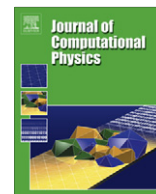




ELSEVIER

Contents lists available at ScienceDirect

## Journal of Computational Physics

journal homepage: [www.elsevier.com/locate/jcp](http://www.elsevier.com/locate/jcp)

# Operator splitting implicit integration factor methods for stiff reaction–diffusion–advection systems

Su Zhao<sup>a</sup>, Jeremy Ovadia<sup>a</sup>, Xinfeng Liu<sup>b</sup>, Yong-Tao Zhang<sup>c</sup>, Qing Nie<sup>a,\*</sup>

<sup>a</sup> Department of Mathematics, University of California, Irvine, CA 92697, USA

<sup>b</sup> Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA

<sup>c</sup> Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556, USA

## ARTICLE INFO

## Article history:

Received 24 November 2010

Received in revised form 5 April 2011

Accepted 6 April 2011

Available online xxxx

## Keywords:

Reaction–diffusion–advection equations

Operator splitting

Implicit integration factor methods

Compact implicit integration factor methods

Weighted essentially non-oscillatory methods

Weighted essentially non-oscillatory methods

## ABSTRACT

For reaction–diffusion–advection equations, the stiffness from the reaction and diffusion terms often requires very restricted time step size, while the nonlinear advection term may lead to a sharp gradient in localized spatial regions. It is challenging to design numerical methods that can efficiently handle both difficulties. For reaction–diffusion systems with both stiff reaction and diffusion terms, implicit integration factor (IIF) method and its higher dimensional analog compact IIF (cIIF) serve as an efficient class of time-stepping methods, and their second order version is linearly unconditionally stable. For nonlinear hyperbolic equations, weighted essentially non-oscillatory (WENO) methods are a class of schemes with a uniformly high order of accuracy in smooth regions of the solution, which can also resolve the sharp gradient in an accurate and essentially non-oscillatory fashion. In this paper, we couple IIF/cIIF with WENO methods using the operator splitting approach to solve reaction–diffusion–advection equations. In particular, we apply the IIF/cIIF method to the stiff reaction and diffusion terms and the WENO method to the advection term in two different splitting sequences. Calculation of local truncation error and direct numerical simulations for both splitting approaches show the second order accuracy of the splitting method, and linear stability analysis and direct comparison with other approaches reveals excellent efficiency and stability properties. Applications of the splitting approach to two biological systems demonstrate that the overall method is accurate and efficient, and the splitting sequence consisting of two reaction–diffusion steps is more desirable than the one consisting of two advection steps, because CWC exhibits better accuracy and stability.

© 2011 Elsevier Inc. All rights reserved.

## 1. Introduction

Consider a system of reaction–diffusion–advection (RDA) equations,

$$\frac{\partial \mathbf{u}}{\partial t} = \nabla \cdot \mathbf{f}(\mathbf{u}) + \bar{\mathbf{D}}\Delta \mathbf{u} + \mathbf{F}_R(\mathbf{u}), \quad (1)$$

where  $\mathbf{u} \in \mathbf{R}^m$  represents a group of physical or biological species,  $\mathbf{f}(\mathbf{u})$  is the flux function and  $\nabla \cdot \mathbf{f}(\mathbf{u})$  represents the advection,  $\bar{\mathbf{D}} \in \mathbf{R}^{m \times m}$  is the diffusion constant matrix,  $\Delta \mathbf{u}$  is the Laplacian associated with the diffusion of the species  $\mathbf{u}$ , and  $\mathbf{F}_R(\mathbf{u})$  represents reactions. Many physical and biological systems may be described by reaction–diffusion–advection equations

\* Corresponding author. Tel.: +1 949 824 5530; fax: +1 949 824 7993.

E-mail address: [qnie@math.uci.edu](mailto:qnie@math.uci.edu) (Q. Nie).

[1,2]. One example is that early shaping of the vertebrate limb bud is due to a diffusion–reaction–advection process between a growth factor and the morphogen Sonic Hedgehog produced in the zone of polarizing activity at the posterior margin of the bud [3].

Each of the three parts of Eq. (1), diffusion, reaction, and advection, exhibits different temporal and spatial characteristics. For example, the reactions may consist of many rate constants varying greatly in magnitude that impose strong stability constraints on the size of the time step. To overcome this, one may use implicit temporal schemes for the entire system that require solving nonlinear algebraic systems at each time step. Due to spatial discretization, the reaction species and the spatial variables are usually coupled, leading to a large nonlinear system which is difficult to solve, especially in two and three spatial dimensions. In addition, implicit temporal schemes are usually not suitable for treating the advection term. Naturally, it is more desirable to treat the distinct parts of the reaction–diffusion–advection system with different methods.

Operator splitting methods, which were originally developed as a technique for separating multi-dimensional spatial operators into a sum of one-dimensional operators, have been used to treat distinct terms of nonlinear partial differential equations in terms of different discretization techniques [4,5]. By applying operator splitting techniques to reaction–diffusion–advection equations, the temporal discretization can be divided into several substeps at each time step. Then at each substep, only part of the reaction–diffusion–advection equation is updated in time. This property of operator splitting techniques enables the flexible usage of different temporal discretizations on distinct parts of the partial differential equations.

In this paper, we present a family of operator splitting methods for reaction–diffusion–advection equations with stiff reactions by splitting the temporal discretization into two major steps: the reaction–diffusion step and the advection step. Using this splitting arrangement, we apply existing temporal and spatial discretizations designed for diffusion–reaction equations to the reaction–diffusion step and existing hyperbolic solvers to the advection term. In particular, for the reaction–diffusion step, we utilize a class of semi-implicit integration factor (IIF) methods [6,7] recently developed for solving stiff reaction–diffusion equations, and for the advection step, we apply the WENO schemes [8].

In integration factor (IF) or exponential time differencing (ETD) methods [9–14], the diffusion term (or the linear differential operator with the highest order derivatives) is evaluated exactly (see [15] for review). To deal with stiff reactions in reaction–diffusion equations, one may discretize the reaction terms implicitly while maintaining the exact treatment of the diffusion term, leading to a class of semi-implicit integration factor (IIF) methods [6]. For a system in high (two or three) spatial dimensions, a compact implicit integration factor method (cIIF) can be implemented to save the storage cost and to improve efficiency [7]. Both IIF and cIIF schemes have excellent stability properties that allow large time steps without sacrificing computational cost significantly at each time step. This result is partly due to the decoupling of the implicit treatment of the nonlinear reaction terms and the exact evaluations of the linear diffusion term. In this paper, we use the second order IIF scheme [6], which is linearly unconditionally stable, for the reaction–diffusion part of the splitting method in a one-dimensional system and its counterpart in high dimensions, cIIF method [7], for any two- or three-dimensional systems.

To treat the advection term at each substep of the splitting method, we use weighted essentially non-oscillatory (WENO) schemes [16,17] with a third order TVD Runge–Kutta time discretization [18]. WENO, a class of high order numerical methods designed for solving hyperbolic PDEs whose solutions may contain discontinuities and sharp gradients, is one of the most used high resolution schemes [19–21] due to its high accuracy and efficiency (see [8] for review).

The rest of this paper is organized as follows: in Section 2, we present two different splitting approaches and analyze their truncation errors and stability regions; in Section 3, we compare these two splitting approaches as well as different temporal schemes for the reaction–diffusion part in the splitting sequences using a system of linear reaction–diffusion–advection equations with known solutions; in Section 4, we apply the two operator splitting approaches to two biological systems; and in Section 5, we conclude the paper.

## 2. Operator-splitting method combined with WENO and cIIF methods

### 2.1. Two splitting schemes

#### 2.1.1. Diffusion–advection–diffusion (DAD) splitting

We first apply the Strang operator splitting [22] method to our reaction–diffusion–advection system in Eq. (1). With this approach, the advection term is computed once and the rest of terms consisting of both diffusion and reaction are computed twice at each time step:

$$\begin{cases} \frac{\partial \mathbf{u}^*}{\partial t} = \bar{D}\Delta \mathbf{u}^* + \mathbf{F}_R(\mathbf{u}^*), & \text{on } (t_n, t_{n+1/2}), & \mathbf{u}^*(t_n) = \mathbf{u}(t_n), \\ \frac{\partial \mathbf{u}^{**}}{\partial t} = \nabla \cdot \mathbf{f}(\mathbf{u}^{**}), & \text{on } (t_n, t_{n+1}), & \mathbf{u}^{**}(t_n) = \mathbf{u}^*(t_{n+1/2}), \\ \frac{\partial \mathbf{u}^{***}}{\partial t} = \bar{D}\Delta \mathbf{u}^{***} + \mathbf{F}_R(\mathbf{u}^{***}), & \text{on } (t_{n+1/2}, t_{n+1}), & \mathbf{u}^{***}(t_{n+1/2}) = \mathbf{u}^{**}(t_{n+1}), \end{cases} \quad (2)$$

with  $\mathbf{u}(t_{n+1}) = \mathbf{u}^{***}(t_{n+1})$ .

In this scheme, referred to as DAD (diffusion–advection–diffusion) splitting, updating the reaction–diffusion–advection system at one time step is reduced to updating two reaction–diffusion systems, the first and third equations of Eq. (2), and one hyperbolic equation, the second equation. With such a simple structure, one may use different temporal schemes that are suitable for each corresponding equation. To handle the stiffness arising from the reaction and diffusion terms, we

use semi-implicit integration factor (IIF) or compact semi-implicit integration factor (cIIF) method [6,7] that has excellent stability properties. To handle the hyperbolic system, we use the third order finite difference WENO scheme [16,17] with a third order TVD Runge–Kutta time discretization [18,23]. These particular choices of temporal schemes for splitting are termed in this paper as CWC, representing the cIIF–WENO–cIIF sequence of computation.

In the CWC scheme, we first choose  $\Delta t = t_{n+1} - t_n$  based on the stability of the reaction–diffusion parts in Eq. (2). When the second order IIF [6] is applied to the reaction–diffusion equations of the system of Eq. (2), we obtain

$$\mathbf{u}_{n+1/2}^* = e^{D\Delta t/2} \left( \mathbf{u}_n^* + \frac{\Delta t}{4} F_R(\mathbf{u}_n^*) \right) + \frac{\Delta t}{4} F_R(\mathbf{u}_{n+1/2}^*), \quad (3)$$

where  $D$  is the matrix due to the second order discretization of the diffusion term in space.

When the third order finite difference WENO scheme [16–18] with a third order TVD Runge–Kutta time discretization is applied to the advection equation from  $t_n$  to  $t_{n+1}$ , the variable time step of the WENO/TVD solver, denoted by  $\Delta t_a$ , is determined by the stability condition. Assume that the WENO/TVD method requires  $\Delta t_w$  for the advection equation of Eq. (2) at  $\mathbf{u}^{**}(t_n)$ . We choose  $\Delta t_a = \Delta t/2$  if  $\Delta t/2 < \Delta t_w$  or  $\Delta t_a = \Delta t/(2m)$  if  $\Delta t_w < \Delta t/2$ , where  $m$  is the smallest integer such that  $\Delta t/(2m) < \Delta t_w$ . This choice of time steps keeps the stability and accuracy of WENO/TVD unchanged in addition to allowing an equal time step for IIF (or cIIF). This lack of constraint on the time step of the reaction–diffusion part of the system allows the exponential matrices in Eq. (3) to be computed only once, leading to an efficient implementation and computation of IIF and cIIF, similar to [6,7].

### 2.1.2. Advection–diffusion–advection (ADA) splitting

In a splitting approach, one can also treat the advection equation twice at half time steps while treating the reaction–diffusion equation only once during each time step. This advection–diffusion–advection (ADA) arrangement may be written in the following form:

$$\begin{cases} \frac{\partial \mathbf{u}^*}{\partial t} = \nabla \cdot f(\mathbf{u}^*), & \text{on } (t_n, t_{n+1/2}), \quad \mathbf{u}^*(t_n) = \mathbf{u}(t_n), \\ \frac{\partial \mathbf{u}^{**}}{\partial t} = \bar{D}\Delta \mathbf{u}^{**} + F_R(\mathbf{u}^{**}), & \text{on } (t_n, t_{n+1}), \quad \mathbf{u}^{**}(t_n) = \mathbf{u}^*(t_{n+1/2}), \\ \frac{\partial \mathbf{u}^{***}}{\partial t} = \nabla \cdot f(\mathbf{u}^{***}), & \text{on } (t_{n+1/2}, t_{n+1}), \quad \mathbf{u}^{***}(t_{n+1/2}) = \mathbf{u}^{**}(t_{n+1}), \end{cases} \quad (4)$$

with  $\mathbf{u}(t_{n+1}) = \mathbf{u}^{***}(t_{n+1})$ .

Similar to the DAD approach, we update the reaction–diffusion equation using the second order IIF (or cIIF) [7,6] and compute the hyperbolic step using the third order TVD Runge–Kutta method in combination with the third order WENO method for the spatial discretization [16,17]. This method is referred as WENO–cIIF–WENO (WCW) method.

## 2.2. Truncation error

We now analyze the overall local temporal truncation error for the CWC and WCW schemes. It is clear that the overall spatial discretization is of order two because a third order WENO method is applied to the advection term and a second order central difference approximation is applied to the diffusion term in IIF (or cIIF). Similar to the previous approaches [4–7], we analyze the following semi-discretization form for the temporal error:

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{A}\mathbf{u} + \mathbf{D}\mathbf{u} + \mathbf{R}\mathbf{u} \quad \mathbf{A}, \mathbf{D}, \mathbf{R} \in \mathbf{R}^{m \times m}. \quad (5)$$

### 2.2.1. CWC scheme

First, we apply DAD splitting to Eq. (5) to obtain:

$$\begin{cases} \frac{\partial \mathbf{u}^*}{\partial t} = \mathbf{D}\mathbf{u}^* + \mathbf{R}\mathbf{u}^* & \text{on } (t_n, t_{n+1/2}), \quad \mathbf{u}^*(t_n) = \mathbf{u}(t_n), \\ \frac{\partial \mathbf{u}^{**}}{\partial t} = \mathbf{A}\mathbf{u}^{**} & \text{on } (t_n, t_{n+1}), \quad \mathbf{u}^{**}(t_n) = \mathbf{u}^*(t_{n+1/2}), \\ \frac{\partial \mathbf{u}^{***}}{\partial t} = \mathbf{D}\mathbf{u}^{***} + \mathbf{R}\mathbf{u}^{***} & \text{on } (t_{n+1/2}, t_{n+1}), \quad \mathbf{u}^{***}(t_{n+1/2}) = \mathbf{u}^{**}(t_{n+1}), \end{cases} \quad (6)$$

with  $\mathbf{u}(t_{n+1}) = \mathbf{u}^{***}(t_{n+1})$ . The direct usage of the second order scheme of IIF [6] and the third order Runge–Kutta method [23] results in

$$\begin{cases} \mathbf{u}_{n+1/2}^* = \left( I - \frac{R\Delta t/2}{2} \right)^{-1} e^{D\Delta t/2} \left( I + \frac{R\Delta t/2}{2} \right) \mathbf{u}_n^*, \\ \mathbf{u}_{n+1}^{**} = \left( I + A\Delta t + \frac{A^2}{2} \Delta t^2 + \frac{A^3}{6} \Delta t^3 \right) \mathbf{u}_n^{**}, \\ \mathbf{u}_{n+1}^{***} = \left( I - \frac{R\Delta t/2}{2} \right)^{-1} e^{D\Delta t/2} \left( I + \frac{R\Delta t/2}{2} \right) \mathbf{u}_{n+1/2}^{***}. \end{cases} \quad (7)$$

Simplification of the Eq. (7) gives an explicit relation between the solutions at the old time step and the new time step:

$$\mathbf{u}_{n+1} = \left(I - \frac{R\Delta t}{2}\right)^{-1} e^{D\Delta t/2} \left(I + \frac{R\Delta t}{2}\right) \left(I + A\Delta t + \frac{A^2}{2}\Delta t^2 + \frac{A^3}{6}\Delta t^3\right) \left(I - \frac{R\Delta t}{2}\right)^{-1} e^{D\Delta t/2} \left(I + \frac{R\Delta t}{2}\right) \mathbf{u}_n. \quad (8)$$

By Taylor expansion, Eq. (8) becomes

$$\begin{aligned} \mathbf{u}_{n+1} &= \left(I + (D+R)\frac{\Delta t}{2} + \frac{(D+R)^2}{2}\left(\frac{\Delta t}{2}\right)^2 + \dots\right) \left(I + A\Delta t + \frac{A^2}{2}\Delta t^2 + \dots\right) \left(I + (D+R)\frac{\Delta t}{2} + \frac{(D+R)^2}{2}\left(\frac{\Delta t}{2}\right)^2 + \dots\right) \mathbf{u}_n \\ &= \left(I + (A+D+R)\Delta t + \frac{(A+D+R)^2}{2}\Delta t^2 + \dots\right) \mathbf{u}_n. \end{aligned} \quad (9)$$

Recall the exact solution of Eq. (5),

$$\mathbf{u}(t_{n+1}) = e^{(A+D+R)\Delta t} \mathbf{u}(t_n).$$

From here, the local truncation error of the CWC splitting method may be expressed as,

$$\left(I + (A+D+R)\Delta t + \frac{(A+D+R)^2}{2}\Delta t^2 + \dots\right) \mathbf{u}_n - e^{(A+D+R)\Delta t} \mathbf{u}_n = O(\Delta t^3) \mathbf{u}_n. \quad (10)$$

Consequently, the CWC splitting method using second order IIF (or cIIF) is of second order in temporal discretization.

### 2.2.2. WCW scheme

Similarly, we apply the ADA splitting scheme to Eq. (5) to obtain:

$$\begin{cases} \frac{\partial \mathbf{u}^*}{\partial t} = A\mathbf{u}^* & \text{on } (t_n, t_{n+1/2}), & \mathbf{u}^*(t_n) = \mathbf{u}(t_n), \\ \frac{\partial \mathbf{u}^{**}}{\partial t} = D\mathbf{u}^{**} + R\mathbf{u}^{**} & \text{on } (t_n, t_{n+1}), & \mathbf{u}^{**}(t_n) = \mathbf{u}^*(t_{n+1/2}), \\ \frac{\partial \mathbf{u}^{***}}{\partial t} = A\mathbf{u}^{***} & \text{on } (t_{n+1/2}, t_{n+1}), & \mathbf{u}^{***}(t_{n+1/2}) = \mathbf{u}^{**}(t_{n+1}), \end{cases} \quad (11)$$

with  $\mathbf{u}(t_{n+1}) = \mathbf{u}^{***}(t_{n+1})$ . Following a derivation similar to that of Eq. (7), we reach solutions at each step in the form:

$$\begin{cases} \mathbf{u}_{n+1/2}^* = \left(I + A\frac{\Delta t}{2} + \frac{A^2}{2}\left(\frac{\Delta t}{2}\right)^2 + \frac{A^3}{6}\left(\frac{\Delta t}{2}\right)^3\right) \mathbf{u}_n^*, \\ \mathbf{u}_{n+1}^{**} = \left(I - \frac{R\Delta t}{2}\right)^{-1} e^{D\Delta t} \left(I + \frac{R\Delta t}{2}\right) \mathbf{u}_n^{**}, \\ \mathbf{u}_{n+1}^{***} = \left(I + A\frac{\Delta t}{2} + \frac{A^2}{2}\left(\frac{\Delta t}{2}\right)^2 + \frac{A^3}{6}\left(\frac{\Delta t}{2}\right)^3\right) \mathbf{u}_{n+1/2}^{***}. \end{cases} \quad (12)$$

Then, the solution becomes

$$\mathbf{u}_{n+1} = \left(I + A\frac{\Delta t}{2} + \frac{A^2}{2}\left(\frac{\Delta t}{2}\right)^2 + \frac{A^3}{6}\left(\frac{\Delta t}{2}\right)^3\right) \left(I - \frac{R\Delta t}{2}\right)^{-1} e^{D\Delta t} \left(I + \frac{R\Delta t}{2}\right) \left(I + A\frac{\Delta t}{2} + \frac{A^2}{2}\left(\frac{\Delta t}{2}\right)^2 + \frac{A^3}{6}\left(\frac{\Delta t}{2}\right)^3\right) \mathbf{u}_n. \quad (13)$$

By Taylor expansion, the solution in Eq. (13) becomes

$$\begin{aligned} \mathbf{u}_{n+1} &= \left(I + A\frac{\Delta t}{2} + \frac{A^2}{2}\left(\frac{\Delta t}{2}\right)^2 + \dots\right) \left(I + (D+R)\Delta t + \frac{(D+R)^2}{2}\Delta t^2 + \dots\right) \left(I + A\frac{\Delta t}{2} + \frac{A^2}{2}\left(\frac{\Delta t}{2}\right)^2 + \dots\right) \mathbf{u}_n \\ &= \left(I + (A+D+R)\Delta t + \frac{(A+D+R)^2}{2}\Delta t^2 + \dots\right) \mathbf{u}_n. \end{aligned} \quad (14)$$

Recall the exact solution of Eq. (5),

$$\mathbf{u}(t_{n+1}) = e^{(A+D+R)\Delta t} \mathbf{u}(t_n).$$

Then, the local truncation error of the WCW splitting method is

$$\left(I + (A+D+R)\Delta t + \frac{(A+D+R)^2}{2}\Delta t^2 + \dots\right) \mathbf{u}_n - e^{(A+D+R)\Delta t} \mathbf{u}(t_n) = O(\Delta t^3) \mathbf{u}_n. \quad (15)$$

Thus, the WCW splitting method is of second order in the time discretization.

### 2.3. Linear stability analysis

To analyze the linear stability of the splitting method, we consider a scalar linear equation,

$$u_t = au - du + ru \quad \text{with } a, r \in \mathbf{R}, d > 0, \quad (16)$$

similar to the approaches in [4–6].

2.3.1. CWC scheme

After applying DAD splitting to Eq. (16), we obtain:

$$\begin{cases} u_t^* = -du^* + ru^*, & \text{on } (t_n, t_{n+1/2}), & u^*(t_n) = u(t_n), \\ u_t^{**} = au^{**}, & \text{on } (t_n, t_{n+1}), & u^{**}(t_n) = u^*(t_{n+1/2}), \\ u_t^{***} = -du^{***} + ru^{***}, & \text{on } (t_{n+1/2}, t_{n+1}), & u^{***}(t_{n+1/2}) = u^{**}(t_{n+1}). \end{cases} \quad (17)$$

Similar to the derivation of Eq. (8), we obtain

$$u_{n+1} = \left(I - \frac{r\Delta t/2}{2}\right)^{-1} e^{-d\Delta t/2} \left(I + \frac{r\Delta t/2}{2}\right) \left(I + a\Delta t + \frac{a^2}{2}\Delta t^2 + \frac{a^3}{6}\Delta t^3\right) \left(I - \frac{r\Delta t/2}{2}\right)^{-1} e^{-d\Delta t/2} \left(I + \frac{r\Delta t/2}{2}\right) u_n. \quad (18)$$

Substituting  $u_n = e^{in\theta}$  into Eq. (18) yields,

$$e^{i\theta} = e^{-d\Delta t} \left[ 1 + a\Delta t + \frac{1}{2}(a\Delta t)^2 + \frac{1}{6}(a\Delta t)^3 \right] \left( \frac{4 + \lambda}{4 - \lambda} \right)^2, \quad (19)$$

where  $\lambda = r\Delta t$ . Denote  $\lambda_r$  as the real part of  $\lambda$  and  $\lambda_i$  as its imaginary part of  $\lambda$ . Similar to the stability analysis for IIF (or CIIF) methods [6,7], we present boundaries of the stability region: a family of curves for different values of  $a\Delta t$  at different values of  $d\Delta t$ .

Based on the amplification factor derived in Eq. (19), the stability regions always include the point  $(-4, 0)$  in the complex plane for any values of  $a\Delta t$  and  $d\Delta t$ . We plot the boundaries of stability regions on the complex plane of the reaction term  $r\Delta t$ , for different values of  $a\Delta t$  and  $d\Delta t$ . As seen in Fig. 1, the closed curves formed by the boundary of the stability region of the DAD splitting method do not stay in the same half plane when the value of  $a\Delta t$  increases from  $-\infty$  to  $+\infty$ . For a fixed  $d\Delta t$  value, the stability regions shrink and converge to the point  $(-4, 0)$  as  $|a\Delta t| \rightarrow \infty$ . Hence, stronger advections induce more severe stability constraints. On the other hand, for the same  $a\Delta t$  value, larger  $d\Delta t$  values allow larger stability regions due to the broad stability of integration factor methods. Since the integration factor methods treat the diffusion exactly, the overall method becomes more advantageous for larger diffusions.

2.3.2. WCW scheme

For the stability of the WCW scheme, applying ADA splitting to Eq. (16) yields

$$\begin{cases} u_t^* = au^*, & \text{on } (t_n, t_{n+1/2}), & u^*(t_n) = u(t_n), \\ u_t^{**} = -du^{**} + ru^{**}, & \text{on } (t_n, t_{n+1}), & u^{**}(t_n) = u^*(t_{n+1/2}), \\ u_t^{***} = au^{***}, & \text{on } (t_{n+1/2}, t_{n+1}), & u^{***}(t_{n+1/2}) = u^{**}(t_{n+1}). \end{cases} \quad (20)$$

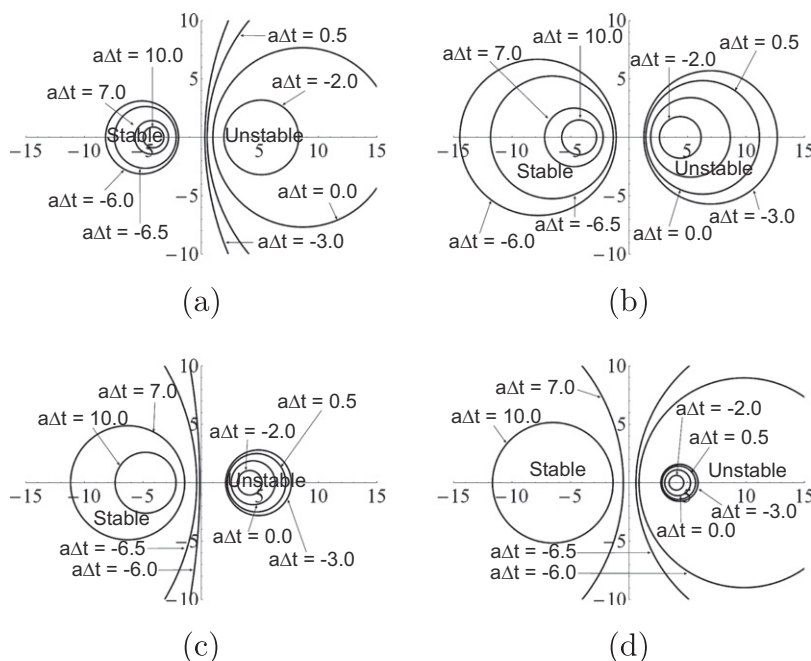


Fig. 1. Stability regions for the second order CWC scheme with different  $a\Delta t$ . (a)  $d\Delta t = 1.0$ ; (b)  $d\Delta t = 2.0$ ; (c)  $d\Delta t = 3.0$ ; (d)  $d\Delta t = 4.0$ .

Using Eq. (13), the second order WCW scheme assumes the following form:

$$u_{n+1} = \left( I + a \frac{\Delta t}{2} + \frac{a^2}{2} \left( \frac{\Delta t}{2} \right)^2 + \frac{a^3}{6} \left( \frac{\Delta t}{2} \right)^3 \right) \left( I - \frac{r \Delta t}{2} \right)^{-1} e^{-d \Delta t} \left( I + \frac{r \Delta t}{2} \right) \left( I + a \frac{\Delta t}{2} + \frac{a^2}{2} \left( \frac{\Delta t}{2} \right)^2 + \frac{a^3}{6} \left( \frac{\Delta t}{2} \right)^3 \right) u_n. \quad (21)$$

Substituting  $u_n = e^{i n \theta}$  to Eq. (21) leads to

$$e^{i \theta} = e^{-d \Delta t} \left[ 1 + a \frac{\Delta t}{2} + \frac{1}{2} \left( a \frac{\Delta t}{2} \right)^2 + \frac{1}{6} \left( a \frac{\Delta t}{2} \right)^3 \right]^2 \left( \frac{2 + \lambda}{2 - \lambda} \right), \quad (22)$$

where  $\lambda = r \Delta t$  has a real part  $\lambda_r$  and an imaginary part  $\lambda_i$ .

For the WCW scheme, the stability regions always include the point  $(-2, 0)$  in the complex plane, for any values of  $a \Delta t$  and  $d \Delta t$  as shown by Eq. (22).

Also, as noted in Fig. 2, the stability regions of the WCW scheme have similar patterns as the CWC scheme. For the case  $a \Delta t = 0$ , both CWC and WCW schemes reduce to the second order IIF scheme and the stability curves coincide with the ones in [6].

### 2.3.3. A comparison between CWC and WCW schemes

Previously, it was observed that placing stiff operators at the end of the splitting sequence leads to smaller solution errors [5]. It is found below that the CWC scheme, which treats the stiff reaction–diffusion terms twice, has better stability properties.

From Eq. (19), the real part, denoted by  $\lambda_r^{CWC}$ , and the imaginary part, denoted by  $\lambda_i^{CWC}$ , of  $\lambda$  for CWC scheme take the form

$$\lambda_r^{CWC} = \begin{cases} \frac{4(1-f(\Delta t, d, a))}{1+2 \cos(\frac{\theta}{2}) \sqrt{f(\Delta t, d, a)} + f(\Delta t, d, a)}, & f(\Delta t, d, a) \geq 0, \\ \frac{4(1-|f(\Delta t, d, a)|)}{1+2 \sin(\frac{\theta}{2}) \sqrt{|f(\Delta t, d, a)|} + |f(\Delta t, d, a)|}, & f(\Delta t, d, a) \leq 0, \end{cases} \quad (23)$$

$$\lambda_i^{CWC} = \begin{cases} \frac{8 \sin(\frac{\theta}{2}) \sqrt{f(\Delta t, d, a)}}{1+2 \cos(\frac{\theta}{2}) \sqrt{f(\Delta t, d, a)} + f(\Delta t, d, a)}, & f(\Delta t, d, a) \geq 0, \\ \frac{-8 \cos(\frac{\theta}{2}) \sqrt{|f(\Delta t, d, a)|}}{1+2 \sin(\frac{\theta}{2}) \sqrt{|f(\Delta t, d, a)|} + |f(\Delta t, d, a)|}, & f(\Delta t, d, a) \leq 0, \end{cases} \quad (24)$$

where

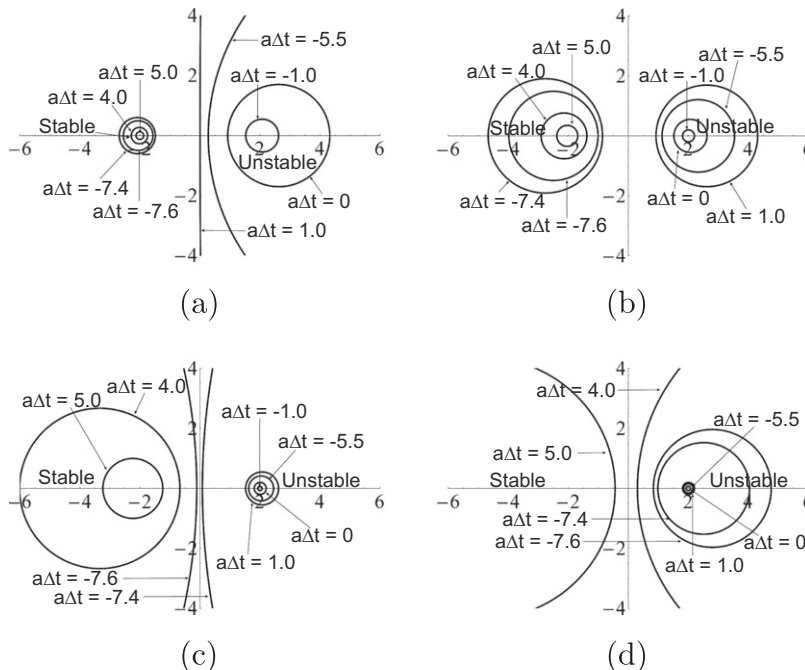


Fig. 2. Stability regions for the second order WCW scheme with different  $a \Delta t$ . (a)  $d \Delta t = 1.0$ ; (b)  $d \Delta t = 2.0$ ; (c)  $d \Delta t = 3.0$ ; (d)  $d \Delta t = 4.0$ .

$$f(\Delta t, d, a) = e^{-d\Delta t} \left[ 1 + a\Delta t + \frac{1}{2}(a\Delta t)^2 + \frac{1}{6}(a\Delta t)^3 \right]. \quad (25)$$

Similarly, for the boundary of stability region of WCW scheme from Eq. (22), the real part, denoted by  $\lambda_r^{WCW}$ , and the imaginary part, denoted by  $\lambda_i^{WCW}$ , of  $\lambda$  are

$$\lambda_r^{WCW} = \frac{2(1 - g^2(\Delta t, d, a))}{1 + 2 \cos(\theta)g(\Delta t, d, a) + g^2(\Delta t, d, a)}, \quad (26)$$

$$\lambda_i^{WCW} = \frac{2 \sin(\theta)g(\Delta t, d, a)}{1 + 2 \cos(\theta)g(\Delta t, d, a) + g^2(\Delta t, d, a)}, \quad (27)$$

where

$$g(\Delta t, d, a) = e^{-d\Delta t} \left[ 1 + \frac{a\Delta t}{2} + \frac{1}{2} \left( \frac{a\Delta t}{2} \right)^2 + \frac{1}{6} \left( \frac{a\Delta t}{2} \right)^3 \right]^2. \quad (28)$$

As evident in the Figs. 1 and 2, the boundaries of the stability region are always circles with centers on the real axis, implying that the  $y$ -coordinate of the center is always zero. From Eq. (23), the  $x$ -coordinate of the center for the stability region's boundary of CWC, denoted by  $x_{CWC}$ , is

$$x_{CWC} = 4 \frac{1 + |f(\Delta t, d, a)|}{1 - |f(\Delta t, d, a)|}. \quad (29)$$

And the radius, denoted by  $r_{CWC}$ , is

$$r_{CWC}(\Delta t, d, a) = 8 \left| \frac{\sqrt{|f(\Delta t, d, a)|}}{1 - |f(\Delta t, d, a)|} \right|. \quad (30)$$

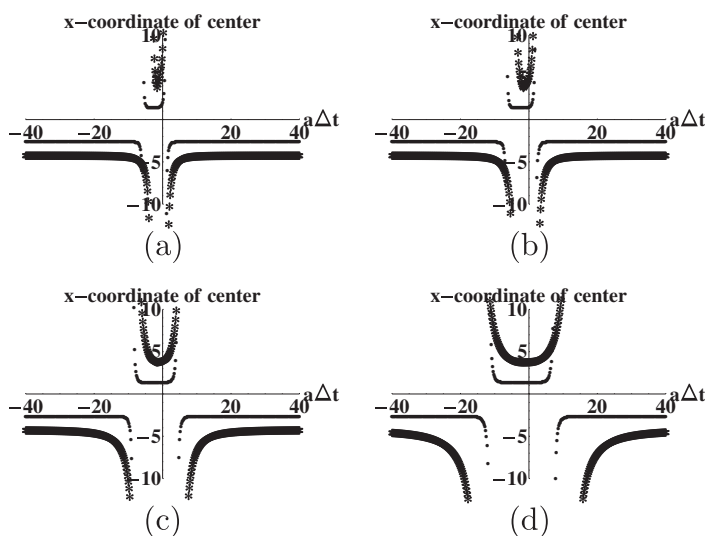
Similarly, from Eq. (26), the  $x$ -coordinate of the center of stability region's boundary of WCW, denoted by  $x_{WCW}$ , is

$$x_{WCW} = 2 \frac{1 + g^2(\Delta t, d, a)}{1 - g^2(\Delta t, d, a)}. \quad (31)$$

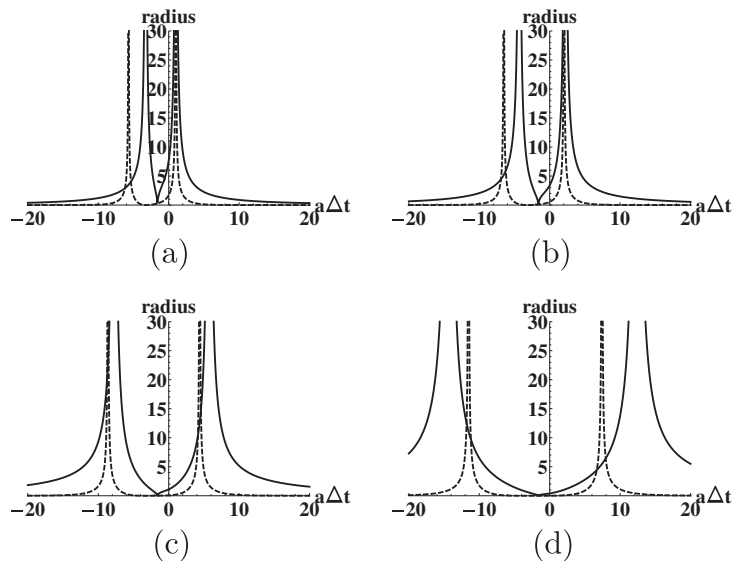
And the radius  $r_{WCW}$  is the following:

$$r_{WCW}(\Delta t, d, a) = 4 \left| \frac{g(\Delta t, d, a)}{1 - g^2(\Delta t, d, a)} \right|. \quad (32)$$

It is obvious that  $r_{CWC} \leq |x_{CWC}|$  and  $r_{WCW} \leq |x_{WCW}|$ , which implies the stability regions' boundaries stay within either of half planes. As shown in Fig. 3, the centers of stability regions' boundaries of both CWC and WCW schemes follow the same patterns: as  $a\Delta t$  increases from  $-\infty$  to  $\infty$ , the center moves from the left half complex plane to the right half through jumping over a singularity, and moves back then to the left half plane, centering around a point.



**Fig. 3.**  $x$ -coordinates of centers of stability region's boundaries versus  $a\Delta t$  for CWC (star) and WCW (dot) schemes: (a)  $d\Delta t = 1.0$ ; (b)  $d\Delta t = 2.0$ ; (c)  $d\Delta t = 4.0$ ; (d)  $d\Delta t = 6.0$ .



**Fig. 4.** Radius of stability region versus  $a\Delta t$  for CWC (solid) and WCW (dashed) schemes: (a)  $d\Delta t = 1.0$ ; (b)  $d\Delta t = 2.0$ ; (c)  $d\Delta t = 4.0$ ; (d)  $d\Delta t = 6.0$ .

Fig. 4 demonstrates that the radius of each stability region's boundary diverges for two values of  $a\Delta t$ , which are where the region's boundary shifts between left and right halves of the complex plane. When the value of  $a\Delta t$  is between these two peaks, the stability regions' boundaries locate on the right half complex plane (see Fig. 3), implying that the stability region of the scheme includes the whole left half complex plane. When  $a\Delta t$  is not between two peaks, the stability regions are bounded within the left half of the complex plane. As a result, the center and radius curves for both schemes dictate the stability properties.

There are three main differences between the center and radius curves of the two schemes. First of all, the size of the interval between the two divergent CWC radius curves is more sensitive to the value of  $d\Delta t$  than that of WCW radius curves. Specifically, when  $d\Delta t$  increases, the range of  $a\Delta t$  values that maintain the inclusion of the left half of the complex plane in the stability region expands at a faster rate for the CWC scheme than for the WCW scheme. Secondly, when  $d\Delta t$  is chosen such that the stability region is entirely contained within the left half of the complex plane, the radii of stability region of CWC are greater than WCW for most of values of  $a\Delta t$ . Finally, the "width" and "steepness" of the peaks of radius curves are different between two schemes; the peaks for CWC are broader and less steep than those of WCW for the same set of parameters, indicating CWC stability region are less sensitive to the changes of  $a\Delta t$  than those of WCW stability region.

In general, the WCW scheme is more sensitive to choice of the size of  $\Delta t$ , diffusion and advection in terms of stability properties than the CWC scheme. As a result, for nonlinear reaction–diffusion–advection equations, the CWC scheme is naturally more robust than the WCW scheme in choice of  $\Delta t$ . Direct simulations on two nonlinear systems in the next section also confirm this result.

**Remark 1.** Although the linear stability analysis performed in this section demonstrates several important features of the new methods, it would be interesting to study nonlinear stability of the splitting method [24–27] for additional insight and better understanding of each method.

**Remark 2.** Because of the explicit treatment of the advection term, both CWC and WCW schemes still need to satisfy CFL condition. For an advection-dominated case ( $a \gg d$ ), the value of  $\Delta t$  has to be small due to the CFL condition. Both CWC and WCW schemes are designed to deal with the stiffness in reactions and the stability constraint in diffusion for a diffusion–reaction–advection system.

### 3. Numerical tests

In this section, we investigate how the splitting sequence and choice of schemes for the reaction–diffusion part of a stiff reaction–diffusion–advection system may affect the accuracy and efficiency of a scheme by solving a linear system with various combinations of ADA and DAD schemes. In the comparison, we always treat the hyperbolic part of the system with the third order WENO/TVD, and use either second order Runge–Kutta (RK2) or IIF/cIIF to treat the diffusion–reaction part with IIF for the one dimensional system and cIIF for two- or three-dimensional systems. Specifically, we study two ADA schemes: WCW and WENO-RK2-WENO (denoted by WRW), and two DAD schemes: CWC and RK2-WENO-RK2 (denoted by RWR).



The four schemes are tested using the following two-variable linear reaction–diffusion–advection system on a rectangle  $\Omega = (0, 2\pi)^k \subset \mathbf{R}^k$  for  $k = 1, 2, 3$ ,

$$\begin{cases} \frac{\partial u}{\partial t} + \nabla \cdot (au) = d\Delta u - bu + v, \\ \frac{\partial v}{\partial t} + \nabla \cdot (av) = d\Delta v - cv, \end{cases} \quad (33)$$

with periodic boundary conditions. For an initial condition with the following form,

$$u|_{t=0} = \cos(x + y + z), \quad v|_{t=0} = \cos(x + y + z), \quad (34)$$

the system has the following exact solution in three-dimensions,

$$\begin{cases} u(x, y, z, t) = (e^{-(b+d)t} + e^{-(c+d)t}) \cos(x + y + z - at), \\ v(x, y, z, t) = (b - c)e^{-(c+d)t} \cos(x + y + z - at). \end{cases} \quad (35)$$

We stiffen the reaction by choosing the set of parameters  $a = c = d = 1$  and  $b = 100$  as a means of examining the effectiveness of different schemes by observing how each handles the stiffness of the reaction [6].

We first study RWR and WRW schemes for the one-dimensional system. As expected, the time step must be proportional to the square of the spatial grid size when the explicit Runge–Kutta is used to solve the diffusion–reaction part as seen in Table 1 due to the stability constraint from the diffusion term. At the same numerical resolution, i.e. for the same number of spatial grid points, both RWR and WRW schemes have similar size of errors and a similar order of accuracy, which is two. For both schemes, when  $N$  doubles, the CPU time increases by a fold of eight in Table 1 because the time steps have to be decreased by a factor of four to meet the stability constraint on diffusion. The CPU time for RWR is about half of that for WRW which is likely because one WENO step is more expensive than one Runge–Kutta step and WRW consists of two WENO steps while RWR has only one WENO step.

The advantages of CWC and WCW schemes over RWR and WRW are evident in Table 2. Although all four schemes achieve similar sizes of errors and an order of two accuracy, the CPU times for CWC and WCW schemes are smaller than RWR and WRW schemes at the same  $N$  with similar errors. This results from the broader stability properties of CWC and WCW schemes that arise from the use of IIF on the reaction–diffusion part of the system.

The improved efficiency of CWC and WCW schemes in comparison to RWR and WRW schemes is highlighted even more so in two- and three-dimensional simulations as shown in Tables 3 and 4. The accuracy and size of errors for the four different schemes are very similar using a fixed spatial resolution  $N$ . For example, CPU times for the WRW scheme are eight to nine times larger than those for the WCW scheme in two-dimensional and three-dimensional systems for  $N = 80$ .

For the case of low spatial resolution  $N = 10$ , both WCW and WRW schemes show divergence while CWC and RWR schemes remain to converge. This indicates that the DAD splitting sequence may be more robust in both stability and error

**Table 1**

Error, accuracy and CPU time of RWR and WRW schemes for a one-dimensional system.  $N$  is the number of spatial grid points and NC stands for no convergence.

| $N$ | RWR & WRW<br>$\Delta t = \Delta x/(6\pi)$ |       |         | RWR<br>$\Delta t = \Delta x^2/(6\pi)$ |       |         | WRW<br>$\Delta t = \Delta x^2/(6\pi)$ |       |         |
|-----|---|-------|---------|---------------------------------------|-------|---------|---------------------------------------|-------|---------|
|     | Error                                     | Order | CPU (s) | Error                                 | Order | CPU (s) | Error                                 | Order | CPU (s) |
| 10  | NC  | –     | –       | 0.859                                 | –     | 1.00e–3 | NC                                    | –     | –       |
| 20  | NC  | –     | –       | 0.136                                 | 2.65  | 8.00e–3 | 0.136                                 | –     | 1.40e–2 |
| 40  | NC  | –     | –       | 2.16e–2                               | 2.65  | 6.00e–2 | 2.16e–2                               | 2.65  | 0.111   |
| 80  | NC  | –     | –       | 5.56e–3                               | 1.96  | 0.477   | 5.56e–3                               | 1.96  | 0.891   |
| 160 | NC  | –     | –       | 1.39e–3                               | 1.99  | 3.80    | 1.39e–3                               | 1.99  | 7.11    |
| 320 | NC  | –     | –       | 3.61e–4                               | 1.94  | 30.7    | 3.61e–4                               | 1.94  | 57.4    |
| 640 | NC  | –     | –       | 9.80e–5                               | 1.88  | 244.7   | 9.80e–5                               | 1.88  | 457.9   |

**Table 2**

Error, accuracy and CPU time of CWC and WCW schemes for a one-dimensional system.  $N$  is the number of spatial grid points and NC stands for no convergence.

| $N$ | CWC<br>$\Delta t = \Delta x/(6\pi)$ |       |         | WCW<br>$\Delta t = \Delta x/(6\pi)$ |       |         |
|-----|-------------------------------------|-------|---------|-------------------------------------|-------|---------|
|     | Error                               | Order | CPU (s) | Error                               | Order | CPU (s) |
| 10  | 0.866                               | –     | 2.00e–3 | NC                                  | –     | –       |
| 20  | 0.136                               | 2.67  | 6.00e–3 | 0.138                               | –     | 9.00e–3 |
| 40  | 2.16e–2                             | 2.65  | 2.30e–2 | 2.16e–2                             | 2.68  | 3.00e–2 |
| 80  | 5.55e–3                             | 1.96  | 0.144   | 5.54e–3                             | 1.96  | 0.176   |
| 160 | 1.39e–3                             | 1.99  | 1.18    | 1.39e–3                             | 1.99  | 1.47    |
| 320 | 3.60e–4                             | 1.94  | 11.9    | 3.59e–4                             | 1.94  | 11.1    |
| 640 | 9.80e–5                             | 1.88  | 110.5   | 9.77e–5                             | 1.88  | 94.4    |

**Table 3**

Error, accuracy and CPU time of RWR and WRW schemes in two- and three-dimensions.

|        | N  | RWR & WRW<br>$\Delta t = \Delta x/(6\pi)$ |       |         | RWR<br>$\Delta t = \Delta x^2/(6\pi)$ |       |         | WRW<br>$\Delta t = \Delta x^2/(6\pi)$ |         |         |
|--------|----|---|-------|---------|---------------------------------------|-------|---------|---------------------------------------|---------|---------|
|        |    | Error                                     | Order | CPU (s) | Error                                 | Order | CPU (s) | Error                                 | Order   | CPU (s) |
|        |    | 2-Dim.                                    | 10    | NC      | –                                     | –     | 4.98    | –                                     | 2.00e-2 | NC      |
|        | 20 | NC  | –     | –       | 0.745                                 | 2.74  | 0.308   | 0.276                                 | –       | 0.564   |
|        | 40 | NC  | –     | –       | 0.118                                 | 2.66  | 4.91    | 4.36e-2                               | 2.66    | 8.99    |
|        | 80 | NC  | –     | –       | 2.98e-2                               | 1.99  | 78.7    | 9.51e-3                               | 2.20    | 144.1   |
| 3-Dim. | 10 | NC  | –     | –       | 2.93                                  | –     | 2.98    | NC                                    | –       | –       |
|        | 20 | NC  | –     | –       | 0.413                                 | 2.83  | 9.45    | 0.413                                 | –       | 17.6    |
|        | 40 | NC  | –     | –       | 6.39e-2                               | 2.69  | 315.4   | 6.40e-2                               | 2.69    | 587.9   |
|        | 80 | NC  | –     | –       | 1.34e-2                               | 2.25  | 10658.6 | 1.34e-2                               | 2.26    | 19088.1 |

**Table 4**

Error, accuracy and CPU time of CWC and WCW methods in two- and three-dimensions.

|        | N  | CWC<br>$\Delta t = \Delta x/(6\pi)$ |       |         | WCW<br>$\Delta t = \Delta x/(6\pi)$ |         |         |
|--------|----|-------------------------------------|-------|---------|-------------------------------------|---------|---------|
|        |    | Error                               | Order | CPU (s) | Error                               | Order   | CPU (s) |
|        |    | 2-Dim.                              | 10    | 1.73    | –                                   | 3.20e-2 | NC      |
|        | 20 | 0.272                               | 2.67  | 0.161   | 0.275                               | –       | 0.279   |
|        | 40 | 4.33e-2                             | 2.65  | 1.42    | 4.32e-2                             | 2.67    | 1.81    |
|        | 80 | 1.11e-2                             | 1.96  | 15.3    | 1.11e-2                             | 1.96    | 16.2    |
| 3-Dim. | 10 | 2.597                               | –     | 0.496   | NC                                  | –       | –       |
|        | 20 | 0.407                               | 2.67  | 4.91    | 0.412                               | –       | 8.26    |
|        | 40 | 6.43e-2                             | 2.66  | 93.3    | 6.42e-2                             | 2.68    | 116.7   |
|        | 80 | 1.56e-2                             | 2.04  | 2563.1  | 1.56e-2                             | 2.04    | 2388.9  |

control than ADA splitting sequence for stiff systems. This observation is consistent with the linear stability analysis and previous results on splitting sequences for stiff systems [5].

Remark: In the splitting method, the advection step and reaction–diffusion step are decoupled. The method used for solving the convection step, such as WENO used here, could be directly applied to nonlinear advection terms.

#### 4. Application to two biological systems

Reaction–diffusion–advection systems often arise in modeling biological systems, where the advection may stem from tissue growth [28,29], tumor formation [30], population dispersion [31,32], interstitial fluid flow [33] or various other sources. Often reactions involving interactions among components at different time and spatial scales are stiff and contain rate constants that vary by several orders of magnitude. In this section, we consider a one-dimensional morphogen system and also a two-dimensional pattern formation system and use both CWC and WCW schemes to handle stiffness of the diffusion–reaction–advection systems.

##### 4.1. A one-dimensional morphogen system with advection

Experimental evidence demonstrates that in the developing *Drosophila* wing disc, the slope of the gradient of the morphogen Decapentaplegic (Dpp) secreted from the dorsal–ventral axis regulates the growth of the disc [34]. This process, along with the binding of Dpp with its receptor Thickveins (Tkv), has been modeled on a growing domain [35]. Here, we study a simplified system with constant advection and nonlinear reactions on a fixed domain. Below is the system of equations [35] in a non-dimensionalized form,

$$\begin{aligned}
 \frac{\partial d}{\partial t} + a \frac{\partial d}{\partial x} &= D \frac{\partial^2 d}{\partial x^2} - \epsilon(k_1 dr + k_{-1} b) - \alpha d, \\
 \frac{\partial r}{\partial t} + a \frac{\partial r}{\partial x} &= -k_1 dr + k_{-1} b + \frac{\rho}{(b+r)^{n_1+1}} - \beta r \left(1 + \frac{\sigma d^{n_2}}{h^{n_2+d^{n_2}}}\right), \\
 \frac{\partial b}{\partial t} + a \frac{\partial b}{\partial x} &= k_1 dr - k_{-1} b - b,
 \end{aligned} \tag{36}$$

on the interval  $[0, 1]$  with no-flux boundary conditions at both ends. Here,  $d$ ,  $r$ , and  $b$ , represent Dpp, Tkv, and their bound complex, respectively;  $D$  is the diffusion coefficient for  $d$ ;  $a$  is the flux coefficient for the tissue growth;  $k_1$  and  $k_{-1}$  are the on- and off-reaction rates of the ligand–receptor binding;  $\alpha$  and  $\beta$  are degradation rates for  $d$  and  $r$ ;  $\rho$  and  $\sigma$  are effective maximum feedback coefficients;  $h$  is an EC-50 value for the feedback of  $d$  onto  $r$ ; and  $\epsilon$  arises from scaling the equation.

**Table 5**

Errors, orders of accuracy in time, and CPU time for numerical simulations of Eq. (36). The chosen parameters are:  $D = a = k_{-1} = \rho = \sigma = \beta = h = n_1 = n_2 = 1$ ,  $k_1 = \epsilon = \alpha = 10$ . Initial conditions are given by  $d(x, t = 0) = r(x, t = 0) = b(x, t = 0) = 3/2 + \cos(\pi x)$ .

| $\Delta t$ | CWC     |       |         | WCW     |       |         |
|------------|---------|-------|---------|---------|-------|---------|
|            | Error   | Order | CPU (s) | Error   | Order | CPU (s) |
| 2.0e-3     | 6.18e-5 | –     | 7.30e-2 | 2.54e-4 | –     | 9.10e-2 |
| 1.0e-3     | 1.54e-5 | 2.01  | 0.117   | 6.34e-5 | 2.00  | 0.166   |
| 5.0e-4     | 3.80e-6 | 2.01  | 0.214   | 1.57e-5 | 2.02  | 0.302   |
| 2.5e-4     | 9.05e-7 | 2.01  | 0.412   | 3.73e-6 | 2.07  | 0.586   |
| 1.25e-4    | 1.81e-7 | –     | 0.814   | 7.47e-7 | –     | 1.16    |
| 6.25e-5    | –       | –     | 1.57    | –       | –     | 2.30    |

**Table 6**

Errors, orders of accuracy in time, and CPU time for numerical simulations of Eq. (36). The chosen parameters given by the original Dpp-Tkv model in [35] are:  $D = a = \epsilon = 1$ ,  $k_1 = 5$ ,  $k_{-1} = 0.1$ ,  $\alpha = 10$ ,  $\rho = 0.8$ ,  $\beta = 0.5$ ,  $\sigma = 10$ ,  $h = 20$ ,  $n_1 = n_2 = 4$ . Initial conditions are given by  $d(x, t = 0) = r(x, t_0) = b(x, t_0) = 10 + 10\cos(\pi x)$ .

| $\Delta t$ | CWC     |       |         | WCW     |       |         |
|------------|---------|-------|---------|---------|-------|---------|
|            | Error   | Order | CPU (s) | Error   | Order | CPU (s) |
| 1.0e-2     | 8.69e-2 | –     | 3.60e-2 | 0.382   | –     | 4.10e-2 |
| 5.0e-3     | 2.11e-2 | 2.04  | 4.70e-2 | 8.81e-2 | 2.11  | 5.40e-2 |
| 2.5e-3     | 5.16e-3 | 2.03  | 7.70e-2 | 2.13e-2 | 2.05  | 8.50e-2 |
| 1.25e-3    | 1.22e-3 | 2.08  | 0.121   | 5.05e-3 | 2.08  | 0.153   |
| 6.25e-4    | 2.44e-4 | –     | 0.211   | 1.01e-3 | –     | 0.272   |
| 3.125e-4   | –       | –     | 0.398   | –       | –     | 0.511   |

We choose two different sets of parameters and initial conditions as shown in Tables 5 and 6 such that the maximum and minimum eigenvalues of the Jacobian of the reaction terms at the initial state are on the orders of  $-100$  and  $-1$ , respectively. We apply CWC and WCW schemes to solve both stiff systems up to non-constant states at time  $t = 0.5$  for both systems. The listed error values in Tables 5 and 6 are calculated by a maximal difference between each simulation and the one using the smallest  $\Delta t$  since an analytical solution for the system cannot be obtained. In Tables 5 and 6, the orders of the second to last row are not calculated, because the order of accuracy would artificially increase when the resolution of simulation is too close to the reference resolution [36]. Different from the linear cases, the order of accuracy in time is calculated by varying  $\Delta t$  on a fixed spatial grid of size  $N = 40$ . Of course, the largest  $\Delta t$  used in the calculations satisfies the CFL condition.

Both CWC and WCW schemes achieve a second order accuracy in time with comparable CPU times, yet it is important to note a stark contrast in error values calculated. Using the same grid size and time step, CWC schemes achieve an approximate fourfold smaller error value. As a result, a WCW scheme must use half the time-step to obtain an error value comparable to that of a CWC scheme. From here, it is evident that the CWC and WCW schemes achieve the same order of accuracy although CWC is still more accurate for computing stiff nonlinear systems, which differs from the results of the stiff linear testing case. This highlights the benefits of splitting the stiff reaction part twice as opposed to the advection part twice. As a result, it seems fitting to implement CWC in nonlinear systems as it may reach desired results more efficiently than WCW. This very notion is visually demonstrated in the following two-dimensional patterning systems.

#### 4.2. A two-dimensional patterning system with advection

Pattern formation in biological species have been well-modeled by reaction–diffusion systems; however it has also been explored using reaction–diffusion–advection systems to account for growth in developmental systems [37,38] and applied to ecological control strategies [39] and phyllotaxic systems [40].

Consider the following Gierer–Meinhardt system [41] with added advection [29],

$$\begin{cases} \frac{\partial u}{\partial t} + a \nabla \cdot u = d_1 \Delta u + p \frac{u^2}{v} - q_1 u + w_1, \\ \frac{\partial v}{\partial t} + a \nabla \cdot v = d_2 \Delta v + p u^2 - q_2 v + w_2, \end{cases} \quad (37)$$

on a rectangle  $(0, 10) \times (0, 10)$  with periodic boundary conditions. The error values listed for CWC and WCW in Table 7 are again calculated through a maximal difference between each simulation and the one using the smallest time step size. We first start from a smooth initial condition and simulate the system up to a transient non-constant state at time  $t = 0.5$ .

Similar to the one-dimensional morphogen system, both CWC and WCW simulations achieve a second order accuracy with comparable CPU times, but differences remain in error values. The CWC scheme has an approximately fourfold smaller error value than that of its WCW counterpart. Since the same fourfold difference in the two schemes is achieved by halving time alone in the one-dimensional morphogen system, it follows that this difference likely occurs as an error in time.

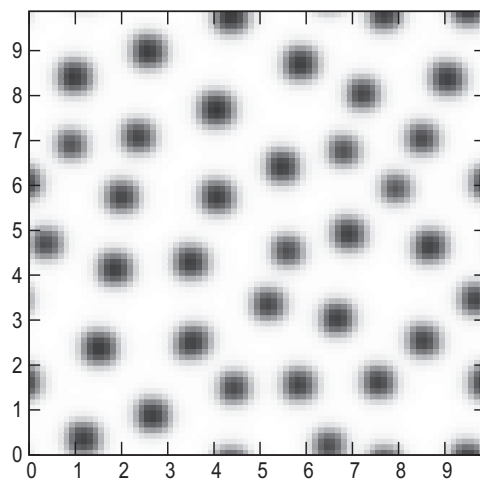
**Table 7**

Errors, orders of accuracy, and CPU time for numerical simulations of Eq. (37). The chosen parameters are:  $a = 0.3$ ,  $d_1 = 10^{-3}$ ,  $d_2 = 50$ ,  $p = 0.1$ ,  $q_1 = 1$ ,  $q_2 = 100$ ,  $w_1 = w_2 = 1$ . The smooth initial conditions are given by  $u(x, t = 0) = v(x, t = 0) = \cos(\frac{\pi}{5}(x + y)) + 10$ .

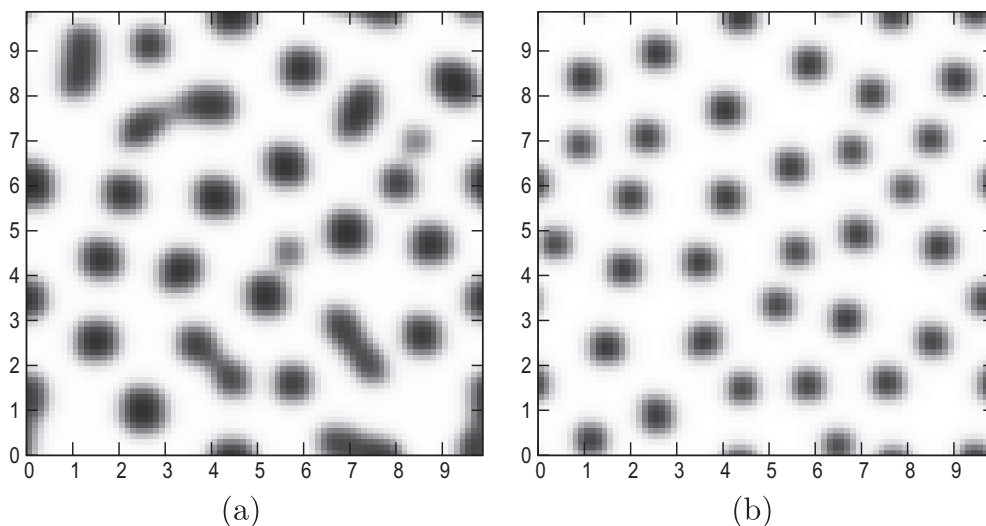
| $\Delta t$ | CWC ( $\Delta t = \Delta x/20$ ) |       |         | WCW ( $\Delta t = \Delta x/20$ ) |       |         |
|------------|----------------------------------|-------|---------|----------------------------------|-------|---------|
|            | Error                            | Order | CPU (s) | Error                            | Order | CPU (s) |
| 20         | 0.575                            | –     | 9.90e–2 | –                                | –     | NC      |
| 40         | 0.125                            | 2.20  | 0.715   | 0.507                            | –     | 0.936   |
| 80         | 2.76e–2                          | 2.18  | 7.12    | 0.109                            | 2.21  | 7.73    |
| 160        | 5.44e–3                          | 2.34  | 87.2    | 2.14e–2                          | 2.35  | 77.8    |
| 320        | –                                | –     | 1305.9  | –                                | –     | 952.6   |

The difference in error between CWC and WCW is more significant when the initial conditions of the system are chosen to be white noise ranging between values of 0 and 10. For this set of initial conditions, the system produces spotty patterns as displayed in Fig. 5 and 6. For a time step of  $\Delta t = 6.25 \times 10^{-3}$ , the spots generated by the CWC scheme in Fig. 5 are clear and distinct while those generated by the WCW scheme in Fig. 6(a) are reasonably more obscure and less refined.

Halving the time step to  $\Delta t = 3.125 \times 10^{-3}$  for a WCW simulation resolves the pattern and the result stands quite comparable to the one produced by the original CWC simulation as shown in Fig. 6(b).



**Fig. 5.** Patterns generated by CWC scheme simulations on an  $80 \times 80$  grid up to  $t = 30$  with  $\Delta t = 6.25 \times 10^{-3}$ . The parameters of the system are the same as those used in Table 7.



**Fig. 6.** Patterns generated by WCW scheme simulations on an  $80 \times 80$  grid up to  $t = 30$  with (a)  $\Delta t = 6.25 \times 10^{-3}$  and (b)  $\Delta t = 3.125 \times 10^{-3}$ . The parameters of the system are the same as those used in Fig. 5.

When considering the errors produced in both nonlinear systems, we note that this discrepancy in the patterns produced by the two schemes is essentially a reflection of the differences in error of the simulations. This demonstrates that the CWC scheme achieves better accuracy than the WCW scheme in solving nonlinear stiff reaction–diffusion–advection systems. This is consistent with the previous observation that the DAD splitting sequence should be more desirable for advection–diffusion–reaction systems [5].

## 5. Conclusions

Stiff reaction–diffusion–advection equations are difficult to solve, because each part of the equation exhibits different characteristics that require particular algorithms for treatment. In this paper, we have introduced a family of second order splitting scheme for reaction–diffusion–advection equations in which the reaction–diffusion term is solved by cIIF while WENO is used for the advection term. A key benefit of cIIF is its unconditional linear stability (of its second order version), leading to better stability properties for the overall splitting method, in particular, when applied to a system with stiff reactions. Stability analysis and direct simulations have demonstrated excellent efficiency of this new class of methods. In particular, we have explored two different splitting sequences: DAD (diffusion–reaction, advection, and diffusion–reaction) and ADA (advection, diffusion–reaction, and advection). Both linear stability analysis and nonlinear simulations of stiff systems have suggested that the DAD splitting approach, in which cIIF is used twice for the reaction–diffusion term and WENO once for the advection term at each time step (CWC scheme), performs better than the ADA splitting (WCW scheme).

The overall approach of CWC scheme (or WCW scheme) can be directly extended to systems with adaptive mesh refinement in space and other coordinates (e.g. polar coordinates) because of the flexibility of cIIF in inclusion of complex spatial components [42]. It is conceivable that for an advection-dominated system, the WCW scheme in which WENO is utilized twice at each time step may be more advantageous than CWC scheme. In general, both CWC and WCW schemes that are easy to implement with good stability properties are efficient algorithms for solving reaction–diffusion–advection equations. Although the cases studied here are with linear advection terms, the approach can be directly applied to the nonlinear advection terms through robust hyperbolic solvers such as WENO schemes.

## Acknowledgments

This work was partially supported by NIH grants R01GM75309, R01GM67247, and P50GM76516, and NSF grant DMS-0917492 (to Q. N.), and NSF grants DMS-1019544 (to X. L.) and DMS-0810413 (to Y. Z.).

## References

- [1] D.A. Fournier, J.R. Sibert, J. Hampton, P.J. Bills, An advection–diffusion–reaction model for the estimation of fish movement parameters from tagging data, with application to skipjack tuna (*Katsuwonus pelamis*), *Canadian Journal of Fisheries and Aquatic Sciences* 56 (1999) 925–938.
- [2] G. Hauke, A simple subgrid scale stabilized method for the advection–diffusion–reaction equation, *Computer Methods in Applied Mechanics and Engineering* 191 (2002) 2925–2947.
- [3] R. Dillon, H.G. Othmer, A mathematical model for outgrowth and spatial patterning of the vertebrate limb bud, *Journal of Theoretical Biology* 197 (1999) 295–330.
- [4] D.L. Ropp, J.N. Shadid, Stability of operator splitting methods for systems with indefinite operators: advection–diffusion–reaction systems, *Journal of Computational Physics* 228 (2009) 3508–3516.
- [5] B. Sportisse, An analysis of operator splitting techniques in the stiff case, *Journal Computational Physics* 161 (1) (2000) 140–168.
- [6] Q. Nie, Y.-T. Zhang, R. Zhao, Efficient semi-implicit schemes for stiff systems, *Journal of Computational Physics* 214 (2) (2006) 521–537.
- [7] Q. Nie, F.Y.M. Wan, Y.-T. Zhang, X.-F. Liu, Compact integration factor methods in high spatial dimensions, *Journal of Computational Physics* 227 (10) (2008) 5238–5255.
- [8] C.-W. Shu, High order weighted essentially non-oscillatory schemes for convection dominated problems, *SIAM Review* 51 (1) (2009) 82–126.
- [9] G. Beylkin, J.M. Keiser, L. Vozovoi, A new class of time discretization schemes for the solution of nonlinear PDEs, *Journal of Computational Physics* 147 (1998) 362–387.
- [10] T.Y. Hou, J. Lowengrub, M.J. Shelley, Removing the stiffness from interfacial flows with surface tension, *Journal of Computational Physics* 114 (1994) 312.
- [11] P.H. Leo, J.S. Lowengrub, Q. Nie, Microstructural evolution in orthotropic elastic media, *Journal of Computational Physics* 157 (2000) 44–88.
- [12] H.J. Jou, P.H. Leo, J.S. Lowengrub, Microstructural evolution in inhomogeneous elastic media, *Journal of Computational Physics* 131 (1997) 109.
- [13] Q. Du, W. Zhu, Stability analysis and applications of the exponential time differencing schemes, *Journal of Computational Mathematics* 22 (2004) 200.
- [14] Q. Du, W. Zhu, Modified exponential time differencing schemes: analysis and applications, *BIT Numerical Mathematics* 45 (2005) 307–328.
- [15] A.-K. Kassam, L.N. Trefethen, Fourth-order time stepping for stiff PDEs, *SIAM Journal on Scientific Computing* 26 (2005) 1214–1233.
- [16] X.-D. Liu, S. Osher, T. Chan, Weighted essentially nonoscillatory schemes, *Journal of Computational Physics* 115 (1994) 200–212.
- [17] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, *Journal of Computational Physics* 126 (1996) 202–228.
- [18] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, *Journal of Computational Physics* 77 (1988) 439–471.
- [19] B. Van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov’s method, *Journal of Computational Physics* 32 (1) (1979) 101–136.
- [20] A. Harten, High resolution schemes for hyperbolic conservation laws, *Journal of Computational Physics* 49 (3) (1983) 357–393.
- [21] P. Colella, P.R. Woodward, The Piecewise Parabolic Method (PPM) for gas-dynamical simulations, *Journal of Computational Physics* 54 (1) (1984) 174–201.
- [22] G. Strang, On the construction and comparison of difference schemes, *SIAM Journal on Numerical Analysis* 5 (3) (1968) 506–517.
- [23] S. Gottlieb, C.-W. Shu, Total variation diminishing Runge–Kutta schemes, *Mathematics of Computation* 67 (221) (1998) 73–85.
- [24] G. Dahlquist, Error analysis for a class of methods for stiff non-linear initial value problems, *Lecture Notes in Mathematics* 506 (1976) 60–72.
- [25] E. Hairer, M. Zennaro, On error growth functions of Runge–Kutta methods, *Applied Numerical Mathematics* 22 (1996) 205–216.
- [26] E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, Springer, 1996.

- [27] A. Stuart, A.R. Humphries, Model problems in numerical stability theory for initial value problems, *SIAM Review* 36 (1994) 226–257.
- [28] C.-S. Chou, W.-C. Lo, K.K. Gokoffski, Y.-T. Zhang, F.Y.M. Wan, A.D. Lander, A.L. Calof, Q. Nie, Spatial dynamics of multi-stage cell-lineages in tissue stratification, *Biophysical Journal* 99 (10) (2010) 3145–3154.
- [29] E.J. Crampin, E.A. Gaffney, P.K. Maini, Reaction and diffusion on growing domains: scenarios for robust pattern formation, *Bulletin of Mathematical Biology* 61 (6) (1999) 1093–1120.
- [30] S.M. Wise, J.S. Lowengrub, H.B. Frieboes, V. Cristini, Three-dimensional multispecies nonlinear tumor growth–I. Model and numerical method, *Journal of Theoretical Biology* 253 (3) (2008) 524–543.
- [31] R. Hambrock, Y. Lou, The evolution of conditional dispersal strategies in spatially heterogeneous habitats, *Bulletin of Mathematical Biology* 71 (8) (2009) 1793–1817.
- [32] M.A. Lewis, P. Kareiva, Allee dynamics and the spread of invading organisms, *Theoretical Population Biology* 43 (2) (1993) 141–158.
- [33] P.H. Feenstra, C.A. Taylor, Drug transport in artery walls: a sequential porohyperelastic-transport approach, *Computer Methods in Biomechanics and Biomedical Engineering* 12 (3) (1987) 263–276.
- [34] D. Rogulja, K.D. Irvine, Regulation of cell proliferation by a morphogen gradient, *Cell* 123 (3) (2005) 449–461.
- [35] R.E. Baker, P.K. Maini, A mechanism for morphogen-controlled domain growth, *Journal of Mathematical Biology* 54 (5) (2007) 597–622.
- [36] C. Ober, J. Shadid, Studies on the accuracy of time-integration methods for the radiation-diffusion equations, *Journal of Computational Physics* 195 (2004) 743–772.
- [37] A.J. Perumpanani, J.A. Sherratt, P.K. Maini, Phase differences in reaction–diffusion–advection systems and applications to morphogenesis, *IMA Journal of Applied Mathematics* 55 (1) (1995) 19–33.
- [38] R.A. Satnoianu, M. Menzinger, A general mechanism for inexact phase differences in reaction–diffusion–advection systems, *Physics Letters A* 304 (5–6) (2002) 149–156.
- [39] J.D. Murray, *Mathematical Biology II: Spatial Models and Biomedical Applications*, *Interdisciplinary Applied Mathematics*, vol. 18, Springer, New York, 2003.
- [40] G.P. Bernasconi, J. Boissonade, Phyllotactic order induced by symmetry breaking in advected turing patterns, *Physics Letters A* 232 (3–4) (1997) 224–230.
- [41] A. Gierer, H. Meinhardt, A theory of biological pattern formation, *Biological Cybernetics* 12 (1) (1972) 30–39.
- [42] X. Liu, Q. Nie, Compact integration factor methods for complex domains and adaptive mesh refinement, *Journal of Computational Physics* 229 (16) (2010) 5692–5706.