A Fourth-order Compact Finite Difference Scheme for Nonlinear Reaction-diffusion Systems

Tingting Xiao, Huiping Cai*

Abstract—**This work focuses on integrating a fourth-order compact finite difference scheme (CDS) with a fourth-order exponential time differencing Runge-Kutta method and dimensional splitting (ETDRK4P22-IF), termed the CDS-ETDRK4P22-IF method, for solving nonlinear reaction-diffusion equations (RDEs). The proposed approach employs the ETDRK4P22-IF scheme for temporal discretization following spatial discretization via CDS, resulting in a fully discrete model. Numerical experiments demonstrate that this method achieves both high convergence rates and enhanced computational efficiency.**

Index Terms—**Compact finite difference, Exponential time differencing , Fourth-order time-stepping, Reaction-diffusion equations**

I. Introduction

TE Reaction-diffusion equations (RDEs) represent a significant class of partial differential equations with extensive applications across diverse scientific domains. This study focuses on developing efficient and accurate numerical schemes for solving RDE systems, specifically addressing models governed by the following mathematical form:

$$\begin{cases} \dfrac{\partial u}{\partial t} = D\Delta u + f(u,t), & \text{in } \Omega_T, \\ u(x,t) = g(x,t), & \text{on} \Gamma_T. \end{cases} \tag{1}$$

Let $\Omega \subset \mathbb{R}^d\,(d = 1, 2, 3)$ be a bounded open subset. Define the spacetime domain as

$$\Omega_T := \Omega \times (0, T],$$

and its parabolic boundary as

$$\Gamma_T := \overline{\Omega}_T \setminus \Omega_T,$$

Over the past few decades, researchers have extensively investigated the numerical solution of Equation (1). When solving such equations numerically, one must address stiff nonlinear terms and complex boundary conditions. Due to their compact stencil and high accuracy, compact difference schemes (CDS) are particularly well-suited for handling boundary conditions through direct function specification at boundary nodes. A numerical framework combining compact difference schemes for spatial discretization with exponential time differencing (ETD) for temporal discretization was proposed, enabling efficient computation of Equation (1) with homogeneous or inhomogeneous boundary conditions via Fast Fourier Transform (FFT)-accelerated algorithms [7]. Subsequently, Zhu and Ju [12] successfully integrated fourth-order compact difference schemes with high-order ETD Runge-Kutta methods, providing an efficient computational approach for solving problems with Dirichlet or periodic boundary conditions. Later, Huang and Wu [6] further extended this fast-solving methodology to reaction-diffusion equations (RDEs) with Neumann boundary conditions. These advancements have progressively established a comprehensive framework for efficient numerical methods tailored to RDE models with diverse boundary conditions.

Among time-stepping methods for solving stiff ODE systems, certain approaches achieve stability through accurate treatment of the diffusion term via approximate integration and matrix exponentials. Subsequent developments in Exponential Time Differencing Runge-Kutta (ETDRK) methods enhanced computational efficiency by employing Padé rational approximations for matrix exponentials. To date, second- and fourth-order ETDRK-Padé schemes [8], [9], [10], [11] have been developed, along with an ETDRDP scheme [1] utilizing real distinct poles (RDP) in rational approximations. A dimension-splitting technique based on rational functions (termed ETDRDP-IF) was proposed, significantly improving computational efficiency for second-order ETDRK schemes solving multidimensional RDEs [2], [3]. To enhance the efficiency of fourth-order ETDRK schemes for multidimensional RDEs, E. O. Asante-Asamani [4] developed the ETDRK4P22-IF scheme, which employs an A-stable Padé(2,2) rational approximation for matrix exponentials. However, a notable research gap persists: the integration of high-order compact difference schemes with the ETDRK4P22-IF method (using Padé(2,2)) remains virtually unexplored for solving RDE systems.

In this study, we integrate a fourth-order compact difference scheme (CDS) with the ETDRK4P22-IF method, proposing a high-efficiency fourth-order numerical framework termed CDS-ETDRK4P22-IF. This integration yields a fully discrete formulation for Equation (1). Compared to the standalone ETDRK4P22-IF method, this approach provides an efficient, high-accuracy, and stable numerical methodology for solving two-dimensional nonlinear reaction-diffusion equations.

The remainder of this paper is organized as follows: Section 2 presents the fourth-order compact difference discretization of the Laplacian operator. Section 3 details the fully discrete formulation achieved through temporal discretization via the fourth-order ETDRK4P22-IF method. Finally, Section 4 provides concluding remarks and outlines future research directions.

II. Discretization in space

We now discretize the Laplacian operator $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ over the spatial domain $[a, b]^2$. A uniform mesh is construct-

Manuscript received April 7, 2025; revised June 13, 2025.

This work was supported in part by the NSF of China (No. 11861054), Natural Science Foundation of Guangxi (No. 2020GXNSFAA297223).

Tingting Xiao is a graduate student of the Guangxi Normal University, Guilin Guangxi 541000, P.R. China (e-mail: tingtng@stu.gxnu.edu.cn).

*Huiping Cai is an associate professor of the College of Mathematics and Statistics in Guangxi Normal University, Guilin 541006, P.R. China (Corresponding author, e-mail: cailp1103@sina.com).

ed by partitioning each spatial dimension into m intervals of length $h = \frac{b-a}{m}$ (where $m \geq 2$), resulting in $m+1$ grid points per dimension. The grid points in the x-direction are defined as $x_i = a + ih$ for $i = 0, 1, \ldots, m$, and analogously in the y-direction. The second-order partial derivative of a function $u(x, y, t)$ with respect to x at position x_i is then discretized using a fourth-order compact finite difference scheme.

To derive the compact finite difference approximation, we first consider Taylor series expansions about the point x_i for its neighboring points x_{i+1} and x_{i-1}:

$$\begin{aligned} u(x_{i+1}) &= u(x_i) + u'(x_i)h + \frac{u''(x_i)}{2}h^2 \\ &\quad + \frac{u'''(x_i)}{6}h^3 + \frac{u^{(4)}(x_i)}{24}h^4 + \frac{u^{(5)}(x_i)}{120}h^5 + O(h^6), \\ u(x_{i-1}) &= u(x_i) - u'(x_i)h + \frac{u''(x_i)}{2}h^2 \\ &\quad - \frac{u'''(x_i)}{6}h^3 + \frac{u^{(4)}(x_i)}{24}h^4 - \frac{u^{(5)}(x_i)}{120}h^5 + O(h^6). \end{aligned}$$

Adding these two equations yields:

$$u(x_{i+1}) + u(x_{i-1}) = 2u(x_i) + h^2 u''(x_i) + \frac{1}{12}h^4 u^{(4)}(x_i) + O(h^6).$$

Rearranging terms to solve for $u''(x_i)$:

$$h^2 u''(x_i) = u(x_{i+1}) - 2u(x_i) + u(x_{i-1}) - \frac{1}{12}h^4 u^{(4)}(x_i) + O(h^6).$$

The second derivative at x_i is then approximated as:

$$u''(x_i) = \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} - \frac{1}{12}h^2 u^{(4)}(x_i) + O(h^4).$$

Equivalently,

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{x_i} = \delta_x^2 u_i - \frac{h^2}{12}\left.\frac{\partial^4 u}{\partial x^4}\right|_{x_i} + O(h^4), \tag{2}$$

where the central difference operator is defined as:

$$\delta_x^2 u_i = \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}.$$

Consider the one-dimensional case of Equation (1) with diffusion coefficient $D = 1$:

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + f(u, t), \tag{3}$$

which implies:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial u}{\partial t} - f(u, t).$$

Defining $v = \frac{\partial^2 u}{\partial x^2}$, the fourth-order derivative is expressed as:

$$\frac{\partial^4 u}{\partial x^4} = \frac{\partial^2 v}{\partial x^2}.$$

Substituting v yields:

$$\frac{\partial^4 u}{\partial x^4} = \frac{\partial^2}{\partial x^2}\left[\frac{\partial u}{\partial t} - f(u, t)\right] = \delta_x^2\left[\left.\frac{\partial u}{\partial t}\right|_{x_i} - f(u_i, t)\right].$$

Substituting into Equation (2), the second derivative at x_i becomes:

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{x_i} = \delta_x^2 u_i - \frac{h^2}{12}\delta_x^2\left[\left.\frac{\partial u}{\partial t}\right|_{x_i} - f(u_i, t)\right].$$

The semi-discrete formulation of Equation (3) at x_i is thus:

$$\left.\frac{\partial u}{\partial t}\right|_{x_i} = \delta_x^2 u_i - \frac{h^2}{12}\delta_x^2\left[\left.\frac{\partial u}{\partial t}\right|_{x_i} - f(u_i, t)\right] + f(u_i, t), \tag{4}$$

where

$$\delta_x^2\left[\left.\frac{\partial u}{\partial t}\right|_{x_i} - f(u_i, t)\right]$$

denotes application of the central difference operator to the expression $\left.\frac{\partial u}{\partial t}\right|_{x_i} - f(u_i, t)$. The subscript i indicates evaluation at spatial point x_i.

Expanding Equation (4) and combining terms yields:

$$\begin{aligned} \frac{1}{12}\left(\left.\frac{\partial u}{\partial t}\right|_{x_{i+1}} + 10\left.\frac{\partial u}{\partial t}\right|_{x_i} + \left.\frac{\partial u}{\partial t}\right|_{x_{i-1}}\right) &= \frac{1}{h^2}\left(u_{i+1} - 2u_i + u_{i-1}\right) \\ &\quad + \frac{1}{12}\Big[f(u_{i+1}, t) + 10f(u_i, t) \\ &\quad + f(u_{i-1}, t)\Big], \end{aligned}$$

where $u_j \equiv u(x_j, t)$ denotes the solution value at grid point x_j, and h is the spatial step size.

For Equation (3) with homogeneous Dirichlet boundary conditions, the matrix formulation of (4) is:

$$\begin{aligned} \frac{d\boldsymbol{U}}{dt} &= \boldsymbol{C}^{-1}\boldsymbol{A}\boldsymbol{U} + \boldsymbol{C}^{-1}\boldsymbol{F}(\boldsymbol{U}, t) \\ \boldsymbol{U}(0) &= \boldsymbol{U}_0, \end{aligned} \tag{5}$$

where the matrices and vectors are defined as follows: The mass matrix $\boldsymbol{C}$ is defined as:

$$\boldsymbol{C} = \frac{1}{12}\begin{pmatrix} 10 & 1 & & & & \\ 1 & 10 & 1 & & & \\ & 1 & 10 & 1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & 10 & 1 \\ & & & & 1 & 10 \end{pmatrix}_{m\times m}.$$

The stiffness matrix $\boldsymbol{A}$ (discrete Laplacian) is given by:

$$\boldsymbol{A} = \frac{1}{h^2}\begin{pmatrix} -2 & 1 & & & \\ 1 & -2 & 1 & & \\ & 1 & -2 & \ddots & \\ & & \ddots & \ddots & 1 \\ & & & 1 & -2 \end{pmatrix}_{m\times m}.$$

For homogeneous Dirichlet boundary conditions, the boundary term vector is:

$$\boldsymbol{D} = \frac{1}{h^2}\begin{pmatrix} u_0 \\ 0 \\ \vdots \\ 0 \\ u_m \end{pmatrix} + \frac{1}{12}\begin{pmatrix} f(u_0, t) \\ 0 \\ \vdots \\ 0 \\ f(u_m, t) \end{pmatrix},$$

where u_0 and u_m denote the fixed boundary values. The state vector and nonlinear term vector are defined as:

$$\boldsymbol{U} = \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{m-1} \end{pmatrix}, \quad \boldsymbol{F} = \begin{pmatrix} f(u_1, t) \\ f(u_2, t) \\ \vdots \\ f(u_{m-1}, t) \end{pmatrix}.$$

Defining $\boldsymbol{A}_p = \boldsymbol{C}^{-1}\boldsymbol{A}$ and $\widetilde{\boldsymbol{F}}(\boldsymbol{U}(t), t) = \boldsymbol{F} + \boldsymbol{C}^{-1}\boldsymbol{D}$, the system can be expressed as:

$$\begin{aligned} \frac{d\boldsymbol{U}}{dt} &= -\boldsymbol{A}_p\boldsymbol{U} + \widetilde{\boldsymbol{F}}(\boldsymbol{U}(t), t), \\ \boldsymbol{U}(0) &= \boldsymbol{U}_0. \end{aligned} \tag{6}$$

The matrix approximation of the 2D Laplacian operator $\Delta = \partial_{xx} + \partial_{yy}$ is constructed using Kronecker products:

$$\boldsymbol{A}_{2\text{D}} = \boldsymbol{A}_x + \boldsymbol{A}_y,$$

where $\boldsymbol{A}_x$ and $\boldsymbol{A}_y$ represent the discrete approximations of ∂_{xx} and ∂_{yy}, respectively. Using the 1D operator $\boldsymbol{A}_p$ and identity matrix $\boldsymbol{I}_p \in \mathbb{R}^{p\times p}$ (p being the number of spatial discretization points per dimension), we have:

$$\boldsymbol{A}_x = \boldsymbol{A}_p \otimes \boldsymbol{I}_p, \quad \boldsymbol{A}_y = \boldsymbol{I}_p \otimes \boldsymbol{A}_p.$$

As established in [2], [3], the matrices $\boldsymbol{A}_x$ and $\boldsymbol{A}_y$ are commutative:

$$\boldsymbol{A}_x\boldsymbol{A}_y = \boldsymbol{A}_y\boldsymbol{A}_x.$$

This commutativity enables the decomposition $\boldsymbol{A}_{2\text{D}} = \boldsymbol{A}_x + \boldsymbol{A}_y$, which reduces the bandwidth of the system matrix and accelerates linear system solutions. To further enhance computational efficiency, we employ the ETDRK4P22-IF method for temporal integration.

III. Discretization in time

A. Dimensional splitting for fourth-order ETD Runge-Kutta scheme

We implement dimensional splitting for the fourth-order exponential time differencing Runge-Kutta (ETDRK4) scheme [5] to solve the semi-discrete ODE system (6). The dimensional splitting scheme is given by:

$$\begin{aligned}
\bar{a}_n &= e^{-\frac{k}{2}\boldsymbol{A}_2}e^{-\frac{k}{2}\boldsymbol{A}_1}\boldsymbol{U}_n + \tilde{\boldsymbol{P}}(k\boldsymbol{A}_2)e^{-\frac{k}{2}\boldsymbol{A}_1}\boldsymbol{F}(\boldsymbol{U}_n, t_n)\\
\bar{b}_n &= e^{-\frac{k}{2}\boldsymbol{A}_2}e^{-\frac{k}{2}\boldsymbol{A}_1}\boldsymbol{U}_n + \tilde{\boldsymbol{P}}(k\boldsymbol{A}_2)\boldsymbol{F}\left(\bar{a}_n, t_n + \tfrac{k}{2}\right)\\
\bar{c}_n &= e^{-\frac{k}{2}\boldsymbol{A}_2}e^{-\frac{k}{2}\boldsymbol{A}_1}\bar{a}_n + \tilde{\boldsymbol{P}}(k\boldsymbol{A}_2)\Big[2e^{-\frac{k}{2}\boldsymbol{A}_1}\boldsymbol{F}\left(\bar{b}_n, t_n + \tfrac{k}{2}\right)\\
&\quad - e^{-k\boldsymbol{A}_1}\boldsymbol{F}(\boldsymbol{U}_n, t_n)\Big] \qquad (7)\\
\boldsymbol{U}_{n+1} &= e^{-k\boldsymbol{A}_1}e^{-k\boldsymbol{A}_2}\boldsymbol{U}_n + \boldsymbol{P}_1(k\boldsymbol{A}_2)e^{-k\boldsymbol{A}_1}\boldsymbol{F}(\boldsymbol{U}_n, t_n)\\
&\quad + 2\boldsymbol{P}_2(k\boldsymbol{A}_2)e^{-\frac{k}{2}\boldsymbol{A}_1}\boldsymbol{G}\left(\bar{a}_n, \bar{b}_n, t_n + \tfrac{k}{2}\right)\\
&\quad + \boldsymbol{P}_3(k\boldsymbol{A}_2)\boldsymbol{F}(\bar{c}_n, t_n + k)
\end{aligned}$$

where the coefficient matrices and coupling term are defined as:

$$\begin{aligned}
\boldsymbol{P}_1(k\boldsymbol{A}) &= \tfrac{1}{k^2}(-\boldsymbol{A})^{-3}\Big[-4\boldsymbol{I} + k\boldsymbol{A} + e^{-k\boldsymbol{A}}(4\boldsymbol{I} + 3k\boldsymbol{A} + k^2\boldsymbol{A}^2)\Big]\\
\boldsymbol{P}_2(k\boldsymbol{A}) &= \tfrac{1}{k^2}(-\boldsymbol{A})^{-3}\Big[2\boldsymbol{I} - k\boldsymbol{A} - e^{-k\boldsymbol{A}}(2\boldsymbol{I} + k\boldsymbol{A})\Big]\\
\boldsymbol{P}_3(k\boldsymbol{A}) &= \tfrac{1}{k^2}(-\boldsymbol{A})^{-3}\Big[-4\boldsymbol{I} + 3k\boldsymbol{A} - k^2\boldsymbol{A}^2 + e^{-k\boldsymbol{A}}(4\boldsymbol{I} + k\boldsymbol{A})\Big]\\
\tilde{\boldsymbol{P}}(k\boldsymbol{A}) &= -\boldsymbol{A}^{-1}(e^{-\frac{k}{2}\boldsymbol{A}} - \boldsymbol{I})\\
\boldsymbol{G}(\bar{a}_n, \bar{b}_n, t_n + \tfrac{k}{2}) &= \boldsymbol{F}(\bar{a}_n, t_n + \tfrac{k}{2}) + \boldsymbol{F}(\bar{b}_n, t_n + \tfrac{k}{2}) \qquad (8)
\end{aligned}$$

To enhance computational efficiency, we approximate the matrix exponentials using the Padé(2,2) rational functions [4]:

$$\begin{aligned}
e^{-k\boldsymbol{A}} &\approx R_{2,2}(k\boldsymbol{A})\\
&= (12\boldsymbol{I} - 6k\boldsymbol{A} + k^2\boldsymbol{A}^2)(12\boldsymbol{I} + 6k\boldsymbol{A} + k^2\boldsymbol{A}^2)^{-1}\\
e^{-\frac{k}{2}\boldsymbol{A}} &\approx \tilde{R}_{2,2}(k\boldsymbol{A})\\
&= (48\boldsymbol{I} - 12k\boldsymbol{A} + k^2\boldsymbol{A}^2)(48\boldsymbol{I} + 12k\boldsymbol{A} + k^2\boldsymbol{A}^2)^{-1}
\end{aligned}$$

Replacing the exponential terms in (7) with their rational approximations yields:

$$\begin{aligned}
\bar{a}_n &= \tilde{R}_{2,2}(k\boldsymbol{A}_2)\tilde{R}_{2,2}(k\boldsymbol{A}_1)\boldsymbol{U}_n\\
&\quad + \tilde{\boldsymbol{P}}(k\boldsymbol{A}_2)\tilde{R}_{2,2}(k\boldsymbol{A}_1)\boldsymbol{F}(\boldsymbol{U}_n, t_n) \qquad (9)\\
\bar{b}_n &= \tilde{R}_{2,2}(k\boldsymbol{A}_2)\tilde{R}_{2,2}(k\boldsymbol{A}_1)\boldsymbol{U}_n + \tilde{\boldsymbol{P}}(k\boldsymbol{A}_2)\boldsymbol{F}(\bar{a}_n, t_n + \tfrac{k}{2}) \qquad (10)\\
\bar{c}_n &= \tilde{R}_{2,2}(k\boldsymbol{A}_2)\tilde{R}_{2,2}(k\boldsymbol{A}_1)\bar{a}_n\\
&\quad + \tilde{\boldsymbol{P}}(k\boldsymbol{A}_2)\Big[2\tilde{R}_{2,2}(k\boldsymbol{A}_1)\boldsymbol{F}(\bar{b}_n, t_n + \tfrac{k}{2})\\
&\quad - R_{2,2}(k\boldsymbol{A}_1)\boldsymbol{F}(\boldsymbol{U}_n, t_n)\Big] \qquad (11)\\
\boldsymbol{U}_{n+1} &= R_{2,2}(k\boldsymbol{A}_1)R_{2,2}(k\boldsymbol{A}_2)\boldsymbol{U}_n\\
&\quad + \boldsymbol{P}_1(k\boldsymbol{A}_2)R_{2,2}(k\boldsymbol{A}_1)\boldsymbol{F}(\boldsymbol{U}_n, t_n)\\
&\quad + 2\boldsymbol{P}_2(k\boldsymbol{A}_2)\tilde{R}_{2,2}(k\boldsymbol{A}_1)\boldsymbol{G}(\bar{a}_n, \bar{b}_n, t_n + \tfrac{k}{2})\\
&\quad + \boldsymbol{P}_3(k\boldsymbol{A}_2)\boldsymbol{F}(\bar{c}_n, t_n + k) \qquad (12)
\end{aligned}$$

with the coefficient matrices simplified to:

$$\begin{aligned}
\boldsymbol{P}_1(k\boldsymbol{A}) &= k(2\boldsymbol{I} - k\boldsymbol{A})(12\boldsymbol{I} + 6k\boldsymbol{A} + k^2\boldsymbol{A}^2)^{-1}\\
\boldsymbol{P}_2(k\boldsymbol{A}) &= 2k(12\boldsymbol{I} + 6k\boldsymbol{A} + k^2\boldsymbol{A}^2)^{-1}\\
\boldsymbol{P}_3(k\boldsymbol{A}) &= k(2\boldsymbol{I} + k\boldsymbol{A})(12\boldsymbol{I} + 6k\boldsymbol{A} + k^2\boldsymbol{A}^2)^{-1}\\
\tilde{\boldsymbol{P}}(k\boldsymbol{A}) &= 24k(48\boldsymbol{I} + 12k\boldsymbol{A} + k^2\boldsymbol{A}^2)^{-1}
\end{aligned}$$

B. Implementation of CDS-ETDRK4P22-IF Scheme

This subsection details the implementation of the CDS-ETDRK4P22-IF scheme with dimensional splitting. We first compute the intermediate solutions $\bar{a}_n$, $\bar{b}_n$ and $\bar{c}_n$. Beginning with $\bar{a}_n$, we simplify Equation (9):

$$\bar{a}_n = \tilde{R}_{2,2}(k\boldsymbol{A}_1)(\boldsymbol{U}_n + 2\,\mathrm{Re}(a_{n1})),$$

where

$$(k\boldsymbol{A}_2 - c_2\boldsymbol{I})a_{n1} = 2w_{11}\boldsymbol{U}_n + 24kw_{51}\boldsymbol{F}(\boldsymbol{U}_n, t_n).$$

Defining $a_{n2} = \boldsymbol{U}_n + 2\,\mathrm{Re}(a_{n1})$ and applying $\tilde{R}_{2,2}(k\boldsymbol{A}_1)$ yields:

$$\begin{aligned}
\bar{a}_n &= \left(\boldsymbol{I} + 4\,\mathrm{Re}\left(w_{11}(k\boldsymbol{A}_1 - c_2\boldsymbol{I})^{-1}\right)\right)a_{n2}\\
&= a_{n2} + 2\,\mathrm{Re}\left((k\boldsymbol{A}_1 - c_2\boldsymbol{I})^{-1}(2w_{11}a_{n2})\right)\\
&= a_{n2} + 2\,\mathrm{Re}(a_{n3}),
\end{aligned}$$

with

$$(k\boldsymbol{A}_1 - c_2\boldsymbol{I})a_{n3} = 2w_{11}a_{n2}.$$

For $\bar{b}_n$:

$$\begin{aligned}
\bar{b}_n &= \tilde{R}_{2,2}(k\boldsymbol{A}_1)\left(\boldsymbol{U}_n + \mathrm{Re}\left[(k\boldsymbol{A}_2 - c_2\boldsymbol{I})^{-1}(4w_{11}\boldsymbol{U}_n)\right]\right)\\
&\quad + k\,\mathrm{Re}\left[(k\boldsymbol{A}_2 - c_2\boldsymbol{I})^{-1}\left(48w_{51}\boldsymbol{F}\left(\bar{a}_n, t_n + \tfrac{k}{2}\right)\right)\right],
\end{aligned}$$

which simplifies to:

$$\bar{b}_n = \tilde{R}_{2,2}(k\boldsymbol{A}_1)\,(\boldsymbol{U}_n + 2\ \mathrm{Re}(b_{n1})) + 2\ \mathrm{Re}(b_{n2}),$$

where

$$(k\boldsymbol{A}_2 - c_2\boldsymbol{I})b_{n1} = 2w_{11}\boldsymbol{U}_n,$$

and

$$(k\boldsymbol{A}_2 - c_2\boldsymbol{I})b_{n2} = 24kw_{51}\boldsymbol{F}\left(\bar{a}_n, t_n + \tfrac{k}{2}\right).$$

Let $b_{n3} = \boldsymbol{U}_n + 2\ \mathrm{Re}(b_{n1})$. Applying $\tilde{R}_{2,2}(k\boldsymbol{A}_1)$ gives:

$$\begin{aligned}\bar{b}_n &= \left(\boldsymbol{I} + 4\ \mathrm{Re}\left(w_{11}(k\boldsymbol{A}_1 - c_2\boldsymbol{I})^{-1}\right)\right) b_{n3} + 2\ \mathrm{Re}(b_{n2}) \\ &= b_{n3} + 2\ \mathrm{Re}\left((k\boldsymbol{A}_1 - c_2\boldsymbol{I})^{-1}(2w_{11}b_{n3})\right) + 2\ \mathrm{Re}(b_{n2}) \\ &= b_{n3} + 2\ \mathrm{Re}(b_{n4}) + 2\ \mathrm{Re}(b_{n2}),\end{aligned}$$

with

$$(k\boldsymbol{A}_1 - c_2\boldsymbol{I})b_{n4} = 2w_{11}b_{n3}.$$

For $\bar{c}_n$:

$$\begin{aligned}\bar{c}_n &= \tilde{R}_{2,2}(k\boldsymbol{A}_1)\left[\tilde{R}_{2,2}(k\boldsymbol{A}_2)\bar{a}_n + 2\tilde{\boldsymbol{P}}(k\boldsymbol{A}_2)\boldsymbol{F}\left(\bar{b}_n, t_n + \tfrac{k}{2}\right)\right] \\ &\quad - R_{2,2}(k\boldsymbol{A}_1)\tilde{\boldsymbol{P}}(k\boldsymbol{A}_2)\boldsymbol{F}(\boldsymbol{U}_n, t_n) \\ &= \tilde{R}_{2,2}(k\boldsymbol{A}_1)c^*_{n1} - R_{2,2}(k\boldsymbol{A}_1)c^*_{n2},\end{aligned}$$

where

$$c^*_{n1} = \tilde{R}_{2,2}(k\boldsymbol{A}_2)\bar{a}_n + 2\tilde{\boldsymbol{P}}(k\boldsymbol{A}_2)\boldsymbol{F}\left(\bar{b}_n, t_n + \tfrac{k}{2}\right)$$

and

$$c^*_{n2} = \tilde{\boldsymbol{P}}(k\boldsymbol{A}_2)\boldsymbol{F}(\boldsymbol{U}_n, t_n).$$

Using partial fraction decomposition yields:

$$\bar{c}_n = c^*_{n1} + 2\ \mathrm{Re}(c_{n3}) - \left[c^*_{n2} + 2\ \mathrm{Re}(c_{n4})\right],$$

with

$$(k\boldsymbol{A}_1 - c_2\boldsymbol{I})c_{n3} = 2w_{11}c^*_{n1} \quad \text{and} \quad (k\boldsymbol{A}_1 - c_1\boldsymbol{I})c_{n4} = w_{11}c^*_{n2}.$$

Additionally:

$$c^*_{n1} = \tilde{R}_{2,2}(k\boldsymbol{A}_2)\bar{a}_n + 2\tilde{\boldsymbol{P}}(k\boldsymbol{A}_2)\boldsymbol{F}\left(\bar{b}_n, t_n + \tfrac{k}{2}\right) = \bar{a}_n + 2\ \mathrm{Re}(c_{n1}),$$

since

$$(k\boldsymbol{A}_2 - c_2\boldsymbol{I})c_{n1} = 2w_{11}\bar{a}_n + 48kw_{51}\boldsymbol{F}\left(\bar{b}_n, t_n + \tfrac{k}{2}\right).$$

Similarly:

$$c^*_{n2} = \tilde{\boldsymbol{P}}(k\boldsymbol{A}_2)\boldsymbol{F}(\boldsymbol{U}_n, t_n) = 2\ \mathrm{Re}(c_{n2}),$$

where

$$(k\boldsymbol{A}_2 - c_2\boldsymbol{I})c_{n2} = 24kw_{51}\boldsymbol{F}(\boldsymbol{U}_n, t_n).$$

After computing the intermediate solutions $\bar{a}_n$, $\bar{b}_n$, and $\bar{c}_n$, we solve Equation (12), define:

$$\begin{aligned}\boldsymbol{U}^*_{n1} &= R_{2,2}(k\boldsymbol{A}_2)\boldsymbol{U}_n + P_1(k\boldsymbol{A}_2)\boldsymbol{F}(\boldsymbol{U}_n, t_n), \\ \boldsymbol{U}^*_{n2} &= 2P_2(k\boldsymbol{A}_2)\boldsymbol{G}\left(\bar{a}_n, \bar{b}_n, t_n + \tfrac{k}{2}\right), \\ \boldsymbol{U}^*_{n3} &= P_3(k\boldsymbol{A}_2)\boldsymbol{F}(\bar{c}_n, t_n + k).\end{aligned}$$

Equation (12) then becomes:

$$\begin{aligned}\boldsymbol{U}_{n+1} &= R_{2,2}(k\boldsymbol{A}_1)\boldsymbol{U}^*_{n1} + \tilde{R}_{2,2}(k\boldsymbol{A}_1)\boldsymbol{U}^*_{n2} + \boldsymbol{U}^*_{n3} \\ &= \left(\boldsymbol{I} + 2\,\mathrm{Re}\left(w_{11}(k\boldsymbol{A}_1 - c_1\boldsymbol{I})^{-1}\right)\right)\boldsymbol{U}^*_{n1} \\ &\quad + \left(\boldsymbol{I} + 4\,\mathrm{Re}\left(w_{11}(k\boldsymbol{A}_1 - c_2\boldsymbol{I})^{-1}\right)\right)\boldsymbol{U}^*_{n2} + \boldsymbol{U}^*_{n3} \\ &= \boldsymbol{U}^*_{n1} + \boldsymbol{U}^*_{n2} + 2\,\mathrm{Re}\left((k\boldsymbol{A}_1 - c_1\boldsymbol{I})^{-1}(w_{11}\boldsymbol{U}^*_{n1})\right) \\ &\quad + 2\,\mathrm{Re}\left((k\boldsymbol{A}_1 - c_2\boldsymbol{I})^{-1}(2w_{11}\boldsymbol{U}^*_{n2})\right) + \boldsymbol{U}^*_{n3} \\ &= \boldsymbol{U}^*_{n1} + \boldsymbol{U}^*_{n2} + 2\,\mathrm{Re}(\boldsymbol{U}_{n4}) + 2\,\mathrm{Re}(\boldsymbol{U}_{n5}) + \boldsymbol{U}^*_{n3},\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{U}^*_{n1} &= \boldsymbol{U}_n + 2\,\mathrm{Re}(\boldsymbol{U}_{n1}), \\ &\quad \text{with} \quad (k\boldsymbol{A}_2 - c_1\boldsymbol{I})\boldsymbol{U}_{n1} = w_{11}\boldsymbol{U}_n + kw_{21}\boldsymbol{F}(\boldsymbol{U}_n, t_n), \\ \boldsymbol{U}^*_{n2} &= 2\,\mathrm{Re}(\boldsymbol{U}_{n2}), \\ &\quad \text{with} \quad (k\boldsymbol{A}_2 - c_1\boldsymbol{I})\boldsymbol{U}_{n2} = 4w_{31}k\boldsymbol{G}\left(\bar{a}_n, \bar{b}_n, t_n + \tfrac{k}{2}\right), \\ \boldsymbol{U}^*_{n3} &= 2\,\mathrm{Re}(\boldsymbol{U}_{n3}), \\ &\quad \text{with} \quad (k\boldsymbol{A}_2 - c_1\boldsymbol{I})\boldsymbol{U}_{n3} = w_{41}k\boldsymbol{F}(\bar{c}_n, t_n + k).\end{aligned}$$

The complete CDS-ETDRK4P22-IF implementation procedure is summarized as follows:

1) Compute intermediate solutions:
$$\bar{a}_n, \quad \bar{b}_n, \quad \text{and} \quad \bar{c}_n$$
2) Using intermediate solutions, compute:
$$\boldsymbol{U}_{n1}, \ \boldsymbol{U}_{n2}, \ \text{and} \ \boldsymbol{U}_{n3}$$
3) Solve for modified variables:
$$\boldsymbol{U}^*_{n1}, \ \boldsymbol{U}^*_{n2}, \ \text{and} \ \boldsymbol{U}^*_{n3}$$
4) Calculate auxiliary variables:
$$\boldsymbol{U}_{n4} \ \text{and} \ \boldsymbol{U}_{n5}$$
5) Compute the numerical solution:
$$\boldsymbol{U}_{n+1}$$

IV. Conclusion and outlook

In this work, we propose a fourth-order compact difference scheme coupled with a fourth-order exponential time differencing (ETD) method via dimensional splitting. By employing the Padé(2,2) rational approximation for efficient evaluation of matrix exponential operators, the proposed method achieves significant improvements in computational efficiency. Numerical experiments demonstrate that this framework maintains high-order accuracy while substantially reducing computational costs. However, our implementation reveals that the resulting matrices exhibit dense structures, contrary to the theoretically expected sparsity patterns. This structural density leads to considerable computational overhead, particularly in memory storage and matrix operations. To mitigate this limitation, future research will focus on developing specialized sparsification techniques for the system matrix $\boldsymbol{A}_p$. These optimizations aim to reduce computational resource demands while preserving the numerical accuracy, thereby enhancing the practical applicability of the proposed algorithm for large-scale simulations.

References

[1] E. O. Asante-Asamani, A. Q. M. Khaliq, and B. A. Wade, A real distinct poles exponential time differencing scheme for reaction-diffusion systems, *J. Comput. Appl. Math*, vol. 299, pp. 24–34, 2016.

[2] E. O. Asante-Asamani and B. A. Wade, A dimensional splitting of ETD schemes for reaction-diffusion systems, *Commun. Comput. Phys*, vol. 19, no. 5, pp. 1343–1356, 2016.

[3] E. O. Asante-Asamani, A. Kleefeld, and B. A. Wade, A second-order exponential time differencing scheme for non-linear reaction-diffusion systems with dimensional splitting, *J. Comput. Phys*, vol. 415, pp. 109490, 2020.

[4] E. O. Asante-Asamani, A. Kleefeld, and B. A. Wade, A fourth-order exponential time differencing scheme with dimensional splitting for non-linear reaction-diffusion systems, *J. Comput. Appl. Math*, 2025.

[5] S. M. Cox and P. C. Matthews, Exponential time differencing for stiff systems, *J. Comput. Phys*, vol. 176, pp. 430–455, 2002.

[6] J. Huang, L. Ju, and B. Wu, A fast compact exponential time differencing method for semilinear parabolic equations with Neumann boundary conditions, *Appl. Math. Lett*, vol. 94, pp. 257–265, 2019.
[7] L. Ju, J. Zhang, L. Zhu, and Q. Du, Fast explicit integration factor methods for semilinear parabolic equations, *J. Sci. Comput*, vol. 62, no. 2, pp. 431–455, 2015.
[8] B. Kleefeld, A. Q. M. Khaliq, and B. A. Wade, An ETD Crank-Nicolson method for reaction-diffusion systems, *Numer. Methods Partial Differ. Equ*, vol. 28, pp. 1309–1335, 2012.
[9] A. Q. M. Khaliq, J. Martin-Vaquero, B. A. Wade, and M. Yousuf, Smoothing schemes for reaction-diffusion systems with nonsmooth data, *J. Comput. Appl. Math*, vol. 223, no. 1, pp. 374–386, 2009.
[10] M. Yousuf, Efficient L-stable method for parabolic problems with application to pricing American options under stochastic volatility, *Appl. Math. Comput*, vol. 213, no. 1, pp. 121–136, 2009.
[11] M. Yousuf, A. Q. M. Khaliq, and B. Kleefeld, The numerical approximation of nonlinear Black-Scholes model for exotic path-dependent American options with transaction cost, *Int. J. Comput. Math*, vol. 89, no. 9, pp. 1239–1254, 2012.
[12] L. Zhu, L. Ju, and W. Zhao, Fast high-order compact exponential time differencing Runge-Kutta methods for second-order semilinear parabolic equations, *J. Sci. Comput*, vol. 67, no. 3, pp. 1043–1065, 2016.