (c)  $a\Delta t = -1.0$ ; (d)  $a\Delta t = -10.0$ .

$$\begin{cases} A_1 = e^{-d\Delta t} \frac{1}{2} \cos \theta + \frac{1}{2} \cos 2\theta, \\ B_1 = -e^{-d\Delta t} \frac{1}{2} \sin \theta - \frac{1}{2} \sin 2\theta, \\ C_1 = -\frac{a}{2} \Delta t e^{-2d\Delta t} + e^{-d\Delta t} \left( 1 + \frac{3}{2} a \Delta t \right) \cos \theta - \cos 2\theta, \\ A_2 = e^{-d\Delta t} \frac{1}{2} \sin \theta + \frac{1}{2} \sin 2\theta, \\ B_2 = e^{-d\Delta t} \frac{1}{2} \cos \theta + \frac{1}{2} \cos 2\theta, \\ C_2 = e^{-d\Delta t} \left( 1 + \frac{3}{2} a \Delta t \right) \sin \theta - \sin 2\theta. \end{cases} \quad (37)$$

We first study stability regions in the complex plane of  $r\Delta t$  for different values of  $d\Delta t$  under a fixed value of  $a\Delta t$ . We choose four different values  $a\Delta t = 1.0$ ,  $a\Delta t = 10.0$ ,  $a\Delta t = -1.0$  and  $a\Delta t = -10.0$  as examples. Based on analyzing the roots of the characteristic polynomials of (35), the stability regions always include the point  $\lambda = (-20, 0)$  for any values of  $d\Delta t$  used in Fig. 1. From Fig. 1 we see that for a fixed  $a\Delta t$ , the stable region becomes bigger with the increase of the value of  $d\Delta t$ . Next we plot stability regions for different values of  $a\Delta t$  under a fixed value of  $d\Delta t$ .  $d\Delta t = 1.0$ ,  $d\Delta t = 2.0$ ,  $d\Delta t = 10.0$  and  $d\Delta t = 20.0$  are chosen as examples. Again, analysis of the characteristic polynomials shows that the point  $\lambda = (-10, 0)$  is always included in the stable regions for the chosen values of  $d\Delta t$ . See Fig. 2 for the stability regions of this case. From Fig. 2 we see that for a fixed  $d\Delta t$ , the stable region becomes smaller with the increase of the value of  $|a|\Delta t$ . We conclude that the diffusion term tends to stabilize the scheme, while the advection term gives constraints on time step sizes. Due to the implicit property of the scheme, the stability regions are quite large and often include the whole left complex plane, with a relative big size diffusion parameter  $d$  and mild size advection parameter  $a$ .

Next, we analyze the third order IIF scheme (21) for ADR systems. Using the same approach, we apply the third order IIF scheme (21) to Eq. (34), then substitute  $u_n = e^{in\theta}$  into the resulting equation and obtain the equation for  $\lambda$ :

$$\left( 1 - \frac{5}{12} \lambda \right) e^{3i\theta} = e^{-d\Delta t} \left( 1 + \frac{2}{3} \lambda + \frac{23}{12} a \Delta t \right) e^{2i\theta} - e^{-2d\Delta t} \left( \frac{\lambda}{12} + \frac{4}{3} a \Delta t \right) e^{i\theta} + \frac{5}{12} a \Delta t e^{-3d\Delta t}. \quad (38)$$

The equations for the real part  $\lambda_r$  and the imaginary part  $\lambda_i$  are

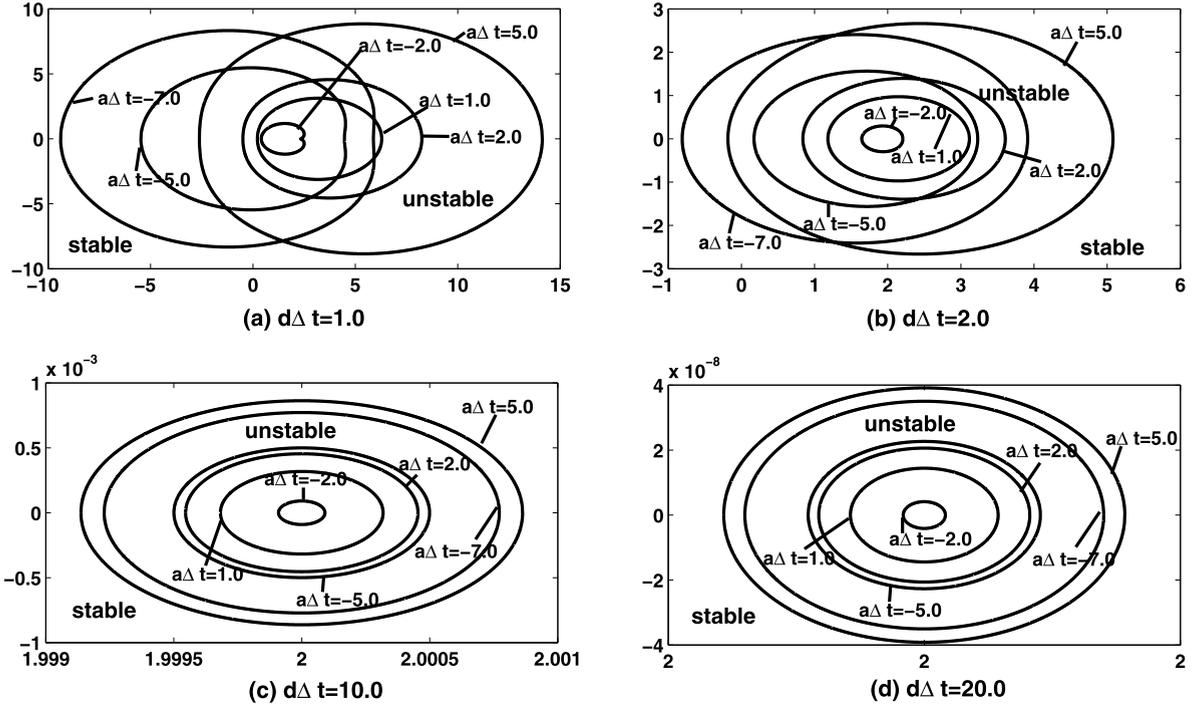


Fig. 2. Linear stability regions of the second order IIF scheme (20) for different values of  $a\Delta t$  under a fixed value of  $d\Delta t$ . (a)  $d\Delta t = 1.0$ ; (b)  $d\Delta t = 2.0$ ; (c)  $d\Delta t = 10.0$ ; (d)  $d\Delta t = 20.0$ .

$$\begin{cases} \lambda_r = \frac{B_1 C_2 - B_2 C_1}{A_1 B_2 - A_2 B_1}; \\ \lambda_i = \frac{A_1 C_2 - A_2 C_1}{A_2 B_1 - A_1 B_2}, \end{cases} \quad (39)$$

where

$$\begin{cases} A_1 = \frac{2}{3}e^{-d\Delta t} \cos 2\theta - \frac{1}{12}e^{-2d\Delta t} \cos \theta + \frac{5}{12} \cos 3\theta, \\ B_1 = -\frac{2}{3}e^{-d\Delta t} \sin 2\theta + \frac{1}{12}e^{-2d\Delta t} \sin \theta - \frac{2}{12} \sin 3\theta, \\ C_1 = \left(1 + \frac{23}{12}a\Delta t\right)e^{-d\Delta t} \cos 2\theta - \frac{4}{3}a\Delta te^{-2d\Delta t} \cos \theta + \frac{5}{12}a\Delta te^{-3d\Delta t} - \cos 3\theta, \\ A_2 = -\frac{2}{3}e^{-d\Delta t} \sin 2\theta + \frac{1}{12}e^{-2d\Delta t} \sin \theta - \frac{2}{12} \sin 3\theta, \\ B_2 = \frac{2}{3}e^{-d\Delta t} \cos 2\theta - \frac{1}{12}e^{-2d\Delta t} \cos \theta + \frac{5}{12} \cos 3\theta, \\ C_2 = \left(1 + \frac{23}{12}a\Delta t\right)e^{-d\Delta t} \sin 2\theta - \frac{4}{3}a\Delta te^{-2d\Delta t} \sin \theta - \sin 3\theta. \end{cases} \quad (40)$$

The stability regions are plotted in Figs. 3 and 4. Fig. 3 shows the stability regions for different values of  $d\Delta t$  under a fixed value of  $a\Delta t$ :  $a\Delta t = 1.0$ ,  $a\Delta t = 10.0$ ,  $a\Delta t = -1.0$  and  $a\Delta t = -10.0$ . In this case, the point  $\lambda = (-50, 0)$  is always included in the stable regions. Fig. 4 shows stability regions for different values of  $a\Delta t$  under a fixed value of  $d\Delta t$ :  $d\Delta t = 1.0$ ,  $d\Delta t = 2.0$ ,  $d\Delta t = 10.0$  and  $d\Delta t = 20.0$ . The point  $\lambda = (-40, 0)$  is always included in the stable regions under these parameters. The same conclusion as the IIF2 case can be drawn by analyzing Figs. 3 and 4. In general, the IIF2 scheme has bigger stability regions than the IIF3 scheme.

#### 4. Numerical experiments

In this section we present numerical examples to show the stability, accuracy and efficiency of the Krylov IIF-WENO methods for solving nonlinear advection–diffusion–reaction PDEs. The methods are firstly tested on a set of problems with exact solutions. Then application of the methods to a biological system will be shown. The Krylov subspace dimension  $M$  is

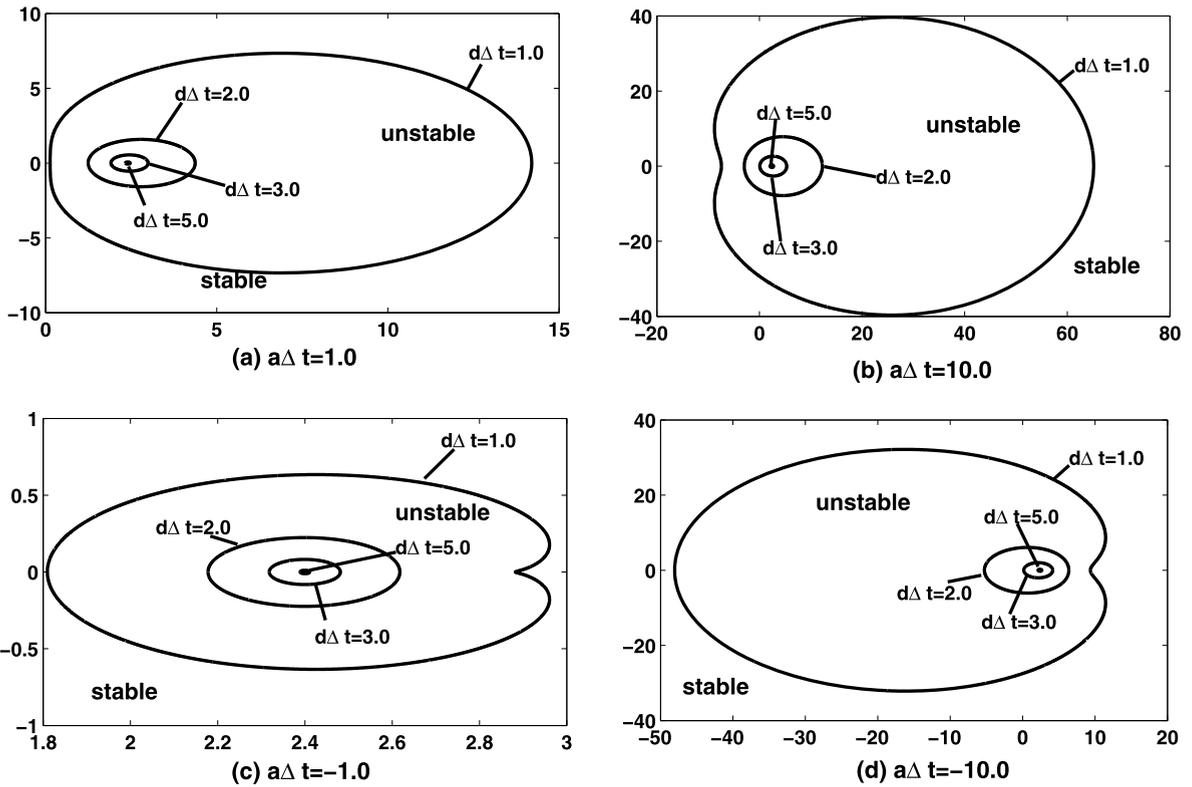


Fig. 3. Linear stability regions of the third order IIF scheme (21) for different values of  $d\Delta t$  under a fixed value of  $a\Delta t$ . (a)  $a\Delta t = 1.0$ ; (b)  $a\Delta t = 10.0$ ; (c)  $a\Delta t = -1.0$ ; (d)  $a\Delta t = -10.0$ .

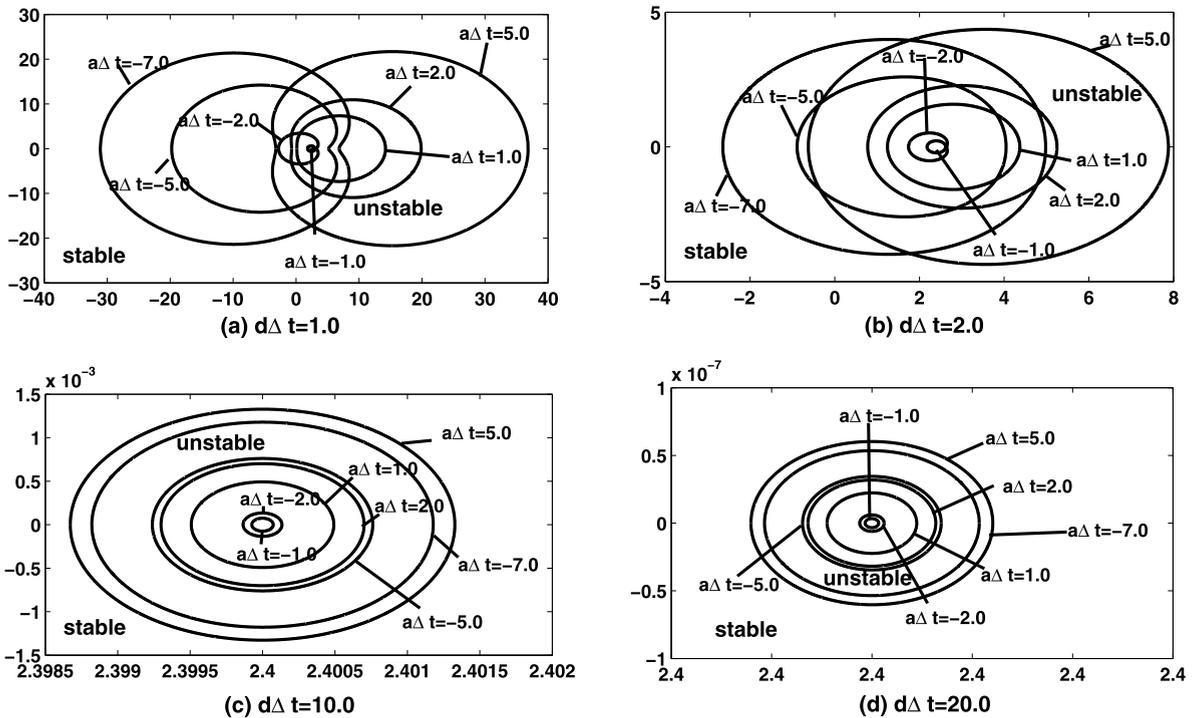


Fig. 4. Linear stability regions of the third order IIF scheme (21) for different values of  $a\Delta t$  under a fixed value of  $d\Delta t$ . (a)  $d\Delta t = 1.0$ ; (b)  $d\Delta t = 2.0$ ; (c)  $d\Delta t = 10.0$ ; (d)  $d\Delta t = 20.0$ .

**Table 1**

**Example 1**, the two-dimensional equation (41). CPU time, error, and order of accuracy of the Krylov IIF2 and IIF3 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.

Krylov IIF2					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	9.28E-2	–	4.28E-2	–	0.56
40	3.48E-2	1.42	1.52E-2	1.49	2.65
80	1.03E-2	1.76	4.50E-3	1.76	10.02
160	2.90E-3	1.83	1.30E-3	1.79	77.62
320	7.50E-4	1.95	3.29E-4	1.98	623.96
Krylov IIF3					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	4.13E-2	–	2.25E-2	–	1.00
40	1.11E-2	1.90	5.40E-3	2.06	3.90
80	2.10E-3	2.40	9.66E-4	2.48	24.58
160	3.23E-4	2.70	1.49E-4	2.70	123.38
320	4.43E-5	2.87	2.03E-5	2.88	990.98

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 advection–diffusion–reaction systems. Here large time step size means that the time step size is only 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 Krylov IIF methods for ADR PDEs are multistep methods. To start the computations at the first few time steps, we use the TVD Runge–Kutta methods [39]. Specifically, the second order TVD Runge–Kutta method is used for the first time step in IIF2, and the third order TVD Runge–Kutta method is used for the first and the second time steps in IIF3. The second order central scheme (7) is used for the diffusion terms when the IIF2 is applied, and the fourth order central scheme (8) for the diffusion terms is coupled with the IIF3 scheme.

**Example 1** (Semilinear equations). We first test the methods for solving semilinear ADR equations on two and three spatial dimensions. The two-dimensional equation is

$$\begin{cases} u_t + \left(\frac{1}{2}u^2\right)_x + \left(\frac{1}{2}u^2\right)_y = u_{xx} + u_{yy} + 2u + \cos(x + y + t)(1 + 2 \sin(x + y + t)), \\ u(x, y, 0) = \sin(x + y), \quad 0 \leq x, y \leq 2\pi, \end{cases} \tag{41}$$

with periodic boundary conditions, and the exact solution is  $u(x, y, t) = \sin(x + y + t)$ . The three-dimensional equation is

$$\begin{cases} u_t + \left(\frac{1}{2}u^2\right)_x + \left(\frac{1}{2}u^2\right)_y + \left(\frac{1}{2}u^2\right)_z \\ = u_{xx} + u_{yy} + u_{zz} + 3u + \cos(x + y + z + t)(1 + 3 \sin(x + y + z + t)), \\ u(x, y, z, 0) = \sin(x + y + z), \quad 0 \leq x, y, z \leq 2\pi, \end{cases} \tag{42}$$

with periodic boundary conditions, and the exact solution is  $u(x, y, z, t) = \sin(x + y + z + t)$ . The computation is carried up to  $T = 1.0$  with  $M = 25$  at which the  $L^1$  and  $L^\infty$  errors are measured. The CFL number for the advection term is taken to be  $CFL = 0.5$ . CPU time, error, and order of accuracy of the Krylov IIF2 and IIF3 methods with the third order WENO spatial discretization for the advection term are reported in Table 1 for the two-dimensional equation (41), and Table 3 for the three-dimensional equation (42). We can observe that we obtain desired accuracy orders for both cases.

Now we analyze the computational cost. In the computation, the time step size  $\Delta t = O(\Delta x)$ . This is consistent with our goal of using a large time step size proportional to the spatial grid size for a stable and accurate computation of a parabolic PDE. In each time step, the major CPU times are spent in WENO procedure, the Arnoldi algorithm for generating orthonormal basis and upper Hessenberg matrix in Krylov subspace approximation, and matrix-vector products in the schemes (25), (26). Computational costs associated with each of these procedures linearly depend on the total number of spatial grid points. For example, a two-dimensional problem has  $N^2$  spatial grid points, where  $N$  is the number of grid points in each spatial direction. The WENO procedure performs approximations to the numerical fluxes, and its operations at a grid point are a constant which only depends on the number of points in the stencil (it is 4 for the third order WENO scheme). So the cost of WENO procedure is  $O(N^2)$ . In the Arnoldi procedure, because the dimension of the Krylov subspace approximation is a constant  $M$  ( $M = 25$  here), the total operation for forming the  $N^2 \times M$  orthonormal basis matrix and  $M \times M$  upper Hessenberg matrix is  $O(M \cdot N^2)$ . For matrix-vector products in the schemes (25) and (26), the total operation is still  $O(M \cdot N^2)$ . Hence in each time step, the total operation is  $O(N^2)$  for a two-dimensional problem. Similarly, it is  $O(N^3)$  for a three-dimensional problem. Since time step size  $\Delta t = O(\Delta x)$ , the total computational cost linearly depends on the total number of spatio-temporal grid points. All computations in this paper are implemented by MATLAB codes.

**Table 2**  
CPU time (seconds) Analysis for Example 1, the two-dimensional equation (41).

Krylov IIF2					
<i>N</i>	Total CPU	WENO	Krylov	Others	
20	0.56	0.18	0.12	0.26	
40	2.65	1.48	0.38	0.79	
80	10.02	5.94	1.19	2.89	
160	77.62	44.75	10.57	22.30	
320	623.96	358.59	100.77	164.60	
Krylov IIF3					
<i>N</i>	Total CPU	WENO	Krylov	Others	
20	1.00	0.20	0.25	0.56	
40	3.90	1.63	0.82	1.46	
80	24.58	10.90	4.99	8.69	
160	123.38	55.47	25.46	42.45	
320	990.98	432.82	240.63	317.52	

**Table 3**  
Example 1, the three-dimensional equation (42). CPU time, error, and order of accuracy of the Krylov IIF2 and IIF3 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.

Krylov IIF2					
<i>N</i>	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
10	9.54E-2	–	4.44E-2	–	2.85
20	3.54E-2	1.43	1.54E-2	1.53	13.81
40	1.49E-2	1.25	7.40E-3	1.06	154.80
80	4.28E-3	1.80	2.18E-3	1.76	2400.61
160	1.13E-3	1.92	5.76E-4	1.92	41 444.05
Krylov IIF3					
<i>N</i>	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
10	1.70E-1	–	8.38E-2	–	2.45
20	2.74E-2	2.63	1.17E-2	2.84	17.38
40	2.12E-3	3.69	1.02E-3	3.52	232.13
80	2.70E-4	2.97	1.37E-4	2.90	3274.94
160	4.00E-5	2.75	1.90E-5	2.85	54 347.57

The CPU time observed in our implementation asymptotically approximates the linear dependence on the total number of spatio-temporal grid points. This is possibly due to the overhead time expenses and automatic code optimization of MATLAB program platform. We list CPU times of major procedures for the two-dimensional problem in Table 2. We can see that for a bigger  $N$ , the linear dependence is more obvious for every part of the computation.

**Example 2 (Fully nonlinear equations I).** We test the methods for solving fully nonlinear ADR equations. First we solve the one-dimensional ADR equation with a nonlinear diffusion term

$$\begin{cases} u_t = (uu_x)_x - u_x + u - 1 - 0.25 \cos(2x - 2t), & 0 \leq x \leq 2\pi, \\ u(x, 0) = 1 + 0.5 \sin(x), \end{cases} \tag{43}$$

with periodic boundary condition, and the exact solution is  $u = 1 + 0.5 \sin(x - t)$ . As an example to show the structure of a Jacobian matrix in IIF schemes for fully nonlinear problems, the Jacobian matrix by using the second order central difference scheme (7) in the IIF2 scheme for this case is

$$C_n = \begin{pmatrix} -\frac{2u_1}{h^2} & \frac{u_2}{h^2} & 0 & \cdots & 0 & 0 & \frac{u_N}{h^2} \\ \frac{u_1}{h^2} & -\frac{2u_2}{h^2} & \frac{u_3}{h^2} & 0 & \cdots & \cdots & 0 \\ 0 & \frac{u_2}{h^2} & -\frac{2u_3}{h^2} & \frac{u_4}{h^2} & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{u_1}{h^2} & \cdots & \cdots & 0 & 0 & \frac{u_{N-1}}{h^2} & -\frac{2u_N}{h^2} \end{pmatrix}_{N \times N}.$$

Again, we do not need to store the matrix  $C_n$ , but just perform corresponding operations. The two-dimensional equation is

$$\begin{cases} u_t = \nabla \cdot (u \nabla u) - 0.5(u_x + u_y) + 2u - 0.5 \cos(2x + 2y - 2t) - 2, & 0 \leq x, y \leq 2\pi, \\ u(x, y, 0) = 1 + 0.5 \sin(x + y), \end{cases} \tag{44}$$

**Table 4**

**Example 2**, the one-dimensional equation (43). CPU time, error, and order of accuracy of the Krylov IIF2 and IIF3 methods with the third order WENO spatial discretization for the advection term. Final time  $T = 1.0$ .  $N$  is the number of grid points.

Krylov IIF2 for nonlinear problems $\Delta t = 0.4h$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
10	3.37E-2	–	1.76E-2	–	0.17
20	1.91E-2	0.82	8.00E-3	1.14	0.25
40	6.30E-3	1.60	2.80E-3	1.51	0.48
80	1.70E-3	1.89	7.95E-4	1.82	1.87
160	4.50E-4	1.92	2.12E-4	1.91	4.13
320	1.18E-4	1.93	5.50E-5	1.95	9.13
Krylov IIF3 for nonlinear problems $\Delta t = 0.3h$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
10	5.69E-2	–	1.87E-2	–	0.22
20	1.08E-2	2.40	3.40E-3	2.46	0.50
40	1.30E-3	3.05	4.80E-4	2.82	1.58
80	1.52E-4	3.10	5.83E-5	3.04	9.66
160	2.01E-5	2.92	7.41E-6	2.98	27.89
320	2.60E-6	2.95	9.42E-7	2.98	84.99

**Table 5**

**Example 2**, the one-dimensional equation (43). CPU time, error, and order of accuracy of the IIF2 and IIF3 methods with the third order WENO spatial discretization for the advection term. Final time  $T = 1.0$ .  $N$  is the number of grid points. Krylov subspace approximation is not used.

IIF2 for nonlinear problems (no Krylov) $\Delta t = 0.4h$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	1.92E-2	–	8.10E-3	–	0.23
40	6.40E-3	1.59	2.80E-3	1.53	0.56
80	1.80E-3	1.83	8.03E-4	1.80	1.38
160	4.59E-4	1.97	2.15E-4	1.90	7.36
320	1.19E-4	1.95	5.54E-5	1.96	33.12
IIF3 for nonlinear problems (no Krylov) $\Delta t = 0.3h$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	1.08E-2	–	3.40E-3	–	0.30
40	1.30E-3	3.05	4.80E-4	2.82	0.72
80	1.52E-4	3.10	5.83E-5	3.04	3.18
160	2.01E-5	2.92	7.41E-6	2.98	14.98
320	2.60E-6	2.95	9.42E-7	2.98	81.85

with periodic boundary condition, and the exact solution is  $u = 1 + 0.5 \sin(x + y - t)$ . CPU time, error, and order of accuracy of the Krylov IIF2-WENO and IIF3-WENO methods are reported in Table 4 for the one-dimensional equation (43), and Table 6 for the two-dimensional equation (44). We can observe that we obtain desired accuracy orders for this fully nonlinear problem. Similar as Example 1, the large time step size  $\Delta t = O(\Delta x)$  is obtained for a stable and accurate computation of a nonlinear parabolic PDE. The CPU time approximately linearly depends on the number of spatio-temporal grid points. For the one-dimensional problem (43), we can directly compute the exponential matrix without using Krylov subspace approximations since the matrix size is not too big. The exponential matrix can be computed and stored before the time evolution process. The new numerical results are shown in Table 5. We can see that the numerical accuracy and CPU times are comparable for these two different approaches.

**Remark.** When time step sizes are a constant, the approach of computing and storing exponential matrix before the time evolution process (e.g. in the original IIF methods [33] or the compact IIF methods [34]) is an efficient way for semilinear reaction–diffusion systems. However, for two-dimensional or three-dimensional problems with nonlinear diffusion or nonlinear advection terms, it is too expensive to compute exponential matrices as that in the original IIF methods since it is not straightforward to apply the compact IIF methods to deal with high dimensionality associated with the nonlinear term  $\vec{F}$  (see Eq. (14)) in the scheme (17). The Krylov subspace approximation as that in the scheme (24) needs to be used.

**Example 3 (Fully nonlinear equations II).** We test the methods for solving fully nonlinear ADR equations with nonlinear lower order terms including nonlinear advection and reaction.

$$\begin{cases} u_t + \left(\frac{1}{2}u^2\right)_x + \left(\frac{1}{2}u^2\right)_y = \nabla \cdot (u\nabla u) - u^2 + f(x, y, t), & 0 \leq x, y \leq 2\pi, \\ u(x, y, 0) = 1 + 0.5 \sin(x + y), \end{cases} \quad (45)$$

**Table 6**

**Example 2**, the two-dimensional equation (44). CPU time, error, and order of accuracy of the Krylov IIF2 and IIF3 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.

Krylov IIF2 for nonlinear problems $\Delta t = 0.2h$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
10	2.73E-2	–	1.51E-2	–	0.41
20	1.57E-2	0.80	8.20E-3	0.88	1.15
40	5.80E-3	1.44	2.90E-3	1.50	5.42
80	1.70E-3	1.77	8.61E-4	1.75	21.20
160	4.51E-4	1.91	2.33E-4	1.89	128.51
320	1.18E-4	1.93	6.06E-5	1.94	1136.50
Krylov IIF3 for nonlinear problems $\Delta t = 0.2h$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
10	3.61E-2	–	1.36E-2	–	4.25
20	7.90E-3	2.19	3.20E-3	2.09	28.59
40	1.60E-3	2.30	6.09E-4	2.39	175.43
80	2.71E-4	2.56	9.82E-5	2.63	1102.31
160	4.21E-5	2.69	1.45E-5	2.76	8304.21
320	5.87E-6	2.84	1.97E-6	2.88	61444.01

**Table 7**

**Example 3**, the fully nonlinear equation (45). CPU time, error, and order of accuracy of the Krylov IIF2 and IIF3 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.

Krylov IIF2 for nonlinear problems $\Delta t = 0.2h$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
10	4.45E-2	–	2.53E-2	–	1.03
20	1.68E-2	1.41	7.07E-3	1.84	4.48
40	5.73E-3	1.55	2.43E-3	1.54	10.15
80	1.63E-3	1.81	7.15E-4	1.76	29.53
160	4.43E-4	1.88	1.96E-4	1.87	125.28
320	1.16E-4	1.93	5.06E-5	1.95	1047.63
Krylov IIF3 for nonlinear problems $\Delta t = 0.1h$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
10	5.53E-2	–	2.84E-2	–	23.26
20	5.77E-3	3.26	1.80E-3	3.98	85.78
40	4.43E-4	3.70	1.78E-4	3.34	232.77
80	5.45E-5	3.02	1.99E-5	3.16	2201.50
160	7.06E-6	2.95	2.46E-6	3.02	11052.31
320	9.22E-7	2.94	3.14E-7	2.97	126876.71

with periodic boundary condition.

$$f(x, y, t) = 1.125 - 0.625 \cos(2x + 2y - 2t) + 0.25 \sin(2x + 2y - 2t) + 0.5 \cos(x + y - t) + 2 \sin(x + y - t),$$

and the exact solution is  $u = 1 + 0.5 \sin(x + y - t)$ . For this problem in which all spatial terms are nonlinear, the Krylov IIF2-WENO and IIF3-WENO schemes perform similarly as in the previous examples, as shown in Table 7. Again, the large time step size  $\Delta t = O(\Delta x)$  is obtained.

**Remark.** Here we test the method in dealing with nonlinear diffusions, and the large time step size  $\Delta t = O(\Delta x)$  is obtained for a stable and accurate computation of fully nonlinear parabolic PDEs. We would like to emphasize the novelty of the IIF method developed in this paper and its potential to be applied to nonlinear diffusion models in application problems.

**Example 4** (A system with exact solution). We consider an advection–diffusion–reaction system on two- and three-dimensional domains  $\Omega = (0, 2\pi)^k \subset \mathbb{R}^k$  for  $k = 2, 3$ , where  $k$  denotes the spatial dimension. The system was used to test different IIF schemes in [33,34,44]. The two-dimensional system has the following form

$$\begin{cases} u_t + (a/2)(u_x + u_y) = (d/2)(u_{xx} + u_{yy}) - bu + v, \\ v_t + (a/2)(v_x + v_y) = (d/2)(v_{xx} + v_{yy}) - cv, \end{cases} \quad (46)$$

with periodic boundary conditions. For the initial condition

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

**Table 8**

Example 4, the 2D case (46). Comparison of CPU time, error, and order of accuracy of the Krylov IIF2 method with the RK2 method. Results by two different Krylov subspace dimension  $M = 25$  and  $M = 10$  are compared. Final time  $T = 1.0$ .  $N$  is the number of grid points in each spatial direction.

Krylov IIF2 $\Delta t = 0.5\Delta x$ , $M = 25$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	3.40E-1	–	1.90E-1	–	0.81
40	7.22E-2	2.24	4.52E-2	2.07	3.31
80	1.54E-2	2.23	1.08E-2	2.07	17.04
160	4.03E-3	1.93	2.68E-3	2.01	117.82
Krylov IIF2 $\Delta t = 0.5\Delta x$ , $M = 10$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	3.40E-1	–	1.90E-1	–	0.69
40	7.23E-2	2.23	4.52E-2	2.07	3.04
80	1.54E-2	2.23	1.07E-2	2.08	15.24
160	4.00E-3	1.94	2.70E-3	1.99	99.73
RK2 $\Delta t = 0.05\Delta x^2$					
$N$	$L^\infty$ error	$L^\infty$ 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	39789.26

**Table 9**

Example 4, the 2D case (46). Comparison of CPU time, error, and order of accuracy of the Krylov IIF3 method with the RK3 method. Results by two different Krylov subspace dimension  $M = 25$  and  $M = 10$  are compared. Final time  $T = 1.0$ .  $N$  is the number of grid points in each spatial direction.

Krylov IIF3 $\Delta t = 0.5\Delta x$ , $M = 25$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	1.40E-1	–	5.16E-2	–	1.40
40	3.03E-2	2.21	9.43E-3	2.45	4.77
80	4.25E-3	2.83	1.25E-3	2.92	24.84
160	4.08E-4	3.38	1.49E-4	3.07	167.62
Krylov IIF3 $\Delta t = 0.5\Delta x$ , $M = 10$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	1.40E-1	–	5.16E-2	–	0.75
40	3.03E-2	2.21	9.43E-3	2.45	3.43
80	4.21E-3	2.85	1.25E-3	2.92	15.87
160	4.08E-4	3.37	1.49E-4	3.07	108.55
RK3 $\Delta t = 0.05\Delta x^2$					
$N$	$L^\infty$ error	$L^\infty$ 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	59435.53

the system has the following exact solution

$$\begin{cases} u(x, y, t) = (e^{-(b+d)t} + e^{-(c+d)t}) \cos(x + y - at), \\ v(x, y, t) = (b - c)e^{-(c+d)t} \cos(x + y - at). \end{cases} \tag{47}$$

The three-dimensional system has the following form

$$\begin{cases} u_t + (a/3)(u_x + u_y + u_z) = (d/3)\Delta u - bu + v, \\ v_t + (a/3)(v_x + v_y + v_z) = (d/3)\Delta v - cv, \end{cases} \tag{48}$$

with periodic boundary conditions. For the initial condition

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

the exact solution of the system is

$$\begin{cases} u(x, y, z, t) = (e^{-(b+d)t} + e^{-(c+d)t}) \cos(x + y + z - at), \\ v(x, y, z, t) = (b - c)e^{-(c+d)t} \cos(x + y + z - at). \end{cases} \tag{49}$$

**Table 10**

Example 4, the 3D case (48). Comparison of CPU time, error, and order of accuracy of the Krylov IIF2 method with the RK2 method. Results by two different Krylov subspace dimension  $M = 25$  and  $M = 10$  are compared. Final time  $T = 1.0$ .  $N$  is the number of grid points in each spatial direction.

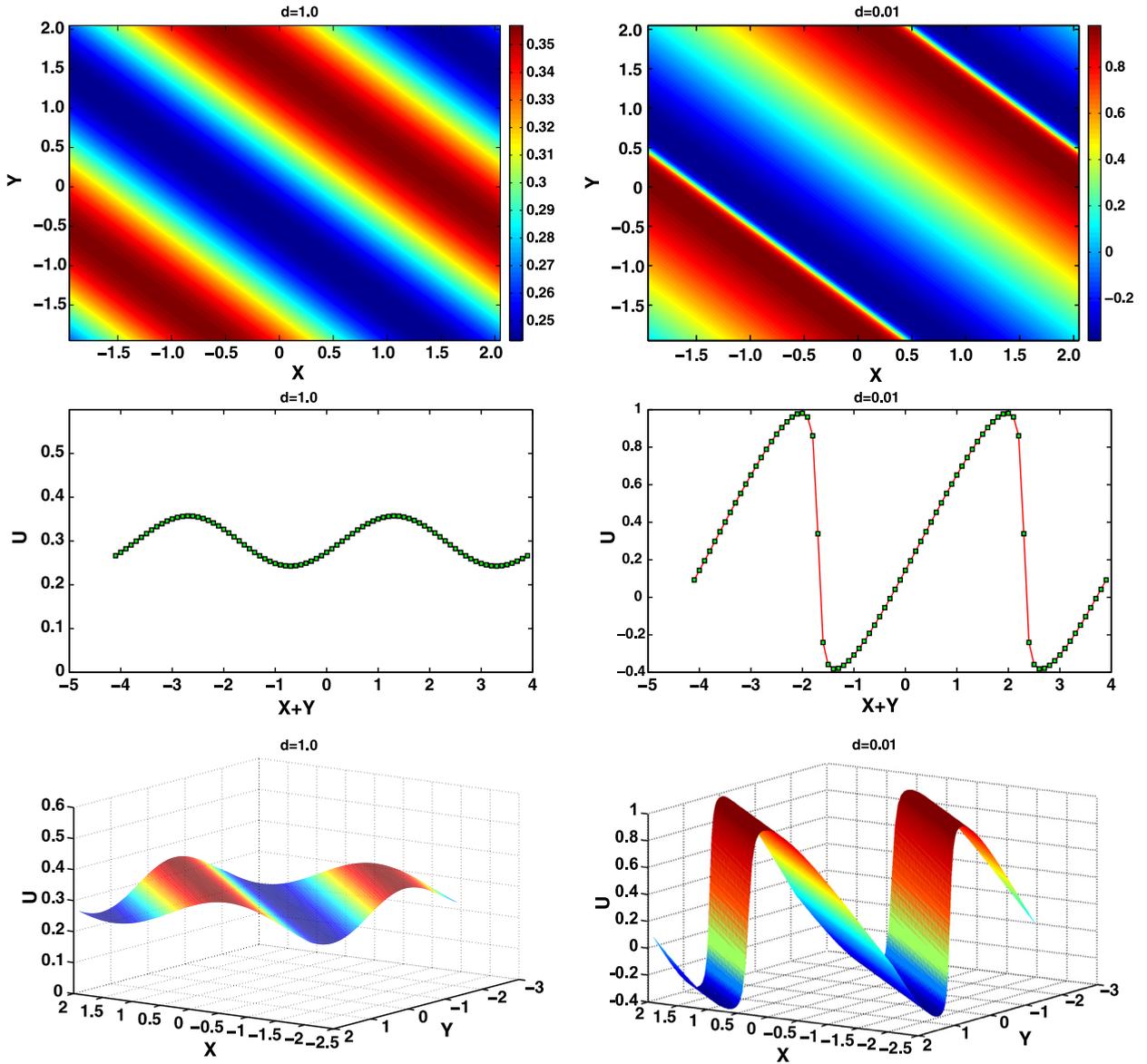
Krylov IIF2 $\Delta t = 0.5\Delta x$ , $M = 25$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	3.40E-1	–	1.90E-1	–	9.70
40	6.98E-2	2.28	4.48E-2	2.08	107.14
80	1.60E-2	2.13	1.08E-2	2.05	1670.04
160	4.16E-3	1.94	2.70E-3	2.00	28334.59
Krylov IIF2 $\Delta t = 0.5\Delta x$ , $M = 10$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	3.40E-1	–	1.90E-1	–	8.67
40	6.98E-2	2.28	4.48E-2	2.08	55.71
80	1.60E-2	2.13	1.08E-2	2.05	786.83
160	4.16E-3	1.94	2.70E-3	2.00	8914.14
RK2 $\Delta t = 0.05\Delta x^2$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	3.70E-1	–	1.50E-1	–	34.06
40	6.51E-2	2.51	3.05E-2	2.30	1092.90
80	1.38E-2	2.24	6.92E-3	2.14	35741.64
160	3.60E-3	1.94	1.89E-3	1.87	1109337.77

**Table 11**

Example 4, the 3D case (48). Comparison of CPU time, error, and order of accuracy of the Krylov IIF3 method with the RK3 method. Results by two different Krylov subspace dimension  $M = 25$  and  $M = 10$  are compared. Final time  $T = 1.0$ .  $N$  is the number of grid points in each spatial direction.

Krylov IIF3 $\Delta t = 0.5\Delta x$ , $M = 25$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	1.40E-1	–	5.15E-2	–	9.98
40	2.89E-2	2.28	9.07E-3	2.51	149.02
80	3.60E-3	3.01	1.10E-3	3.04	2368.60
160	3.18E-4	3.50	1.38E-4	2.99	42642.81
Krylov IIF3 $\Delta t = 0.5\Delta x$ , $M = 10$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	1.40E-1	–	5.15E-2	–	9.08
40	2.89E-2	2.28	9.07E-3	2.51	77.57
80	3.58E-3	3.01	1.10E-3	3.04	1036.74
160	3.19E-4	3.49	1.38E-4	2.99	16017.44
RK3 $\Delta t = 0.05\Delta x^2$					
$N$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
20	2.30E-1	–	1.00E-1	–	171.20
40	4.03E-2	2.51	1.54E-2	2.70	5527.97
80	4.41E-3	3.19	1.76E-3	3.13	172885.29
160	4.32E-4	3.35	2.06E-4	3.09	5411307.32

The parameters are chosen as  $a = c = d = 1$  and  $b = 100$  to give stiff reaction terms. The final time  $T = 1.0$  for both the 2D problem and the 3D problem. We compare the performance of Krylov IIF2 and Krylov IIF3 methods with a second order and a third order Runge–Kutta methods. Results by two different Krylov subspace dimension  $M = 25$  and  $M = 10$  are compared. See Tables 8, 9, 10, 11 for computation results. Designed second or third order accuracy is obtained for different second order or third order methods. In this example, we can see that Krylov IIF methods by using two different Krylov subspace dimensions  $M = 25$  and  $M = 10$  generate similar numerical errors. This fact shows that the numerical errors generated by Krylov subspace approximations are much smaller than those from the truncation errors of numerical schemes. And the computation is more efficient by using  $M = 10$ . Especially for the 3D problem, much less CPU time is needed in the simulation by using the smaller Krylov subspace. Again, we can see that Krylov IIF methods have a linear computational complexity, based on the facts that the CPU time approximately linearly depends on successively refined spatio-temporal meshes, and the time step size  $\Delta t = O(\Delta x)$ . For the regular Runge–Kutta methods,  $\Delta t = O(\Delta x)^2$  is needed for the stability of the computations. From the tables, we can observe that much less CPU time is needed by using Krylov IIF methods to reach a similar level numerical error. Hence Krylov IIF methods are much more efficient than the Runge–Kutta methods used in this example.



**Fig. 5.** Example 5, nonlinear viscous Burgers' equation. Simulation on a  $80 \times 80$  mesh by the Krylov IIF2-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.

**Example 5 (Nonlinear viscous Burgers' equation).** We consider the two-dimensional nonlinear viscous Burgers' equation

$$\begin{cases} u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = d\Delta u, & -2 \leq x \leq 2, -2 \leq y \leq 2, \\ u(x, y, 0) = 0.3 + 0.7 \sin\left(\frac{\pi}{2}(x + y)\right), \end{cases} \quad (50)$$

with periodic boundary condition.  $d$  is the viscous coefficient. The Krylov IIF2-WENO scheme is used to solve the PDE to  $T = 5/\pi^2$ . In this example, we test the performance of the scheme 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. 5. 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 new Krylov IIF time discretization technique.

**Table 12**

**Example 6**, Schnakenberg model. CPU time, error, and order of accuracy of Krylov IIF2-WENO and Krylov IIF3-WENO.  $M$  is the Krylov subspace dimension.  $T = 1.0$ .

Krylov IIF2 $M = 25$					
$\Delta t$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
5.00E-4	4.00E-1	–	4.25E-2	–	489.72
2.50E-4	1.20E-1	1.78	1.34E-2	1.67	984.18
1.25E-4	3.07E-2	1.93	3.70E-3	1.87	1877.40
6.25E-5	7.80E-3	1.97	9.66E-4	1.94	3694.10
Krylov IIF2 $M = 10$					
$\Delta t$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
5.00E-4	4.00E-1	–	4.25E-2	–	346.98
2.50E-4	1.20E-1	1.78	1.34E-2	1.67	705.81
1.25E-4	3.07E-2	1.93	3.70E-3	1.87	1398.62
6.25E-5	7.80E-3	1.97	9.66E-4	1.94	2647.90
Krylov IIF3 $M = 25$					
$\Delta t$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
5.00E-4	2.50E-1	–	2.68E-2	–	552.23
2.50E-4	4.95E-2	2.31	5.50E-3	2.28	1132.90
1.25E-4	7.80E-3	2.67	8.45E-4	2.70	2283.80
6.25E-5	1.10E-3	2.87	1.13E-4	2.90	4668.20
Krylov IIF3 $M = 10$					
$\Delta t$	$L^\infty$ error	$L^\infty$ order	$L^1$ error	$L^1$ order	CPU time (seconds)
5.00E-4	2.40E-1	–	2.68E-2	–	333.20
2.50E-4	4.95E-2	2.31	5.50E-3	2.28	669.93
1.25E-4	7.80E-3	2.67	8.45E-4	2.70	1436.80
6.25E-5	1.10E-3	2.87	1.13E-4	2.90	2837.40

**Example 6** (Schnakenberg model). The Schnakenberg system [37] 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* [13]. 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

$$\begin{cases} \frac{\partial C_a}{\partial t} + \frac{\partial C_a}{\partial x} + \frac{\partial C_a}{\partial y} = D_1 \nabla^2 C_a + \kappa (a - C_a + C_a^2 C_i), \\ \frac{\partial C_i}{\partial t} + \frac{\partial C_i}{\partial x} + \frac{\partial C_i}{\partial y} = D_2 \nabla^2 C_i + \kappa (b - C_a^2 C_i), \end{cases} \quad (51)$$

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

$$\begin{cases} C_a(x, y, 0) = a + b + 10^{-3} e^{-100((x-\frac{1}{3})^2 + (y-\frac{1}{2})^2)}, \\ C_i(x, y, 0) = \frac{b}{(a+b)^2}, \end{cases} \quad (52)$$

and the boundary conditions are taken as periodic boundary conditions. The parameters values are  $\kappa = 100$ ,  $a = 0.1305$ ,  $b = 0.7695$ ,  $D_1 = 0.05$ ,  $D_2 = 1$ . This is a stiff Turing system [46]. We simulate the system on the unit square domain  $\Omega = (0, 1)^2$ . To study the performance and convergence of the Krylov IIF-WENO methods for this system, we list in Table 12 the CPU time, error, and order of accuracy for simulations of the Schnakenberg model, on a fixed spatial resolution of  $32 \times 32$  mesh. The error at  $\Delta t$  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.,

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

The Krylov IIF2-WENO/IIF3-WENO clearly shows a second order/third order of accuracy in time as expected. Again in this example, we can see that Krylov IIF methods by using two different Krylov subspace dimensions  $M = 25$  and  $M = 10$  generate similar numerical errors. This fact shows that the numerical errors generated by Krylov subspace approximations are much smaller than those from the truncation errors of numerical schemes. The time evolution of the concentration of activator  $C_a$  is shown in Fig. 6. We can observe that the initial perturbation in (52) is amplified and spreads, leading to a formation of spot-like patterns.

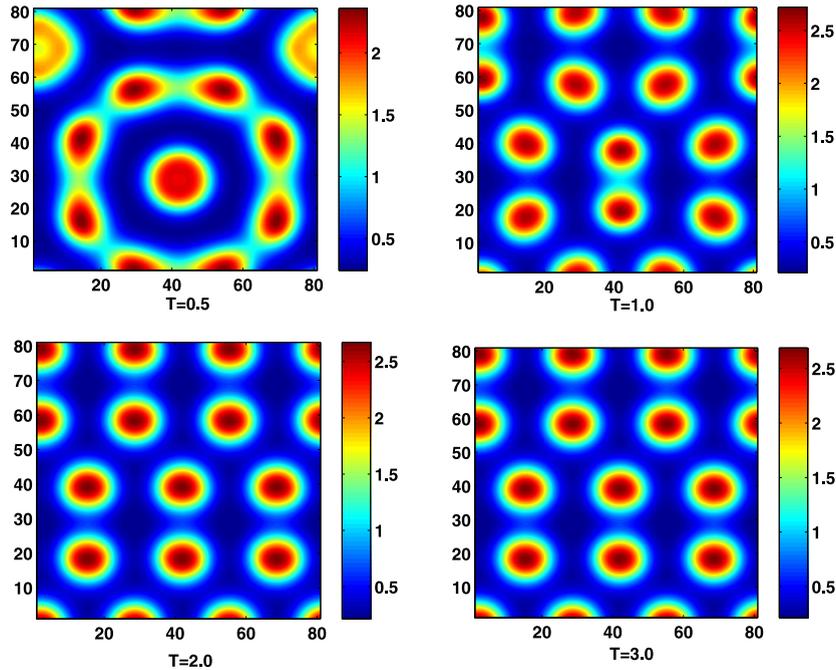


Fig. 6. Numerical solution of the Schnakenberg system by the Krylov IIF2-WENO method on a  $80 \times 80$  mesh. Contour plots of time evolution of the concentration of the activator  $C_a$ .

## 5. Conclusions

Implicit integration factor (IIF) methods [33,34,26,44] are a class of efficient time discretization methods for solving stiff reaction–diffusion systems. In this paper, we designed a new Krylov IIF technique for semilinear and fully nonlinear advection–diffusion–reaction (ADR) systems. This new Krylov IIF approach is coupled with WENO method for the advection part of the system. This new Krylov IIF approach extends previous Krylov IIF methods [7] in the following two aspects: (1) the new method can be designed for arbitrary order of accuracy for solving ADR systems; (2) the new method can solve fully nonlinear PDE system, i.e., the highest order term in the PDE is nonlinear. Via numerical experiments and linear stability analysis, we verified that the new method has large stability region, and is efficient and accurate for simulating nonlinear ADR systems.

## References

- [1] M. Alber, N. Chen, P. Lushnikov, S. Newman, Continuous macroscopic limit of a discrete stochastic model for interaction of living cells, *Phys. Rev. Lett.* 99 (2007) 168102.
- [2] U. Ascher, S. Ruuth, B. Wetton, Implicit-explicit methods for time-dependent PDE's, *SIAM J. Numer. Anal.* 32 (1995) 797–823.
- [3] G. Beylkin, J.M. Keiser, L. Vozvoi, 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] R.L. Bowers, J.R. Wilson, *Numerical Modeling in Applied Physics and Astrophysics*, Jones and Bartlett Publishers, 1991.
- [6] P. Carmeliet, Mechanisms of angiogenesis and arteriogenesis, *Nature Medicine* 6 (2000) 389–395.
- [7] 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.
- [8] C.-S. Chou, Y.-T. Zhang, R. Zhao, Q. Nie, Numerical methods for stiff reaction–diffusion systems, *Discrete Contin. Dyn. Syst. Ser. B* 7 (2007) 515–525.
- [9] A. Christlieb, B. Ong, J.-M. Qiu, Integral deferred correction methods constructed with high order Runge–Kutta integrators, *Math. Comput.* 79 (2010) 761–783.
- [10] S.M. Cox, P.C. Matthews, Exponential time differencing for stiff systems, *J. Comput. Phys.* 176 (2002) 430–455.
- [11] A. Dutt, L. Greengard, V. Rokhlin, Spectral deferred correction methods for ordinary differential equations, *BIT* 40 (2) (2000) 241–266.
- [12] E. Gallopoulos, Y. Saad, Efficient solution of parabolic equations by Krylov approximation methods, *SIAM J. Sci. Stat. Comput.* 13 (5) (1992) 1236–1264.
- [13] 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.
- [14] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge–Kutta schemes, *Math. Comput.* 67 (1998) 73–85.
- [15] S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability preserving high order time discretization methods, *SIAM Rev.* 43 (2001) 89–112.
- [16] M.A. Herrero, J.J.L. Velazquez, Singularity patterns in a chemotaxis model, *Math. Ann.* 306 (1996) 583–623.
- [17] M.P. Brenner, P. Constantin, L.P. Kadanoff, A. Shenkel, S.C. Venkataramani, Diffusion, attraction and collapse, *Nonlinearity* 12 (1999) 1071–1098.
- [18] J. Huang, J. Jia, M. Minion, Arbitrary order Krylov deferred correction methods for differential algebraic equations, *J. Comput. Phys.* 221 (2) (2007) 739–760.
- [19] W. Hundsdorfer, J. Verwer, *Numerical Solution of Time-Dependent Advection–Diffusion–Reaction Equations*, Springer-Verlag, 2003.
- [20] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1996) 202–228.

- [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 (2) (2007) 1753–1781.
- [22] A.-K. Kassam, L.N. Trefethen, Fourth-order time stepping for stiff PDEs, *SIAM J. Sci. Comput.* 26 (4) (2005) 1214–1233.
- [23] E.F. Keller, L.A. Segel, Traveling bands of chemotactic bacteria: A theoretical analysis, *J. Theor. Biol.* 30 (1971) 235–248.
- [24] C.A. Kennedy, M.H. Carpenter, Additive Runge–Kutta schemes for convection–diffusion–reaction equations, *Appl. Numer. Math.* 44 (2003) 139–181.
- [25] A.T. Layton, M.L. Minion, Conservative multi-implicit spectral deferred correction methods for reacting gas dynamics, *J. Comput. Phys.* 194 (2) (2004) 697–715.
- [26] X.F. Liu, Q. Nie, Compact integration factor methods for complex domains and adaptive mesh refinement, *J. Comput. Phys.* 229 (16) (2010) 5692–5706.
- [27] P. Lushnikov, N. Chen, M.S. Alber, Macroscopic dynamics of biological cells interacting via chemotaxis and direct contact, *Phys. Rev. E* 78 (2008) 061904.
- [28] 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.
- [29] D. Mihalas, B.W. Mihalas, *Foundation of Radiation Hydrodynamics*, Oxford University press, 1984.
- [30] M.L. Minion, Semi-implicit spectral deferred correction methods for ordinary differential equations, *Commun. Math. Sci.* 1 (3) (2003) 471–500.
- [31] 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.
- [32] V.A. Mousseau, D.A. Knoll, W.J. Rider, Physical-based preconditioning and the Newton–Krylov method for non-equilibrium radiation diffusion, *J. Comput. Phys.* 160 (2000) 743–765.
- [33] Q. Nie, Y.-T. Zhang, R. Zhao, Efficient semi-implicit schemes for stiff systems, *J. Comput. Phys.* 214 (2006) 521–537.
- [34] 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.
- [35] J. Rider, A. Knoll, L. Olson, A multigrid Newton–Krylov method for multimaterial equilibrium radiation diffusion, *J. Comput. Phys.* 152 (1999) 164–191.
- [36] D.L. Ropp, J.N. Shadid, Stability of operator splitting methods for systems with indefinite operators: Advection–diffusion–reaction systems, *J. Comput. Phys.* 228 (2009) 3508–3516.
- [37] J. Schnakenberg, Simple chemical reaction systems with limit cycle behavior, *J. Theor. Biol.* 81 (1979) 389–400.
- [38] C.-W. Shu, TVD time discretizations, *SIAM J. Sci. Stat. Comput.* 9 (1988) 1073–1084.
- [39] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.* 77 (1988) 439–471.
- [40] 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.
- [41] B. Sportisse, An analysis of operator splitting techniques in the stiff case, *J. Comput. Phys.* 161 (2000) 140–168.
- [42] L.N. Trefethen, D. Bau, *Numerical Linear Algebra*, SIAM, 1997.
- [43] J.G. Verwer, B.P. Sommeijer, W. Hundsdorfer, RK4 time-stepping for advection–diffusion–reaction problems, *J. Comput. Phys.* 201 (2004) 61–79.
- [44] S. Zhao, J. Ovardia, 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.
- [45] X. Zhong, Additive semi-implicit Runge–Kutta methods for computing high-speed nonequilibrium reactive flows, *J. Comput. Phys.* 128 (1996) 19–31.
- [46] 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.