Published in final edited form as:
J Comput Phys. 2008 ; 227(10): 5238–5255. doi:10.1016/j.jcp.2008.01.050.

Compact integration factor methods in high spatial dimensions

Qing Nie[a,*], **Frederic Y.M. Wan**[a], **Yong-Tao Zhang**[b], and **Xin-Feng Liu**[a]

[a] Department of Mathematics, University of California, Irvine, CA 92697-3875, United States

[b] Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556-4618, United States

Abstract

The dominant cost for integration factor (IF) or exponential time differencing (ETD) methods is the repeated vector–matrix multiplications involving exponentials of discretization matrices of differential operators. Although the discretization matrices usually are sparse, their exponentials are not, unless the discretization matrices are diagonal. For example, a two-dimensional system of $N \times N$ spatial points, the exponential matrix is of a size of $N^2 \times N^2$ based on direct representations. The vector–matrix multiplication is of $O(N^4)$, and the storage of such matrix is usually prohibitive even for a moderate size N. In this paper, we introduce a compact representation of the discretized differential operators for the IF and ETD methods in both two- and three-dimensions. In this approach, the storage and CPU cost are significantly reduced for both IF and ETD methods such that the use of this type of methods becomes possible and attractive for two- or three-dimensional systems. For the case of two-dimensional systems, the required storage and CPU cost are reduced to $O(N^2)$ and $O(N^3)$, respectively. The improvement on three-dimensional systems is even more significant. We analyze and apply this technique to a class of semi-implicit integration factor method recently developed for stiff reaction–diffusion equations. Direct simulations on test equations along with applications to a morphogen system in two-dimensions and an intra-cellular signaling system in three-dimensions demonstrate an excellent efficiency of the new approach.

Keywords

Integration factor methods; Exponential time differencing methods; Stiff reaction-diffusion equations; Morphogen systems; High spatial dimensions

1. Introduction

Integration factor (IF) or exponential differencing time (ETD) methods are popular methods for temporal partial differential equations. In these methods, the linear operators of the highest order derivative are treated exactly. As a result, the stability constraint associated with the highest order derivatives are totally removed, and large time steps can be used. However, the exact treatment of the differential operator requires evaluating exponentials of the approximation matrix for the linear differential operator. For periodic systems, this calculation is cheap both in CPU and storage because the approximation matrix can be diagonalized in the Fourier space [1–7]. For non-periodic systems, in which the approximation matrices are not diagonal, storage and calculation of exponentials of the matrices are significantly more expensive. In two or three spatial dimensions, this computational cost becomes prohibitive for any practical use, consequently neither IF nor ETD methods have been used for non-periodic systems.

*Corresponding author. Tel.: +1 949 8245530; fax: +1 949 8247993. E-mail address: qnie@math.uci.edu (Q. Nie).

To illustrate this, we apply the IF or ETD methods to reaction diffusion equations of this form:

$$\frac{\partial \mathbf{u}}{\partial t} = D\Delta \mathbf{u} + \mathbf{F}(\mathbf{u}), \tag{1}$$

where $\mathbf{u} \in \mathbf{R}^m$ represent a group of physical or biological species, $D \in \mathbf{R}^{m\times m}$ is the diffusion constant matrix, $\Delta\mathbf{u}$ is the Laplacian associated with the diffusion of the species $\mathbf{u}$, and $\mathbf{F}(\mathbf{u})$ describes the chemical or biological reactions. The first step of constructing the IF or ETD schemes is to reduce the (1) to a system of ODEs using method of lines:

$$u_t = \mathcal{C}u + \mathcal{F}(u), \tag{2}$$

where $\mathcal{C}u$ is assumed to be a finite difference approximation of the differential operator $D\Delta\mathbf{u}$. Let n denote the total number of spatial grid points (the sum of the points in every dimension of R^m) for the approximation of the Laplacian $\Delta\mathbf{u}$, then $u(t) \in R^{n\times m}$ and $\mathcal{C}$ representing a spatial discretization of the diffusion is a block matrix with each block of a size $n \times n$. In a one-dimensional system with one diffusion, $\mathcal{C}$ is a tri-diagonal matrix for a second order central difference approximation on the diffusion.

The next step of the construction is to multiply (2) by $e^{\mathcal{C}t}$ and to integrate it in time. Different approximation of the integral involving nonlinear term $\mathcal{F}(u)$ results in either the integration factor (IF) method or the exponential time differencing (ETD) method [8]. For example, the second order integration factor Adams–Bashforth method (IFAB2 [9]) has the form

$$u_{k+1} = e^{C\Delta t}u_k + \Delta t\left(\frac{3}{2}e^{C\Delta t}\mathcal{F}(u_k) - \frac{1}{2}e^{2C\Delta t}\mathcal{F}(u_{k-1})\right); \tag{3}$$

and the second order ETD method [2,9] has a form:

$$u_{k+1} = e^{C\Delta t}u_k + \frac{1}{\Delta t}C^{-2}\{[(I+\Delta tC)e^{C\Delta t} - I - 2\Delta tC]\mathcal{F}(u_k) - [e^{C\Delta t} - I - \Delta tC]\mathcal{F}(u_{k-1})\}. \tag{4}$$

In (3) and (4), $e^{\mathcal{C}\Delta t}$ is a matrix of size $n \times n$ for a system with only one diffusive species, and u_k is the approximate solution at the kth time step. The computational cost for updating u_{k+1} at one time step is of order of n^2 due to the three vector–matrix multiplications associated with $e^{\mathcal{C}\Delta t}$ and $e^{2\mathcal{C}\Delta t}$ in (3). During the temporal updating, these two matrices remain the same for a fixed Δt, and they only need to be evaluated once initially from $\mathcal{C}$ and be stored. All IF or ETD methods require storage of these types of exponential matrices because re-calculating them at each time step is not efficient.

For a system in one spatial dimension, the size of $e^{\mathcal{C}\Delta t}$ usually can be handled [9,5,8]. In two or three spatial dimensions, corresponding to a large n, the required storage of $e^{\mathcal{C}\Delta t}$ may become prohibitive if the one-dimensional approach [9,5,8] is directly applied to the higher dimensional system. For instance, in a three-dimensional system with a moderate number of spatial grid points such as $40 \times 40 \times 40$, it yields $n = 6.4 \times 10^4$. The required storage is $O(n^2) = O(10^9)$; that is not manageable for a typical machine. This bottle-neck limits application of IF and ETD methods for non-periodic systems in two or three spatial dimensions.

In this paper, we reduce the required storage by introducing a compact representation for the matrix approximating the differential operator. The new compact form, in the case of equal spacing in each spatial direction (for simplicity of illustration), involves storage only proportional to the number of unknowns, i.e., the dimension of u, unlike the non-compact approach, which is proportional to the square of the unknowns. For example, in a two-dimensional system of $N \times N$ grid points, the unknown values of u at the grid points are stored as a $N \times N$ matrix analogous to its natural spatial partition [10]. As a result, the exponential of the discretized Laplacians in ETD and IF methods is $N \times N$. The new approach needs only $O(N^2)$ storage and $O(N^3)$ operations, compared to the $O(N^4)$ storage requirement and an $O(N^4)$ operation count in the non-compact approach. In three-dimensions, the improvement for the new approach is even more significant. For a system with $N \times N \times N$ grid points, the new approach needs $O(N^3)$ storage and $O(N^4)$ operations, compared to $O(N^6)$ storage and $O(N^6)$ operations using the non-compact representation.

The compact representation can be easily used in IF and ETD methods without altering the stability properties of these methods. Their implementations are straightforward, and the number of grid points in each spatial direction does not need to be the same. The new technique is tested for simple linear systems as well as nonlinear systems arising from biological applications in both two- and three-dimensions, using a class of semi-implicit integration factor (IIF) method developed recently for systems with stiff reactions [8].

In Section 2, we derive the semi-implicit integration factor methods using the compact representation, along with a stability analysis. This is done for both two- and three-dimensions. In Section 3, we test the new methods on linear systems and a couple of nonlinear models in cell and developmental biology.

2. Compact implicit integration factor (cIIF) in high spatial dimensions

2.1. Two-dimensions

To distinguish the new compact implicit integration factor method from the standard IIF, we denote it as cIIF. In this section, we illustrate the new method by applying the IIF method with the new compact representation to a two-dimensional reaction–diffusion equation with periodic boundary conditions in the x direction and Neumann boundary conditions in the y direction:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} = D\left(\frac{\partial^2 \mathbf{u}}{\partial x^2} + \frac{\partial^2 \mathbf{u}}{\partial y^2}\right) + \mathbf{F}(\mathbf{u}), \quad (x, y) \in \Omega = \{a < x < b, c < y < d\}; \\ \frac{\partial u}{\partial x}(a, y, t) = \frac{\partial u}{\partial x}(b, y, t) = 0; \\ u(x, c, t) = u(x, d, t), \quad \frac{\partial u}{\partial y}(x, c, t) = \frac{\partial u}{\partial y}(x, d, t). \end{cases} \tag{5}$$

We first discretize the spatial domain by a rectangular mesh: $(x_i, y_j) = (a + i \times h_x, c + j \times h_y)$ where $h_x = (b - a)/(N_x + 1)$, $h_y = (d - c)/(N_y + 1)$ and $0 \le i \le N_x + 1$ and $0 \le j \le N_y + 1$. Using the second order central difference discretization on the diffusion, we obtain a system of nonlinear ODEs

$$\frac{\mathrm{d}u_{i,j}}{\mathrm{d}t} = D\left(\frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h_y^2}\right) + \mathbf{F}(u_{i,j}). \tag{6}$$

Next we define three matrices **U**, **A** and **B** by

$$\mathbf{U}=\begin{pmatrix} u_{1,1} & u_{1,2} & \cdots & u_{1,N_y} & u_{1,N_y+1} \\ u_{2,1} & u_{2,2} & \cdots & u_{2,N_y} & u_{2,N_y+1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ u_{N_x,1} & u_{N_x,2} & \cdots & u_{N_x,N_y} & u_{N_x,N_y+1} \end{pmatrix}_{N_x\times(N_y+1)}, \tag{7}$$

$$\mathbf{A}=\frac{D}{h_x^2}\times\begin{pmatrix} -\frac{2}{3} & \frac{2}{3} & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & -2 & 1 \\ & & & & \frac{2}{3} & -\frac{2}{3} \end{pmatrix}_{N_x\times N_x}, \tag{8}$$

and

$$\mathbf{B}=\frac{D}{h_y^2}\times\begin{pmatrix} -2 & 1 & 0 & 0 & \cdots & 1 \\ 1 & -2 & 1 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & \cdots & 0 \\ & & \ddots & \ddots & \ddots & \\ 0 & 0 & \cdots & 1 & -2 & 1 \\ 1 & 0 & \cdots & 0 & 1 & -2 \end{pmatrix}_{(N_y+1)\times(N_y+1)}. \tag{9}$$

In terms of these three matrices, the semi-discretized form (6) becomes

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t}=\mathbf{A}\mathbf{U}+\mathbf{U}\mathbf{B}+\mathcal{F}(\mathbf{U}). \tag{10}$$

This formulation is based on a compact representation previously developed for solving a two-dimensional Poisson's equation and other related separable equations [10].

To apply the integration factor technique to the compact discretization form (10), we multiply (10) by exponential matrix $\mathrm{e}^{-\mathbf{A}t}$ from the left, and $\mathrm{e}^{-\mathbf{B}t}$ from the right to obtain

$$\frac{\mathrm{d}(\mathrm{e}^{-\mathbf{A}t}\mathbf{U}\mathrm{e}^{-\mathbf{B}t})}{\mathrm{d}t}=\mathrm{e}^{-\mathbf{A}t}\mathcal{F}(\mathbf{U})\mathrm{e}^{-\mathbf{B}t}. \tag{11}$$

Integration of (11) over one time step from t_n to $t_{n+1} \equiv t_n + \Delta t$, where Δt is the time step, leads to

$$\mathbf{U}_{n+1}=\mathrm{e}^{\mathbf{A}\Delta t}\mathbf{U}_n\mathrm{e}^{\mathbf{B}\Delta t}+\mathrm{e}^{\mathbf{A}\Delta t}\left(\int_0^{\Delta t}\mathrm{e}^{-\mathbf{A}\tau}\mathcal{F}(\mathbf{U}(t_n+\tau))\mathrm{e}^{-\mathbf{B}\tau}\mathrm{d}\tau\right)\mathrm{e}^{\mathbf{B}\Delta t}. \tag{12}$$

To construct a scheme of rth order truncation error, we approximate the integrand in (12),

$$\mathcal{G}(\tau) \equiv \mathrm{e}^{-\mathbf{A}\tau}\mathcal{F}(\mathbf{U}(t_n+\tau))\mathrm{e}^{-\mathbf{B}\tau}, \tag{13}$$

using a $(r-1)$th order Lagrange polynomial at a set of interpolation points $t_{n+1}, t_n, \ldots, t_{n+2-r}$:

$$\mathcal{P}(\tau) \equiv \sum_{j=-1}^{r-2} \mathrm{e}^{j\mathbf{A}\Delta t}\mathcal{F}(\mathbf{U}_{n-j})\mathrm{e}^{j\mathbf{B}\Delta t} p_j(\tau), \quad 0 \le \tau \le \Delta t, \tag{14}$$

where

$$p_j(\tau) \equiv \prod_{\substack{k=-1 \\ k\neq j}}^{r-2} \frac{\tau+k\Delta t}{(k-j)\Delta t}. \tag{15}$$

The specific form of the polynomial (15) at low orders is listed in Table 1. In terms of $\mathcal{P}(\tau)$, (12) takes the form,

$$\mathbf{U}_{n+1}=\mathrm{e}^{\mathbf{A}\Delta t}\mathbf{U}_n\mathrm{e}^{\mathbf{B}\Delta t}+\mathrm{e}^{\mathbf{A}\Delta t}\left(\int_0^{\Delta t}\mathcal{P}(\tau)\mathrm{d}\tau\right)\mathrm{e}^{\mathbf{B}\Delta t}. \tag{16}$$

So the new r-th order implicit schemes are

$$\mathbf{U}_{n+1}=\mathrm{e}^{\mathbf{A}\Delta t}\mathbf{U}_n\mathrm{e}^{\mathbf{B}\Delta t}+\Delta t\left(\alpha_1\mathcal{F}(U_{n+1})+\sum_{j=0}^{r-2}\alpha_{-j}\mathrm{e}^{(j+1)\mathbf{A}\Delta t}\mathcal{F}(\mathbf{U}_{n-j})\mathrm{e}^{(j+1)\mathbf{B}\Delta t}\right), \tag{17}$$

where $\alpha_1, \alpha_0, \alpha_{-1}, \cdots, \alpha_{-r+2}$ are coefficients calculated from the integrals of the polynomial in $\mathcal{P}(\tau)$,

$$\alpha_{-j}=\frac{1}{\Delta t}\int_0^{\Delta t}\prod_{\substack{k=-1 \\ k\neq j}}^{r-2}\frac{\tau+k\Delta t}{(k-j)\Delta t}\mathrm{d}\tau, \quad -1 \le j \le r-2. \tag{18}$$

In Table 2, the value of coefficients, α_{-j}, for schemes of order up to four are listed.

In particular, the second order approximation of $\int_0^{\Delta t}\mathcal{G}(\tau)\mathrm{d}\tau$

$$\int_0^{\Delta t} \mathcal{G}(\tau)\,\mathrm{d}\tau \approx \frac{\mathcal{F}(\mathbf{U}_n)+\mathrm{e}^{-\mathbf{A}\Delta t}\mathcal{F}(\mathbf{U}_{n+1})\mathrm{e}^{-\mathbf{B}\Delta t}}{2}\Delta t \tag{19}$$

leads to the second order IIF scheme (cIIF2)

$$\mathbf{U}_{n+1}=\mathrm{e}^{\mathbf{A}\Delta t}\left(\mathbf{U_n}+\frac{\Delta t}{2}\mathcal{F}(\mathbf{U_n})\right)\mathrm{e}^{\mathbf{B}\Delta t}+\frac{\Delta t}{2}\mathcal{F}(\mathbf{U_{n+1}}). \tag{20}$$

Like the one-dimensional form [8], the nonlinear reaction term at t_{n+1} in (20) is decoupled from the diffusion terms. As a result, only a local nonlinear system needs to be solved at each spatial grid point. The two matrices $\mathrm{e}^{\mathbf{A}\Delta t}$ and $\mathrm{e}^{\mathbf{B}\Delta t}$ are $N_x \times N_x$ and $(N_y + 1) \times (N_y + 1)$, respectively. Both are orders of magnitude smaller than the size of the matrix, $N_x(N_y + 1) \times N_x(N_y + 1)$, in the non-compact representation. As to be demonstrated in direct numerical simulations in Section 3, this saving in storage is critical for carrying out simulations with even moderate numbers of spatial grid points. Also, the new approach requires fewer operations. In the non-compact approach, a matrix–vector multiplication with operations of the order of $N_x^2(N_y+1)^2$ dominates the computational cost at each time step. In the new approach, the corresponding calculations are two matrix–matrix operations of an order of $N_x(N_y+1)^2+N_x^2(N_y+1)$, which is significantly smaller. Also, because of the smaller size of those matrices, the initial calculations of the exponentials of those matrices become cheaper as well. Therefore, the new method is advantageous in both CPU time and memory savings.

Remark 1—The compact explicit IF (cIF) can be derived in a similar way. For example, the compact form of the IFAB2 (3) takes the form:

$$\mathbf{U}_{n+1}=\mathrm{e}^{\mathbf{A}\Delta t}\mathbf{U_n}\mathrm{e}^{\mathbf{B}\Delta t}+\Delta t\left(\frac{3}{2}\mathrm{e}^{\mathbf{A}\Delta t}\mathcal{F}(\mathbf{U_n})\mathrm{e}^{\mathbf{B}\Delta t}-\frac{1}{2}\mathrm{e}^{2\mathbf{A}\Delta t}\mathcal{F}(\mathbf{U_{n-1}})\mathrm{e}^{2\mathbf{B}\Delta t}\right). \tag{21}$$

Remark 2—The compact semi-discretization system (10) can also be used for other types of methods, such as the ETD methods. In the derivation of ETD schemes based on a non-compact representation, only the $\mathcal{F}$ in the integrand (an integral similar to the one in (12)) is approximated by an interpolation polynomial, with the exponential function unchanged in the integrand; then a direct integration of the approximate integrand leads to the ETD methods [8].

For a compact system with (12), one needs to evaluate

$$\int_0^{\Delta t} \mathrm{e}^{-\mathbf{A}\tau}P(\tau)\mathrm{e}^{-\mathbf{B}\tau}\,\mathrm{d}\tau \tag{22}$$

where $P(\tau)$ is a polynomial matrix, and matrices **A**, **B** have dimensions $N_x \times N_x$, $N_y \times N_y$, respectively. After assuming that

$$P(\tau)=\sum_p \mathbf{C}^{(p)}\tau^p, \tag{23}$$

(22) takes the form of

$$\sum_p \int_0^{\Delta t} \mathrm{e}^{-\mathbf{A}\tau} \mathbf{C}^{(p)} \tau^p \mathrm{e}^{-\mathbf{B}\tau} \mathrm{d}\tau. \tag{24}$$

If the matrices $\mathbf{A}$, $\mathbf{C}^{(p)}$, and $\mathbf{B}$ commute with each other, (24) can be simplified as

$$\sum_p \mathbf{C}^{(p)} \int_0^{\Delta t} \mathrm{e}^{-(\mathbf{A}+\mathbf{B})\tau} \tau^p \mathrm{d}\tau. \tag{25}$$

The integral in (25) can be integrated explicitly through integration by parts, and the matrices $\mathbf{C}^{(p)}$ and $\int_0^{\Delta t} \mathrm{e}^{-(\mathbf{A}+\mathbf{B})\tau} \tau^p \mathrm{d}\tau$ have the same dimension $N = N_x = N_y$. The total operation of evaluating (25) is $\mathrm{O}(N^3)$ due to the matrix–matrix multiplication in (25). The overall computational cost is similar to the cIIF methods discussed above.

If matrices $\mathbf{A}$, $\mathbf{C}^{(p)}$, and $\mathbf{B}$ do not commute, one may need to consider the eigenspace of matrices $\mathbf{A}$ and $\mathbf{B}$ in order to evaluate (22) explicitly [11,12]. Assuming an eigenvalue decomposition $\mathbf{A} = V \,\mathrm{diag}(a_1,\ldots, a_{N_x})V^{-1}$ and $\mathbf{B} = W \,\mathrm{diag}(b_1,\ldots, b_{N_y})W^{-1}$, then the (i, n)th element of matrix (24) is

$$\sum_p \sum_{j,k=1}^{N_x} \sum_{l,m=1}^{N_y} f_{jm} V_{ij} V_{jk}^{-1} C_{kl}^{(p)} W_{lm} W_{mn}^{-1}, \tag{26}$$

where $f_{jm} = \int_0^{\Delta t} \mathrm{e}^{-(a_j+b_m)\tau} \tau^p \mathrm{d}\tau$ can be evaluated recursively through integration by parts. The operation count of (26) is $\mathrm{O}(N_x^2 N_y^2)$, and it leads to an operation count of $\mathrm{O}(N_x^3 N_y^3)$ for generating the whole matrix (24). The cost associated with such an approach is even more expensive than the non-compact approach.

To obtain the ETD type methods that do not require any commutativity properties on the matrices $\mathbf{A}$ and $\mathbf{B}$, and have the same order of operation count as that of the cIIF schemes, one may leave only one of the exponential functions unchanged and apply the polynomial approximation to the rest of the integrand in (12). To illustrate this approach, we first approximate the integrand

$$\mathcal{G}(\tau) \equiv \mathcal{F}(\mathbf{U}(t_n+\tau))\mathrm{e}^{-\mathbf{B}\tau}, \tag{27}$$

using a $r-1$th order Lagrange polynomial at a set of interpolation points $t_{n+1}, t_n, \ldots, t_{n+2-r}$:

$$\mathcal{P}(\tau) \equiv \sum_{j=-1}^{r-2} \mathcal{F}(\mathbf{U}_{n-j}) \mathrm{e}^{j\mathbf{B}\Delta t} \prod_{\substack{k=-1 \\ k \neq j}}^{r-2} \frac{\tau + k\Delta t}{(k-j)\Delta t}. \tag{28}$$

Then (12) becomes

$$\mathbf{U}_{n+1}=e^{\mathbf{A}\Delta t}\mathbf{U}_n e^{\mathbf{B}\Delta t}+e^{\mathbf{A}\Delta t}\left(\int_0^{\Delta t} e^{-\mathbf{A}\tau}\mathcal{P}(\tau)d\tau\right)e^{\mathbf{B}\Delta t}. \tag{29}$$

A first order implicit approximation to $\mathcal{G}(\tau)$ of the form

$$\mathcal{P}(\tau)=\mathcal{F}(\mathbf{U_{n+1}})e^{-\mathbf{B}\Delta t},\ 0\le\tau\le\Delta t, \tag{30}$$

leads to a first order implicit scheme

$$\mathbf{U}_{n+1}=e^{\mathbf{A}\Delta t}\mathbf{U_n}e^{\mathbf{B}\Delta t}+\mathbf{A^{-1}}(e^{\mathbf{A}\Delta t}-I)\mathcal{F}(\mathbf{U_{n+1}}). \tag{31}$$

A second order implicit approximation to $\mathcal{G}(\tau)$,

$$\mathcal{P}(\tau)=\frac{1}{\Delta t}[\mathcal{F}(\mathbf{U}_n)(\Delta t-\tau)+\mathcal{F}(\mathbf{U}_{n+1})e^{-\mathbf{B}\Delta t}\tau],\ 0\le\tau\le\Delta t, \tag{32}$$

leads to a second order implicit scheme

$$\mathbf{U}_{n+1}=e^{\mathbf{A}\Delta t}\mathbf{U_n}e^{\mathbf{B}\Delta t}+\frac{1}{\Delta t}\{[\mathbf{A^{-2}}(I-e^{\mathbf{A}\Delta t})+\Delta t\mathbf{A^{-1}}e^{\mathbf{A}\Delta t}]\mathcal{F}(\mathbf{U_n})e^{\mathbf{B}\Delta t}+[\mathbf{A^{-2}}(e^{\mathbf{A}\Delta t}-I)-\Delta t\mathbf{A^{-1}}]\mathcal{F}(\mathbf{U_{n+1}})\}. \tag{33}$$

Like the implicit ETD methods based on the non-compact representation, the nonlinear function of $\mathbf{U}_{n+1}$ in the compact implicit ETD (33) is also multiplied by terms involving the approximated differential operators and their exponentials. This non-local coupling makes the implicit ETD method inefficient. In contrast, in IIF [8] and cIIF, the diffusion term and nonlinear reaction term are decoupled. This makes IIF more desirable.

The compact explicit ETD (cETD) methods can be derived similarly. For example, the compact form of the second order ETD method (4) becomes

$$\mathbf{U}_{n+1}=e^{\mathbf{A}\Delta t}\mathbf{U_n}e^{\mathbf{B}\Delta t}+\frac{1}{\Delta t}\mathbf{A^{-2}}\{[(I+\Delta tA)e^{\mathbf{A}\Delta t}-I-2\Delta tA]\mathcal{F}(\mathbf{U_n})e^{\mathbf{B}\Delta t}-[e^{\mathbf{A}\Delta t}-I-\Delta tA]\mathcal{F}(\mathbf{U_{n-1}})e^{2\mathbf{B}\Delta t}\}. \tag{34}$$

2.2. Stability analysis of cIIF methods

The linear stability of the high-dimensional cIIF methods can be analyzed by an approach similar to that for the one-dimensional system [2,13,8]. We test the linear stability on the the following linear equation

$$u_t=-q_1u-q_2u+\mathrm{d}u\ \text{with}\ q_1,q_2>0, \tag{35}$$

where q_1 and q_2 represent diffusions in the x and y directions, respectively. The boundaries of the stability region, a family of curves for different values of $(q_1 + q_2)\,\Delta t$, based on the test problem (35) are presented for the second and third order implicit integration factor methods. The quantity $(q_1 + q_2)\,\Delta t$ involves the ratio between the time step and the spatial grid for the discretization of the reaction–diffusion Eq. (1).

To obtain the stability region, we apply cIIF2 (20) to the Eq. (35), then substitute $u_n = e^{in\theta}$ into the resulted equation. This leads to

$$e^{i\theta}=e^{-q_1\Delta t}\left(1+\frac{1}{2}\lambda\right)e^{-q_2\Delta t}+\frac{1}{2}\lambda e^{i\theta}, \tag{36}$$

where $\lambda = d\Delta t$. The equations for λ_r, the real part of λ, and λ_i, the imaginary part of λ, become

$$\begin{aligned}\lambda_r&=\frac{2(1-e^{-2(q_1+q_2)\Delta t})}{(1-e^{-(q_1+q_2)\Delta t})^2+2(1+\cos\theta)e^{-(q_1+q_2)\Delta t}},\\ \lambda_i&=\frac{4(\sin\theta)e^{-(q_1+q_2)\Delta t}}{(1-e^{-(q_1+q_2)\Delta t})^2+2(1+\cos\theta)e^{-(q_1+q_2)\Delta t}}.\end{aligned} \tag{37}$$

Since $q_1 + q_2 > 0$, we have $\lambda_r > 0$ for $0 \le \theta \le 2\pi$. Therefore, the stability region is in the left half complex plane. This implies that the second order IIF is *A*–stable. In Fig. 1, the stability region of the method is plotted for $(q_1 + q_2)\,\Delta t = 0.5, 1, 2$. The exterior of the closed curves located on the complex plane at $\lambda_r > 0$ is the stability region.

When $q_1 + q_2 \to 0$, the stability region will coincide with the domain $\lambda_r < 0$; and when $q_1 + q_2 \to \infty$, the stability region becomes the entire complex plane excluding the point (2, 0).

For the third order two-dimensional cIIF scheme:

$$u_{n+1}=e^{A\Delta t}u_n e^{B\Delta t}+\Delta t\left(\frac{5}{12}\mathcal{F}(u_{n+1})+\frac{2}{3}e^{A\Delta t}\mathcal{F}(u_n)e^{B\Delta t}-\frac{1}{12}e^{2A\Delta t}\mathcal{F}(u_{n-1})e^{2B\Delta t}\right). \tag{38}$$

We can perform a similar analysis for the stability to obtain λ

$$\lambda=\frac{e^{i\theta}-e^{-(q_1+q_2)\Delta t}}{\frac{5}{12}e^{i\theta}+\frac{2}{3}e^{-(q_1+q_2)\Delta t}-\frac{1}{12}e^{-2(q_1+q_2)\Delta t-i\theta}}. \tag{39}$$

As seen in Fig. 2 for $(q_1 + q_2)\,\Delta t = 0, 0.45, 0.5, 0.6, 1.0$, the third order scheme is not A-stable. Similar to the one-dimensional case, the stability region sensitively depends on the value of $(q_1 + q_2)\,\Delta t$. The size of the region is an increasing function of $(q_1 + q_2)\,\Delta t$. For $(q_1 + q_2)\,\Delta t < 0.54$, the stability region is in the left half of the complex plan λ bounded by a closed curve. For $(q_1 + q_2)\,\Delta t > 0.55$, the stability region contains the entire left half plane and most of the right half plane. When $q_1 + q_2 \to \infty$, the stability region becomes the entire complex plane excluding one point on the real axis.

The stability analysis for cETD schemes is similar to the stability analysis of ETD [2,14,6,7, 9], and the stability analysis of cIIF presented above.

2.3. Three-dimensions

The compact representation of the Laplacian operator like (10) for two-dimensional systems can be extended to higher dimensional systems. In this section, we present a derivation for a three-dimensional reaction–diffusion equation in a cube with no-flux boundary conditions:

$$\begin{cases} \frac{\partial \mathbf{u}}{\partial t} = D\Delta u + \mathbf{F}(\mathbf{u}), & (x,y,z) \in \Omega = \{a_l < x < a_u, b_l < y < b_u, c_l < z < c_u\}; \\ \mathbf{n} \cdot \nabla u = 0, & (x,y,z) \in \partial\Omega \end{cases} \tag{40}$$

where $\mathbf{n}$ is the unit outward normal direction of $\partial\Omega$.

Let N_x, N_y, N_z denote the number of spatial grid points in x, y, z-direction, respectively, h_x, h_y, h_z be the grid size, and $u_{i,j,k}$ represents the approximate solution at the grid (x_i, y_j, z_k). A second order central difference discretization on the Laplacian operator yields

$$\frac{\mathrm{d}u_{i,j,k}}{\mathrm{d}t} = D\left(\frac{u_{i+1,j,k} - 2u_{i,j,k} + u_{i-1,j,k}}{h_x^2} + \frac{u_{i,j-1,k} - 2u_{i,j,k} + u_{i,j+1,k}}{h_y^2} + \frac{u_{i,j,k-1} - 2u_{i,j,k} + u_{i,j,k+1}}{h_z^2}\right) + \mathcal{F}(u_{i,j,k}). \tag{41}$$

Define $A_x = \frac{D}{h_x^2} A_{N_x \times N_x}$, $A_y = \frac{D}{h_y^2} A_{N_y \times N_y}$, and $A_z = \frac{D}{h_z^2} A_{N_z \times N_z}$, where

$$A_{P\times P} = \begin{pmatrix} -2 & 2 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & 2 & -2 \end{pmatrix}_{P\times P}. \tag{42}$$

Then (41) has the following compact representation

$$\mathbf{U}_t = \left(\sum_{l=1}^{N_x} (A_x)_{i,l} u_{l,j,k} + \sum_{l=1}^{N_y} (A_y)_{j,l} u_{i,l,k} + \sum_{l=1}^{N_z} (A_z)_{k,l} u_{i,j,l}\right) + \mathcal{F}(\mathbf{U}) \tag{43}$$

where $\mathbf{U} = (u_{i,j,k})$ and $\mathcal{F}(\mathbf{U}) = (\mathcal{F}(u_{i,j,k}))$. The three summation terms in (43) are similar to the two vector–matrix multiplications in the two-dimensional case in (10). In addition to a left multiplication and a right multiplication in (10), there is a ‘middle’ multiplication in (43).

Define an operator $\mathcal{L}(t)$ by

$$\mathcal{L}(t)\mathbf{U} = \left(\sum_{n=1}^{N_z}\sum_{m=1}^{N_y}\sum_{l=1}^{N_x} (\mathrm{e}^{-A_z t})_{k,n} (\mathrm{e}^{-A_y t})_{j,m} (\mathrm{e}^{-A_x t})_{i,l} u_{l,m,n}\right). \tag{44}$$

Taking derivatives of (44) yields

$$\frac{\mathrm{d}(\mathcal{L}(t)\mathbf{U})}{\mathrm{d}t}=\mathcal{L}(t)\left(\mathbf{U}_t-\left(\sum_{l=1}^{N_x}(A_x)_{i,l}u_{l,j,k}+\sum_{l=1}^{N_y}(A_y)_{j,l}u_{i,l,k}+\sum_{l=1}^{N_z}(A_z)_{k,l}u_{i,j,l}\right)\right). \tag{45}$$

Letting $\mathcal{L}(t)$ act on both sides of (43), and using (45), we obtain

$$\frac{\mathrm{d}(\mathcal{L}(t)\mathbf{U})}{\mathrm{d}t}=\mathcal{L}(t)\mathcal{F}(\mathbf{U}). \tag{46}$$

Integrating (46) over one time step from t_n to t_{n+1}, and using a transformation $s = t_n + \tau$ for the integration, we obtain

$$\mathcal{L}(t_{n+1})\mathbf{U}_{n+1}=\mathcal{L}(t_n)\mathbf{U}_n+\mathcal{L}(t_n)\int_0^{\Delta t}\mathcal{L}(\tau)\mathcal{F}(\mathbf{U}(t_n+\tau))\mathrm{d}\tau. \tag{47}$$

Applying $\mathcal{L}(-t_{n+1})$ on both sides of (47) yields

$$\mathbf{U}_{n+1}=\mathcal{L}(-\Delta t)\mathbf{U}_n+\mathcal{L}(-\Delta t)\left(\int_0^{\Delta t}\mathcal{G}(\tau)\mathrm{d}\tau\right), \tag{48}$$

where

$$\mathcal{G}(\tau)=\mathcal{L}(\tau)\mathcal{F}(\mathbf{U}(t_n+\tau)). \tag{49}$$

To derive (48), we've used two identities:

$$\mathcal{L}(-rt)\mathcal{L}(rt)\mathbf{U}=\mathbf{U} \tag{50}$$

and

$$\mathcal{L}(-rt)\mathcal{L}(st)\mathbf{U}=\mathcal{L}((s-r)t)\mathbf{U} \tag{51}$$

for any two scalars r and s. Both (50) and (51) can be easily proved based on the definition of $\mathcal{L}$.

Similar to the construction for the two-dimensional system, the approximation of $\mathcal{G}(\tau)$ using a $r-1$th order Lagrange polynomial results in a scheme with truncation error of rth order. Specifically, a second order approximation,

$$\int_0^{\Delta t}\mathcal{G}(\tau)\mathrm{d}\tau\approx\frac{\mathcal{F}(\mathbf{U}_n)+\mathcal{L}(\Delta t)\mathcal{F}(\mathbf{U}_{n+1})}{2}\Delta t,$$

leads to the second order IIF (cIIF2) method for a three-dimensional system:

$$\mathbf{U}_{n+1}=\mathcal{L}(-\Delta t)\left(\mathbf{U}_n+\frac{\Delta t}{2}\mathcal{F}(\mathbf{U}_n)\right)+\frac{\Delta t}{2}\mathcal{F}(\mathbf{U}_{n+1}). \tag{52}$$

The scheme (52) has a form similar to the one- and two-dimensional case. The evaluation of the nonlinear term $\mathcal{F}$ at t_{n+1} is still local and decoupled from the global diffusion term such that a nonlinear system of the size of $\mathcal{F}$ needs to be solved at each spatial grid point.

To evaluate $\mathcal{L}(-\Delta t)$ in (52), three square matrices $e^{A_x\Delta t}$, $e^{A_y\Delta t}$, and $e^{A_z\Delta t}$ have to be pre-calculated and stored. The size of the three matrices is N_x^2, N_y^2, and N_z^2, respectively. The size of $\mathbf{U}$ is of order $N_xN_yN_z$. In a non-compact representation, the matrix which needs to be stored has a size of the order of $N_x^2N_y^2N_z^2$. Clearly, the storage requirement for the new approach is smaller by orders of magnitude. Even for moderate N_x, N_y and N_z, the storage requirement for the non-compact representation usually becomes prohibitive even for computers with large memory, as seen in the numerical examples in Section 3. In contrast, the new approach can easily handle the same system with even higher spatial resolutions.

In addition, the new approach takes considerably fewer CPU operations. The operation count for evaluating $\mathcal{L}(-\Delta t)\mathbf{U}$ is of the order of $N_x^2N_yN_z+N_xN_y^2N_z+N_xN_yN_z^2$. This is much smaller than $N_x^2N_y^2N_z^2$, the operation count for the corresponding matrix–vector multiplication in a non-compact representation. The savings in CPU and memory for the new approach are more significant as N_x, N_y, N_z becomes larger.

Remark—(43) can be re-written in the following form

$$\mathbf{U}_t = A_x\textcircled{x}\mathbf{U} + A_y\textcircled{y}\mathbf{U} + A_z\textcircled{z}\mathbf{U} + \mathcal{F}(\mathbf{U}) \tag{53}$$

by defining three operators,

$$(A_x\textcircled{x}\mathbf{U})_{i,j,k} = \sum_{l=1}^{N_x}(A_x)_{i,l}u_{l,j,k}, \tag{54}$$

$$(A_y \textcircled{y} \mathbf{U})_{i,j,k} = \sum_{l=1}^{N_y} (A_y)_{j,l} u_{i,l,k},$$

(55)

and

$$(A_z \textcircled{z} \mathbf{U})_{i,j,k} = \sum_{l=1}^{N_z} (A_z)_{k,l} u_{i,j,l}.$$

(56)

As a result, the Eq. (44) for $\mathcal{L}(t)$ becomes

$$\mathcal{L}(t)\mathbf{U} = \mathrm{e}^{A_z \Delta t} \textcircled{z} \mathrm{e}^{A_y \Delta t} \textcircled{y} \mathrm{e}^{A_x \Delta t} \textcircled{x} \mathbf{U},$$

(57)

and cIIF2 (52) becomes

$$\mathbf{U}_{n+1} = \mathrm{e}^{A_z \Delta t} \textcircled{z} \mathrm{e}^{A_y \Delta t} \textcircled{y} \mathrm{e}^{A_x \Delta t} \textcircled{x} \left(\mathbf{U}_n + \frac{\Delta t}{2} \mathcal{F}(\mathbf{U}_n) \right) + \frac{\Delta t}{2} \mathcal{F}(\mathbf{U}_{n+1})$$

(58)

which has a form similar to its two-dimensional counterpart (20).

One can also easily obtain other types of cIF and cETD schemes similar to the two-dimensional case. For example, the second order implicit cETD takes the form:

$$\begin{aligned} \mathbf{U}_{n+1} = \mathcal{L}(-\Delta t)\mathbf{U}_n + \frac{1}{\Delta t} \Big\{ \left[A_x^{-2} (I - \mathrm{e}^{A_x \Delta t}) + \Delta t A_x^{-1} \mathrm{e}^{A_x \Delta t} \right] \textcircled{x} \mathrm{e}^{A_y \Delta t} \textcircled{y} \mathrm{e}^{A_z \Delta t} \textcircled{z} \mathcal{F}(\mathbf{U}_n) \\ + \left[A_x^{-2} (\mathrm{e}^{A_x \Delta t} - I) - \Delta t A_x^{-1} \right] \textcircled{x} \mathcal{F}(\mathbf{U}_{n+1}) \Big\}, \end{aligned}$$

(59)

and the second order explicit cETD takes the form:

$$\mathbf{U}_{n+1} = \mathcal{L}(-\Delta t)\mathbf{U}_n + \frac{1}{\Delta t}\left\{\left[A_x^{-2}\left((I+\Delta t A_x)e^{A_x\Delta t} - I - 2\Delta t A_x\right)\right]\textcircled{x}e^{A_y\Delta t}\textcircled{y}e^{A_z\Delta t}\textcircled{z}\mathcal{F}(\mathbf{U}_n) - \left[A_x^{-2}\left(e^{A_x\Delta t} - I - \Delta t A_x\right)\right]\textcircled{x}e^{2A_y\Delta t}\textcircled{y}e^{2A_z\Delta t}\textcircled{z}\mathcal{F}(\mathbf{U}_{n-1})\right\}. \tag{60}$$

3. Numerical simulations

To study the efficiency and accuracy of the new approach for the IF methods, we will implement and test the second order implicit integration factor method using the new approach (cIIF2). We will compare it with IIF2 [8] and with a regular second order Runge–Kutta method (RK2). In addition to testing them on linear systems in two- and three-dimensions, we will also demonstrate the efficiency of cIIF2 by applying it to two reaction–diffusion systems arising from models in developmental and cell biology.

In the calculation, the exponential of the square matrix is computed using a scaling and squaring algorithm with a Pade approximation as implemented in “expm” of Matlab similar to the one dimensional case [8].

Because the matrix exponentials depend only on the spatial grid size, the time step, and diffusion coefficients, during the entire temporal updating they only need to be calculated once initially for a fixed numerical resolution. The local nonlinear systems resulting from IIF2 and cIIF2 are solved using a fixed point iteration procedure similar to that used in the one-dimensional case [8].

3.1. Tests on simple systems

3.1.1. A linear problem in two-dimensions—We consider a linear reaction–diffusion equation

$$\begin{cases} \frac{\partial u}{\partial t} = 0.2\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + 0.1u, \quad (x,y) \in \Omega = \{0<x<2\pi, 0<y<2\pi\}; \\ \frac{\partial u}{\partial x}(0,y,t) = \frac{\partial u}{\partial x}(2\pi,y,t) = 0; \\ u(x,0,t) = u(x,z\pi,t) = 0; \\ u(x,y,0) = \cos(x) + \sin(y). \end{cases} \tag{61}$$

The exact solution of the system is

$$u(x,y,t) = e^{-0.1t}(\cos(x) + \sin(y)). \tag{62}$$

Because of the simple structure of the cIIF2 scheme, it can be easily implemented using MATLAB. The simulation is carried up to $t = 1$ at which the L^∞ difference between the numerical solution and the exact solution is measured. For the convenience of comparison between IIF2 and cIIF3, we also set $h_x = h_y$ for this case.

As seen in Table 3, the IIF2 method on a workstation with 1GB-RMB runs out of memory when $N = 80$ because IIF2 needs to store matrices with a size of $N^2 \times N^2$. In contrast, cIIF2 implemented on the same machine can handle much larger N. For smaller N such as $N = 40$,

although the machine has enough memory for IIF2, it needs almost 2000 times more CPU time to achieve the same accuracy as cIIF2. On the other hand, RK2 can run because of its small memory requirement, but its stability constraint (Δt must be proportional to h_x^2) demands a much smaller time step, and consequently results in more CPU time than cIIF2 for the same accuracy. Overall, cIIF2 is more efficient than both IIF2 and RK2.

3.1.2. A linear problem in three-dimensions—In three-dimensions, we consider a similar system

$$\begin{cases} \frac{\partial u}{\partial t} = d\Delta u + au, & (x, y, z) \in \Omega, \\ \mathbf{n} \cdot \nabla u = 0 & (x, y, z) \in \partial\Omega, \end{cases} \tag{63}$$

where $\Omega = \{0 < x < \pi, 0 < y < \pi, 0 < z < \pi\}$, $\mathbf{n}$ is outward normal of $\partial\Omega$, and $d = 0.2$, $a = 0.1$. The exact solution of (63) has a form similar to (62). The initial condition in the simulations is taken from the exact solution of (63) at $t = 0$. The computation is carried up to $t = 2$ at which the error is measured. We also chose $h_x = h_y = h_z$ for convenience of comparisons with other methods.

Similar to the two-dimensional case, the machine quickly runs out of memory for the IIF2 in three-dimensions when $N > 15$. For three dimensional systems, the required memory for IIF2 is so large that IIF2 is practically impossible to handle any moderate spatial resolutions.

When cIIF2 is compared to RK2 which needs much less memory, cIIF2 shows superiority in CPU times as seen in Table 4. As expected, RK2 does not converge if Δt is set to the same value as used in cIIF2 for most values of N. Because of the sever stability constraint on Δt, RK2 requires a much smaller time-step and becomes more expensive. As shown in Table 4, cIIF2 requires less CPU time than RK2 but achieves the same accuracy.

3.2. Applications to two models in biology

Many models in developmental and cell biology take the form of reaction–diffusion Eq. (1). In such systems, the rate constants in biochemical reactions in **F** usually vary by more than five orders of magnitude. As demonstrated in one-dimensional systems [8], a standard IF, ETD or RK method is not efficient, and the implicit integration factor method (IIF) is much more desirable for such applications with stiff reactions. In this section, we apply cIIF2 to two different models for the study of embryonic patterning and cell signaling, one in two-dimensions and one in three-dimensions.

3.2.1. A two-dimensional model for dorsal-ventral patterning—For proper functioning of tissues, organs and embryos, each cell is required to differentiate appropriately for its position. Positional information that instructs cells about their prospective fate is often conveyed by concentration gradients of morphogens bound to cell signaling receptors. Morphogens are signaling molecules that, when bound to cell receptors, assign different cell fates at different concentrations [15,16]. This role of morphogens has been the prevailing thought in tissue patterning for over half a century; but only recently have there been sufficient experimental data and adequate modeling for us to begin to understand how various morphogens interact and patterns emerge [17–19].

One example is the dorsal-ventral patterning in *Drosophila* embryos, a well-known regulatory system involving several zygotic genes. Among them, decapentaplegic (Dpp)promotes dorsal cell fates such as amnioserosa and inhibits development of the ventral central nervous system; and another gene Sog promotes central nervous system development. In this system, Dpp is

produced only in the dorsal region while Sog is produced only in the ventral region. For the wild-type, the Dpp activity has a sharp peak around the mid-line of the dorsal with the presence of its "inhibitor" Sog. Intriguingly, mutation of Sog results in a loss of ventral structure as expected, but, in addition, the amnioserosa is reduced as well. It appears that the Dpp antagonist, Sog, is required for maximal Dpp signaling [20–23]. In [24–26], simulations and analysis for a simplified one-dimensional dynamic Dpp–Sog model were carried out along with experimental studies. The robustness and temporal dynamics of the morphogens were investigated under various genetic mutations [24–26].

Recently, motivated by experimental study of over-expression of the receptors along the anterior-posterior axis of the embryo [26], a two-dimensional model was developed [27] to examine the Dpp activities outside the area of elevated receptors in a *Drosophila* embryo. In this paper, we apply the cIIF2 to obtain accurate numerical solutions for this two-dimensional system [27].

Let $[L]$, $[S]$, $[LS]$, $[LR]$ denote the concentration of Dpp, Sog, Dpp–Sog complexes, and Dpp–receptor complex, respectively. In the model formulated in [25,27], the dynamics of the Dpp–Sog system is governed by the following reaction diffusion equations:

$$\begin{aligned}
\frac{\partial[L]}{\partial T} &= D_L\left(\frac{\partial^2[L]}{\partial X^2}+\frac{\partial^2[L]}{\partial Y^2}\right) - k_{\text{on}}[L](R(X,Y)-[LR])+k_{\text{off}}[LR] \\
&\quad - j_{\text{on}}[L][S]+(j_{\text{off}}+\tau j_{\text{deg}})[LS]+V_L(X,Y) \\
\frac{\partial[LR]}{\partial T} &= k_{\text{on}}[L](R(X,Y)-[LR]) - (k_{\text{off}}+k_{\text{deg}})[LR] \\
\frac{\partial[LS]}{\partial T} &= D_{LS}\left(\frac{\partial^2[LS]}{\partial X^2}+\frac{\partial^2[LS]}{\partial Y^2}\right)+j_{\text{on}}[L][S] - (j_{\text{off}}+j_{\text{deg}})[LS] \\
\frac{\partial[S]}{\partial T} &= D_S\left(\frac{\partial^2[S]}{\partial X^2}+\frac{\partial^2[S]}{\partial Y^2}\right) - j_{\text{on}}[L][S]+j_{\text{off}}[LS]+VS(X,Y)
\end{aligned} \tag{64}$$

in the domain $0 < X < X_{\max}$, $0 < Y < Y_{\max}$, where

$$R(X,Y)=\begin{cases} R_h, & X \le X_h, \\ R_0, & X>X_h. \end{cases} \tag{65}$$

$$V_L(X,Y)=\begin{cases} \upsilon_L, & Y<\frac{1}{2}Y_{\max}, \\ 0, & Y \ge \frac{1}{2}Y_{\max}. \end{cases} \tag{66}$$

$$V_S(X,Y)=\begin{cases} 0, & Y<\frac{1}{2}Y_{\max}, \\ \upsilon_S, & Y \ge \frac{1}{2}Y_{\max}. \end{cases} \tag{67}$$

The boundary conditions for $[L]$, $[LS]$, and $[S]$ are no-flux at $X = 0$ and $X = X_{\max}$, and periodic at $Y = 0$ and $Y = Y_{\max}$. $R(X, Y)$ is the concentration of the initially available receptor in space; $X = X_h$ is the boundary between the two regions with different level of receptors; $V_L(X, Y)$ and $V_S(X, Y)$ are the production rates for Dpp and Sog, respectively; D_L, D_{LS}, D_S are diffusion

coefficients; τ is the cleavage rate for Sog; and other coefficients are on, off and degradation rate constants for the corresponding bio-chemical reactions.

The initial concentrations of all morphogen molecules are zeros. Both $X_{\max}$ and $Y_{\max}$ are taken to be 0.055 cm, based on the *Drosophila* embryo size at its appropriate developmental stage [26].

To study the performance and convergence of cIIF2, we list in Table 5 the error, order of accuracy and CPU time for simulations using cIIF2 to solve (64) for the set of parameters presented in Fig. 3 without the receptor over-expression. In this case, the spatial resolution is fixed as $N = 40$ in both directions. The error at Δt is measured as a difference between this solution, $u_{\Delta t}$, and the solution $u_{2\Delta t}$ for time step size $2\Delta t$ at $T = 10$, i.e.,

$$E_{\Delta t} = \| u_{\Delta t} - u_{2\Delta t} \|_{L^\infty}. \tag{68}$$

The cIIF2 clearly shows a second order of accuracy in time as expected. As demonstrated in Table 3, the IIF2 for this case will be much slower than cIIF2 for small N, and it runs out of memory for $N > 40$.

Next we study the over-expression experiments in [26] by setting $R_h = 9$ μM in the region $0 < X < X_h = 0.02$ cm [26]. The concentrations of Dpp, Dpp–receptor, Dpp–Sog and Sog are plotted in Fig. 3. It is worth of noting that in the simulations the over-expression of receptor induces a local boost of Dpp–receptor activities near the boundary of two different concentration regions of receptors, similar to the experimental observations [26]. This two-dimensional spatial effect was not modeled in the previous study [26]. A more systematic study on the receptor over-expression will be reported in [27].

3.2.2. A three-dimensional model for intra-cellular signaling—When a hormone or growth factor binds to a cell-surface receptor, a cascade of proteins inside the cell relays the signal to specific intra-cellular targets. A class of proteins referred to as scaffolds are thought to play many important roles during this process [28–30]. Scaffold usually binds dynamically to two or more consecutively-acting components of a signaling cascade. Experimental work suggests that scaffolds may promote signal transmission by tethering consecutively acting kinases near each other [31,32]. However, it has also been experimentally observed that some scaffold inhibit signaling when over-expressed [33–35]. In support of these observations, computations of non-spatial models have demonstrated that scaffold proteins may either enhance or suppress signaling, depending on the concentration of scaffold. In [36], a model of generic, spatially localized scaffold protein was developed for one and two spatial dimensions, and the model indicated that a scaffold protein could boost signaling locally (in and near the region where it was localized) while simultaneously suppressing signaling at a distance.

In this paper, we present simulations for the set of reaction–diffusion equations formulated in [36] that describes a spatially localized scaffold and freely diffusing products and reactants in three dimensions. The model contains a scaffold protein (S), which can bind to two other proteins (A and B). In the absence of the scaffold protein, A and B can bind directly to each other. In the presence of the scaffold protein S, first A binds to S, forming AS. Next B binds to AS forming ASB. Finally, A and B bind to each other on the scaffold and an AB complex is released. The symmetrical path, where B binds to the scaffold before A, is also available. Denote [ ] as the concentration of the proteins, the mass reaction equations with diffusion take the form,

$$\begin{aligned}
\frac{d[S]}{dt} &= -j_{on}([A][S]+[B][S])+j_{off}([AS]+[BS])+j_{con}[ABS],\\
\frac{d[AS]}{dt} &= j_{on}([A][S]-[AS][B])-j_{off}([AS]-[ABS]),\\
\frac{d[BS]}{dt} &= j_{on}([B][S]-[BS][A])-j_{off}([BS]-[ABS]),\\
\frac{d[ABS]}{dt} &= j_{on}([AS][B]+[BS][A])-(2j_{off}+j_{con})[ABS],\\
\frac{d[A]}{dt} &= D\Delta[A]-k_{on}[A][B]+k_{off}[AB]-j_{on}([A][S]+[BS][A])+j_{off}([AS]+[ABS]),\\
\frac{d[B]}{dt} &= D\Delta[B]-k_{on}[A][B]+k_{off}[AB]-j_{on}([B][S]+[AS][B])+j_{off}([BS]+[ABS]),\\
\frac{d[AB]}{dt} &= D\Delta[AB]+k_{on}[A][B]-k_{off}[AB]+j_{con}[ABS].
\end{aligned} \tag{69}$$

In the system (69), D is the diffusion constant; k_{on}, k_{off} are the on and off rates for the off-scaffold reactions, j_{on}, j_{off}, j_{con} are the rate constants for the on-scaffold reactions. The system (69) holds in the cell: $\Omega = \{0 \le x \le 10\ \mu m, 0 \le y \le 10\ \mu m, 0 \le z \le 10\ \mu m\}$, with no-flux boundary conditions for *A*, *B*, *AB*.

First, we test the convergence of cIIF2 when it is applied to the system (69). In this simulation, the initial concentrations of *A* and *B* are set at 1 μM, and they are uniformly distributed throughout the cell. And the scaffolds initially are localized in part of the cell: $4\ \mu m \le x^2 + y^2 + z^2 \le 9\ \mu m$, with [*S*] = 50 μM in this region. The diffusion and rate constants are chosen to be $D = 1\ \mu m^2\ s^{-1}$, $k_{on} = 0.1\ (\mu Ms)^{-1}$, $k_{off} = 0.3\ s^{-1}$, $j_{on} = 1\ (\mu Ms)^{-1}$, $j_{off} = 0.005\ s^{-1}$, and $j_{con} = 0.1\ (\mu Ms)^{-1}$. In Table 6, the error and order of accuracy are estimated at $T = 1$ second using a spatial resolution $N = 40$ in three-directions. As expected, the cIIF2 converges in second order in time, and it has excellent efficiency. And for the IIF, the machine runs out of memory for this spatial resolution: $N = 40$.

Next, we present a case study on the effect of scaffolds in Fig. 4. Due to the symmetry of chemical reaction pathways between *A* and *B*, we only need to show four different products. In this simulation, the initial distribution of each protein and the scaffold are the same as in Table 6. The concentration of each component is represented using density of dots: more dots represent more proteins.

Compared to the case without scaffolds but with other reaction rates being the same, the desired product *AB*, in the case of Fig. 4, is more concentrated in the region where scaffolds are initially distributed, and it is suppressing away from the scaffold region in the meantime. This unevenly distributed *AB* results from an intimate interaction between reactions and diffusions. It is similar to the corresponding one- or two-dimensional systems studied in [36], in which a detailed analysis has been carried out on the condition under which the boost and the suppressing of *AB* simultaneously occur. Although the qualitative features of the system remain the same in different spatial dimensions, we have observed the expected quantitative differences arising in these systems.

4. Conclusions and discussions

In integration factor (IF) and exponential time differencing (ETD) methods, the linear operator with the highest order spatial derivatives in the differential equation is treated exactly in time discretization. This temporal integration involving exponentials of the differential operator leads to unconditional stability associated with that term; however, the computational cost resulting from the approximation usually is very expensive for systems with general boundary conditions, and often it becomes prohibitive in two- or three-dimensions.

In this paper, we introduced a compact representation of the linear differential operator in two- and three-dimensions. Such a representation in IF and ETD methods reduces the computational cost significantly in both storage and CPUs, and it makes IF and ETD in two- and three-

dimensions efficient and attractive methods. We analyzed and implemented such an approach for an implicit integration factor (IIF) method for stiff reaction–diffusion equations. The new compact IIF (cIIF) preserves the stability property of the IIF; and our direct simulations on linear and nonlinear systems in both two- and three-dimensions demonstrated that cIIF is much more efficient than the IIF.

Although we only implemented the new compact approach for reaction–diffusion equations, this technique may be applied to other type of systems, such as equations involving higher order derivatives. Also, the tensor-like representation of the linear differential operators presented in the remark of Section 2.3 can easily be extended to systems in dimensions higher than three. In addition, its excellent stability condition (assuring unconditional linear stability with respect to both diffusions and reactions) along with its compact structure and CPU efficiency make cIIF particularly suitable and useful for spatially adaptive methods. Currently, we are incorporating cIIF with AMR (Adaptive Mesh Refinement) in two- and three-dimensions, and good performance has been observed [37].

Acknowledgments

The research work was partially supported by the NSF DMS-0511169 Grant, the NIH/NSF initiative on Mathematical Biology through Grants R01GM57309 and R01GM67247 from the National Institute of General Medical Sciences, and the NIH P50GM76516 Grant. We also appreciate very much the referee's several valuable suggestions which helped to improve our original manuscript.

References

1. Hou TY, Lowengrub JS, Shelley MJ. Removing the stiffness from interfacial flows with surface tension. Journal of Computational Physics 1994;114:312.
2. Beylkin G, Keiser JM, Vozovoi L. A new class of time discretization schemes for the solution of nonlinear PDEs. Journal of Computational Physics 1998;147:362–387.
3. Leo PH, Lowengrub JS, Nie Q. Microstructural evolution in orthotropic elastic media. Journal of Computational Physics 2000;157:44–88.
4. Jou HJ, Leo PH, Lowengrub JS. Microstructural evolution in inhomogeneous elastic media. Journal of Computational Physics 1997;131:109.
5. Kassam AK, Trefethen LN. Fourth-order time stepping for stiff PDEs. SIAM Journal on Scientific Computing 2005;26:1214–1233.
6. Du Q, Zhu W. Stability analysis and applications of the exponential time differencing schemes. Journal of Computational Mathematics 2004;22:200.
7. Du Q, Zhu W. Analysis and applications of the exponential time differencing schemes and their contour integration modifications, BIT. Numerische Mathematik 2005;45:307–328.
8. Nie Q, Zhang YT, Zhao R. Efficient semi-implicit schemes for stiff systems. Journal of Computational Physics 2006;214:521–537.
9. Cox SM, Matthews PC. Exponential time differencing for stiff systems. Journal of Computational Physics 2002;176:430–455.
10. Wan FYM. An in-core finite difference method for separable boundary value problems on a rectangle. Studies in Applied Mathematics 1973;52:103–113.
11. Friesner RA, Tuckerman LS, Dornblaser BC, Russo TV. A method for exponential propagation of large systems of stiff nonlinear differential equations. Journal of Scientific Computing 1989;4:327–354.
12. Edwards WS, Tuckerman LS, Friesner RA, Sorensen DC. Krylov methods for the incompressible Navier–Stokes equations. Journal of Computational Physics 1994;110:82–102.
13. Karniadakis GE, Israeli M, Orszag SA. High order splitting methods for the incompressible Navier–Stokes equations. Journal of Computational Physics 1991;97:414.
14. Beylkin G, Keiser JM. On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases. Journal of Computational Physics 1997;132:233–259.

15. Teleman AA, Strigini M, Cohen SM. Shaping morphogen gradients. Cell 2001;105:559–562. [PubMed: 11389824]
16. Wolpert, L.; Beddington, R.; Brockes, J.; Jessel, T.; Lawrence, P.; Meyerowitz, E. Principles of Development. Oxford University press; 2002.
17. Gurdon JB, Bourillot PY. Morphogen gradient interpretation. Nature 2001;413:797–803. [PubMed: 11677596]
18. Lander A, Nie Q, Wan F. Do morphogen gradients arise by diffusion? Developmental Cell 2002;2:785–796. [PubMed: 12062090]
19. Lander A, Nie Q, Wan F. Spatially distributed morphogen production and morphogen gradient formation. Mathematical Biosciences and Engineering 2005;2:239–262.
20. Ashe HL, Levine M. Local inhibition and long-range enhancement of Dpp signal transduction by Sog. Nature 1999;398:427–431. [PubMed: 10201373]
21. Bier E. A unity of opposites. Nature 1999;398:375–376. [PubMed: 10201364]
22. Oelgeschlager M, Larrain J, Geissert D, Roberts EM. The evolutionarily conserved bmp-binding protein twisted gastrulation promotes bmp signaling. Nature 2000;405:757–762. [PubMed: 10866189]
23. Ross JJ, Shimmi O, Vilmos P, Petryk A, Kim H, Gaudenz K, Hermanson S, Ekker SC, O'Connor MB, Marsh JL. Twisted gastrulation is a conserved extracellular BMP antagonist. Nature 2001;410:479–483. [PubMed: 11260716]
24. Kao, J.; Nie, Q.; Teng, A.; Wan, FYM.; Lander, AD.; Marsh, JL. Can morphogen activity be enhanced by its inhibitors?. Proceedings of the 2nd MIT Conference on Computational Mechanics; 2003. p. 1729-1733.
25. Lou Y, Nie Q, Wan F. Effects of Sog on Dpp–receptor binding. SIAM Journal on Applied Mathematics 2005;65:1748–1771. [PubMed: 17377624]
26. Mizutant C, Nie Q, Wan F, Zhang YT, Vilmos P, Bier E, Marsh L, Lander A. Formation of the bmp activity gradient in the drosophila embryo. Developmental Cell 2005;8(6):915–924. [PubMed: 15935780]
27. Lander A, Nie Q, Wan F, Zhang Y-T. Localized over-expression of Dpp receptors in a Drosophila embryo. 2007Preprint
28. Whitmarsh AJ, Davis RJ. Structural organization of MAP-kinase signaling modules by Scaffold proteins in yeast and mammals. Trends in Biochemical Sciences 1998;23:481–485. [PubMed: 9868371]
29. Morrison DK, Davis RJ. Regulation of MAP kinase signaling modules by Scaffold proteins in mammals. Annual Review of Cell and Developmental Biology 2003;19:91–118.
30. Wong W, Scott JD. AKAP signaling complexes: focal points in space and time. Nature Reviews Molecular Cell Biology 2004;5:959–970.
31. Park SH, Zarrinpar A, Lim WA. Rewiring MAP kinase pathways using alternative Scaffold assembly mechanisms. Science 2003;299:1061–1064. [PubMed: 12511654]
32. Harris K, Lamson RE, Nelson B, Hughes TR, Marton MJ, Roberts CJ, Boone C, Pryciak PM. Role of scaffolds in MAP kinase pathway specificity revealed by custom design of pathway-dedicated signaling proteins. Current Biology 2001;11:1815–1824. [PubMed: 11728304]
33. Dickens M, Rogers JS, Cavanagh J, Raitano A, Xia Z, Halpern JR, Greenberg ME, Sawyers CL, Davis RJ. A cytoplasmic inhibitor of the JNK signal transduction pathway. Science 1997;277:693–696. [PubMed: 9235893]
34. Cohen L, Henzel WJ, Baeuerle PA. IKAP is a Scaffold protein of the IKB kinase complex. Nature 1998;395:292–296. [PubMed: 9751059]
35. Kortum RL, Lewis RE. The molecular Scaffold KSR1 regulates the proliferative and oncogenic potential of cells. Molecular and Cellular Biology 2004;24:4407–4416. [PubMed: 15121859]
36. Bardwell L, Moore RD, Liu XF, Nie Q. Spatially-localized Scaffold proteins may simultaneously boost and suppress signaling. 2007Preprint
37. Liu XF, Nie Q. An implicit integration factor method with adaptive mesh refinements. 2007Preprint

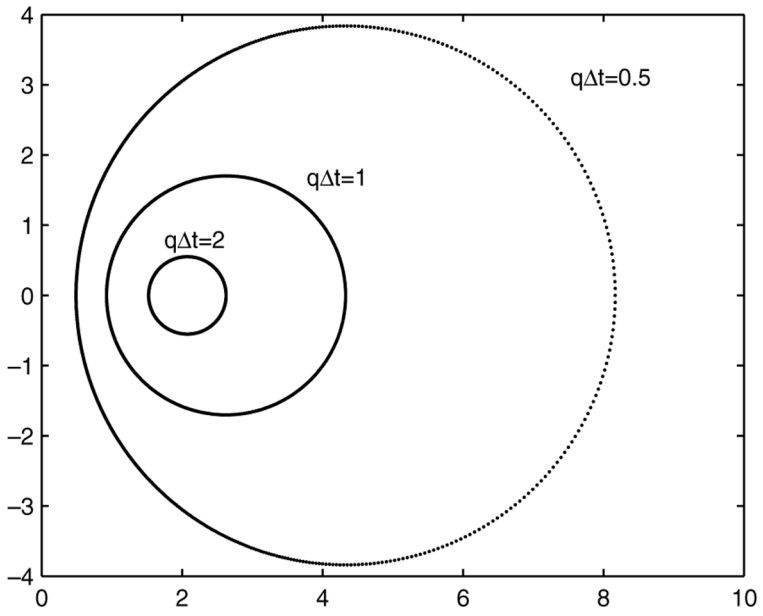

Fig. 1.
Stability regions (exterior of the closed curves) for cIIF2 with $(q_1 + q_2)\ \Delta t = 0.5, 1, 2$, where $q = q_1 + q_2$.

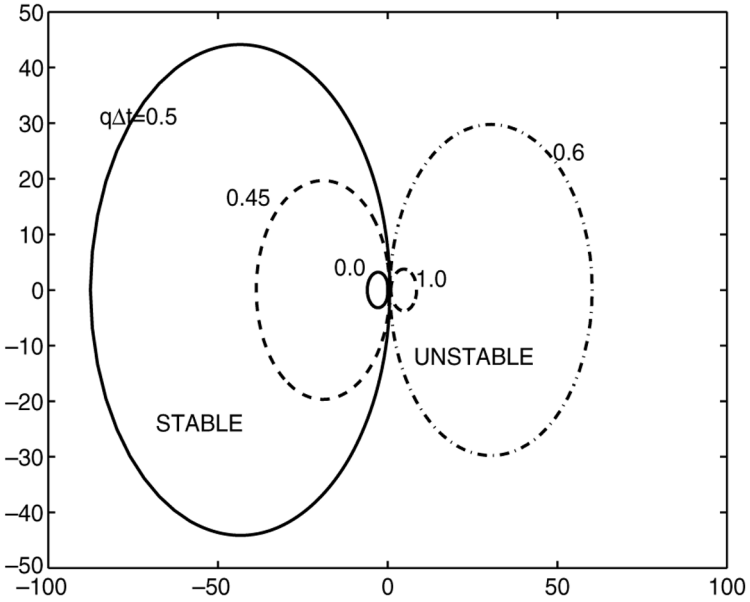

Fig. 2.
Stability regions for the third order cIIF scheme with $(q_1 + q_2)\,\Delta t = 0, 0.45, 0.5, 0.6, 1.0$, where $q = q_1 + q_2$.

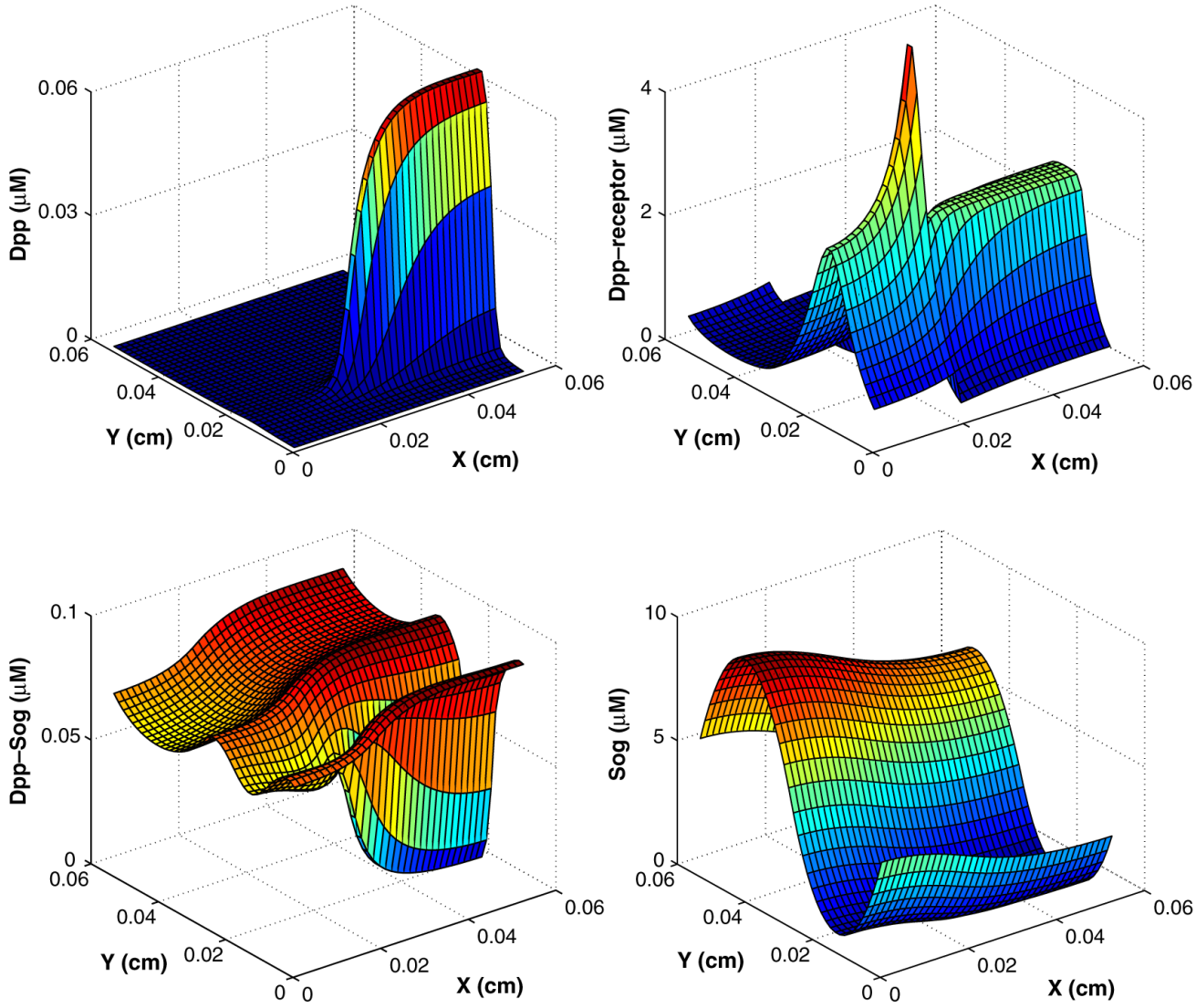

Fig. 3.

Concentrations of [*L*], [*LR*], [*LS*], [*S*] at $T = 7200$ s for the two-dimensional Dpp–Sog system (64) when receptors are over-expressed. $\Delta t = h_x = h_y = 0.001375$ in the simulation. Parameters are $D_L = D_{LS} = D_S = 85\ \mu m^2\ s^{-1}$; $v_L = 1$ nM s^{-1}; $v_S = 80$ nM s^{-1}; $k_{on} = 0.4\ \mu M^{-1}\ s^{-1}$; $k_{off} = 4 \times 10^{-6}\ s^{-1}$; $k_{deg} = 5 \times 10^{-4}\ s^{-1}$; $j_{on} = 95 \mu M^{-1}\ s^{-1}$; $j_{off} = 4 \times 10^{-6}\ s^{-1}$; $j_{deg} = 0.54\ s^{-1}$; $\tau = 1$; $R_0 = 3\mu M$.

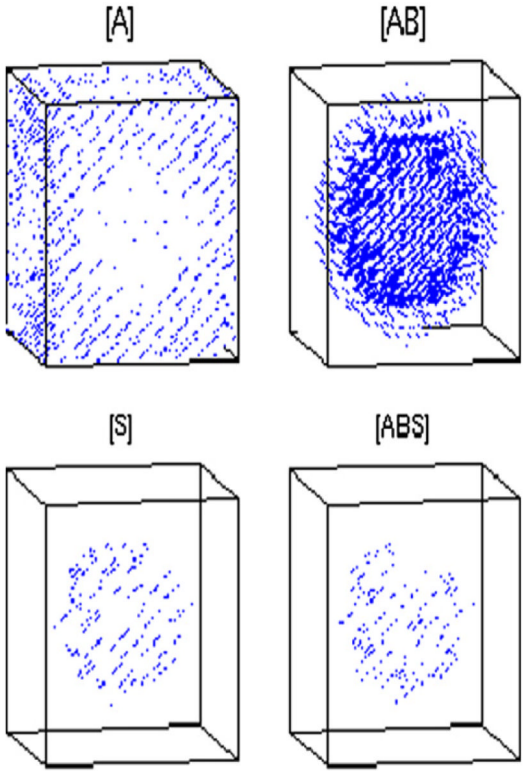

Fig. 4.

Concentrations for *A*, *B*, *AB* and *ABS* at $T = 10$ s. The dot density represents the level of concentrations. The parameters are $D = 1\ \mu m^2\ s^{-1}$, $k_{on} = 0.1\ (\mu Ms)^{-1}$, $k_{off} = 0.3\ s^{-1}$ $j_{on} = 100\ (\mu Ms)^{-1}$, $j_{off} = 0.05\ s^{-1}$, $j_{con} = 0.1\ s^{-1}$.

Table 1

A list of polynomials defined in (15) that correspond to the second, third and fourth order methods

Order	**2**	**3**	**4**
$p_{-1}(\tau)$	$\tau/\Delta t$	$\tau(\tau + \Delta t)/(2\Delta t^2)$	$\tau(\tau + \Delta t)(\tau + 2\Delta t)/(6\Delta t^3)$
$p_0(\tau)$	$(\Delta t - \tau)/\Delta t$	$-(\tau + \Delta t)(\tau - \Delta t)/\Delta t^2$	$-(\tau - \Delta t)(\tau + \Delta t)(\tau + 2\Delta t)/(2\Delta t^3)$
$p_1(\tau)$	0	$\tau(\tau - \Delta t)/(2\Delta t^2)$	$(\tau - \Delta t)\tau(\tau + 2\Delta t)/(2\Delta t^3)$
$p_2(\tau)$	0	0	$-(\tau - \Delta t)\tau(\tau + \Delta t)/(6\Delta t^3)$

Table 2

Coefficients for cIIF schemes with localized nonlinear systems

Order	α_1	α_0	α_{-1}	α_{-2}
1	1	0	0	0
2	$\frac{1}{2}$	$\frac{1}{2}$	0	0
3	$\frac{5}{12}$	$\frac{2}{3}$	$-\frac{1}{12}$	0
4	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$

Table 3

Error, order of accuracy, and CPU time for cIIF2, IIF2, and RK2 for a two-dimensional case

	cIIF2, $\Delta t = h_x/2$			IIF2, $\Delta t = h_x/2$			RK2, $\Delta t = h_x^2$		
$N \times N$	L^∞ error	Order	CPU (s)	L^∞ error	Order	CPU(s)	L^∞ error	Order	CPU (s)
40 × 40	5.65×10^{-4}	–	0.08	5.65×10^{-4}	–	142.47	5.65×10^{-4}	–	0.03
80 × 80	1.56×10^{-4}	1.86	0.15	Out of memory	–	–	1.55×10^{-4}	1.86	0.27
160 × 160	4.16×10^{-5}	1.91	1.02	Out of memory	–	–	4.16×10^{-5}	1.91	7.43
320 × 320	1.09×10^{-5}	1.93	33.54	Out of memory	–	–	1.09×10^{-5}	1.93	210.62

Table 4

Error, order of accuracy, and CPU time for cIIF2 and RK2 for a three-dimensional case

	cIIF2, $\Delta t = h_x/3$			RK2, $\Delta t = h_x/3$			RK2, $\Delta t = h_x^2/3$		
$N \times N$	L^∞ error	Order	CPU (s)	L^∞ error	Order	CPU(s)	L^∞ error	Order	CPU (s)
10 × 10 × 10	8.07×10^{-3}	–	0.0	8.07×10^{-3}	–	0.0	8.07×10^{-3}	–	0.0
20 × 20 × 20	2.02×10^{-3}	2.0	0.08	NC	–	–	2.02×10^{-3}	2.0	0.18
40 × 40 × 40	5.05×10^{-4}	2.0	2.18	NC	–	–	5.05×10^{-4}	2.0	6.85
80 × 80 × 80	1.26×10^{-4}	2.0	126.56	NC	–	–	1.26×10^{-4}	2.0	293.4

Table 5

Error, order of accuracy, and CPU time for cIIF2 applied to a two-dimensional system

Δt	$E_{\Delta t}$	Order	CPU (s)
1.375×10^{-3}	1.76×10^{-8}	–	7.54
6.875×10^{-4}	4.40×10^{-9}	2.00	15.08
3.438×10^{-4}	1.10×10^{-9}	2.00	30.20

Table 6

Error, order of accuracy, and CPU time for cIIF2 applied to a three-dimensional system

Δt	$E_{\Delta t}$	Order	CPU (s)
2.5×10^{-2}	2.09×10^{-4}	–	18.91
1.25×10^{-2}	5.24×10^{-5}	2.0	37.64
6.25×10^{-3}	1.32×10^{-5}	1.99	75.37