Compact implicit integration factor method for two-dimensional space-fractional advection-diffusion-reaction equations

Huanyan Jian$^{1,3}$, Tingzhu Huang$^1$, Xianming Gu$^2$ and Yongliang Zhao$^1$

$^1$School of Mathematical Sciences, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, P. R. China
$^2$School of Economic Mathematics/Institute of Mathematics, Southwestern University of Finance and Economics, Chengdu, Sichuan 611130, P. R. China

$^3$Email: uestc_hyjian@sina.com

Abstract. In this paper, we intend to develop an effective numerical method to solve a class of two-dimensional space-fractional advection-diffusion-reaction equations. After spatially discretizing this equation using the fractional centered difference formula, it leads to a system of nonlinear ordinary differential equations. The compact implicit integration factor method is applied to solve the resulting system to achieve good stability and robustness. Linear stability analysis and numerical experiments are given to verify that the compact implicit integration factor method has excellent efficiency and stability properties.

1. Introduction
Consider a class of two-dimensional space-fractional advection-diffusion-reaction (ADR) equations,
\[
\begin{aligned}
\frac{\partial u(x,y,t)}{\partial t} &= a_1(x) \frac{\partial u}{\partial x} + a_2(y) \frac{\partial u}{\partial y} + d_1(x) \frac{\partial^{\alpha} u}{\partial |x|^\alpha} + d_2(y) \frac{\partial^{\beta} u}{\partial |y|^\beta} + f(u), \quad (x,y) \in \Omega, \quad 0 < t \leq T, \\
u(x,y,0) &= u_0(x,y),
\end{aligned}
\]
(1)
where $a_1, a_2 \in (1,2]$, $\Omega = (a,b) \times (c,d)$, $a_1(x)$ and $a_2(y)$ represent the advection coefficients in the $x$ and $y$ directions, $d_1(x)$ and $d_2(y)$ represent the diffusion coefficients in the $x$ and $y$ directions, respectively. $f(u)$ is the nonlinear reaction term, and $\frac{\partial^{\alpha} u}{\partial |x|^\alpha}$ denotes the Riesz space fractional derivative [1-2] defined as:
\[
\frac{\partial^{\alpha} u}{\partial |x|^\alpha} = \frac{-1}{2 \cos(\alpha \pi /2) \Gamma(2-\alpha)} \int_a^b d^2 \int_a^b \frac{|x-\theta|^{1-\alpha} u(\theta,y,t) d\theta d\theta}{u_x(x,y,t)}, \quad 1 < \alpha < 2,
\]
where $\Gamma(\cdot)$ denotes the Gamma function. $\frac{\partial^{\beta} u}{\partial |y|^\beta}$ can be defined similarly.

Nonlinear ADR equations [3-5] are common mathematical tools widely used to describe many biological, chemical and physical phenomena. For example: the estimation of fish movement parameters [6], early shaping of the vertebrate limb bud [7] and modeling of flow reactors with complex chemical kinetics [8].
Each of the three parts of the ADR system (1), advection, reaction, and diffusion, exhibits different spatial and temporal characteristics. For example, the reaction term usually has severe stiffness \([9-10]\) which imposes a strong stability constraint on the size of the time step. Besides, the diffusion terms are spatial fractional-order operators with non-local characteristics \([11-12]\), which will make the stability of numerical approximations very sensitive and lead to significant computational difficulties, especially for two-dimensional and high-dimensional systems.

In integration factor (IF) methods \([13-16]\) or exponential time differencing (ETD) methods \([17-20]\), the part of the stability constraint due to diffusion can be completely eliminated because the linear diffusions are treated exactly. In order to deal with systems with severely stiff reactions, Nie et al. \([9]\) developed a class of semi-implicit integration factor (IIF) methods, which treats the nonlinear reactions implicitly and maintains the exact treatment of the linear diffusions. For two-dimensional or high-dimensional systems, the compact IIF (cIIF) method \([21-22]\) can be implemented to save the storage costs and improve efficiency. A distinctive feature of the cIIF and IIF methods is the decoupling between the implicit treatment of the nonlinear reaction terms and the exact evaluations of the linear diffusion terms. This leads to excellent stability properties of the cIIF and IIF schemes, in particular, the second-order version is unconditionally linearly stable.

In this paper, we extend the cIIF method to solve the two-dimensional space-fractional ADR system (1). Specifically, we apply the second-order cIIF scheme. In order to obtain the spatial second-order accuracy corresponding to the time direction, we use the second-order difference formula \([23]\) to discretize the advection term, and use the second-order fractional centered difference formula \([24]\) to approximate the Riesz diffusion term. Linear stability analysis and numerical experiments are given to demonstrate the effectiveness and stability of the cIIF method for the problem (1).

The rest of this paper is organized as follows: in Section 2, we derive and formulate the cIIF method for the two-dimensional space-fractional ADR problem. Linear stability analysis is presented in Section 3. In Section 4, we carry out numerical experiments to investigate the performance of the proposed cIIF method. Finally, we draw our conclusions in Section 5.

2. Compact implicit integration factor (cIIF) method
In this section, we introduce the cIIF method for solving the ADR problem (1). We need to first describe the spatial discretization.

Let \(h_1 = \frac{b-a}{M_1}\) and \(h_2 = \frac{d-c}{M_2}\) be the spatial grid sizes in the \(x\) and \(y\) directions, respectively, where \(M_1\) and \(M_2\) are positive integers. Then a spatial partition can be defined as \(x_i = a + ih_1\) for \(i = 0, 1, \ldots, M_1\) and \(y_j = c + jh_2\) for \(j = 0, 1, \ldots, M_2\).

For the discretization of the Riesz space fractional derivative, we apply the fractional centered difference formula \([24]\):

\[
\frac{\partial^\alpha u(x_i, y_j)}{\partial |x|^\alpha} = \frac{1}{h_1^\alpha} \sum_{k=-M_1}^{k=M_1} g_k^{(\alpha)} u(x_i-k, y_j) + O(h_1^2),
\]

\[
\frac{\partial^\beta u(x_i, y_j)}{\partial |y|^\beta} = \frac{1}{h_2^\beta} \sum_{k=-M_2}^{k=M_2} g_k^{(\beta)} u(x_i, y_j-k) + O(h_2^2),
\]

where

\[
g_k^{(\alpha)} = \frac{(-1)^k \Gamma(\alpha + 1)}{\Gamma(\alpha - k + 1) \Gamma(\alpha + k + 1)}.
\]

For the discretization of the advection term, we use the second-order difference scheme \([23]\):

\[
\frac{\partial u(x_i, y_j)}{\partial x} = \frac{u(x_{i+1}, y_j) - u(x_{i-1}, y_j)}{2h_1} + O(h_1^2),
\]

\[
\frac{\partial u(x_i, y_j)}{\partial y} = \frac{u(x_i, y_{j+1}) - u(x_i, y_{j-1})}{2h_2} + O(h_2^2).
\]

Let \(u_{ij}\) be the numerical approximation to \(u(x_i, y_j)\). Then apply the approximations (2), (3), (4) and (5) to the ADR problem (1), and we get the following semi-discrete scheme:
\[ a_1(x_i) \frac{u_{i,j} - u_{i-1,j}}{2h_1} + a_2(y_j) \frac{u_{j+1,i} - u_{j-1,i}}{2h_2} - d_1(x_i) \frac{1}{h_1^2} \sum_{k=-M_1}^{M_1} g_k^{(\alpha)} u_{i-1,k} - d_2(y_j) \frac{1}{h_2^2} \sum_{k=-M_2}^{M_2} g_k^{(\beta)} u_{j,k} = f(u_j). \] 

Denote
\[ U = \begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1,M_2-1} \\ u_{21} & u_{22} & \cdots & u_{2,M_2-1} \\ \vdots & \vdots & \ddots & \vdots \\ u_{M_1-1,1} & u_{M_1-1,2} & \cdots & u_{M_1-1,M_2-1} \end{bmatrix} \]

and
\[ U_0 = \begin{bmatrix} u_0(x_1,y_1) & u_0(x_1,y_2) & \cdots & u_0(x_1,y_{M_2-1}) \\ u_0(x_2,y_1) & u_0(x_2,y_2) & \cdots & u_0(x_2,y_{M_2-1}) \\ \vdots & \vdots & \ddots & \vdots \\ u_0(x_{M_1-1},y_1) & u_0(x_{M_1-1},y_2) & \cdots & u_0(x_{M_1-1},y_{M_2-1}) \end{bmatrix}. \]

We can rewrite the semi-discrete scheme (6) to the following nonlinear ODEs system form:
\[ \frac{dU}{dt} = B_1 U + UB_2 + f(U), \quad U(0) = U_0, \] 

where
\[ B_1 = \frac{A_1 C_1 D_1 G_\alpha}{h_1^2} \]

and
\[ B_2 = \frac{A_2 C_2 D_2 G_\beta}{h_2^2} \]

in which
\[ A_1 = \text{diag} [a_1(x_1), a_1(x_2), \ldots, a_1(x_{M_1-1})], \quad D_1 = \text{diag} [d_1(x_1), d_1(x_2), \ldots, d_1(x_{M_1-1})], \]

\[ C_1 = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ -1 & 0 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -1 & 0_{(M_1-1)} \end{bmatrix} \]

and
\[ G_\alpha = \begin{bmatrix} g_0^{(\alpha)} & g_1^{(\alpha)} & \cdots & g_{M_2-1}^{(\alpha)} \\ g_1^{(\alpha)} & g_0^{(\alpha)} & \cdots & \vdots \\ \vdots & \vdots & \ddots & g_1^{(\alpha)} \\ g_{M_1-2}^{(\alpha)} & \cdots & g_1^{(\alpha)} & g_0^{(\alpha)} \end{bmatrix} \]

the \( A_2, D_2, C_2, G_\beta \) can be similarly defined.

In order to achieve good robustness and stability, we use the cIIF method proposed in \([21]\) to solve the nonlinear ODEs system (7). After multiplying (7) by the integration factor \( e^{B_1^2 \Delta t} \) from the left and \( e^{B_2^2 \Delta t} \) from the right, we integrate the resulting equation over one time step from \( t_n \) to \( t_{n+1} = t_n + \Delta t \) to obtain:
\[ U_{n+1} = e^{B_1^2 \Delta t} U_n e^{B_2^2 \Delta t} + e^{B_1^2 \Delta t} \left[ \int_0^{\Delta t} e^{-B_1^2 \tau} f(U(t_n + \tau)) e^{-B_2^2 \tau} d\tau \right] e^{B_2^2 \Delta t}. \] 

To construct a scheme of \( r \)th order truncation error, we approximate the integrand in (8) with a \((r - 1)\)th order Lagrange polynomial, \( \Psi(\tau) \), at a set of interpolation points \( t_{n+1}, t_n, \ldots, t_{n+2-r} \):
\[ \Psi(\tau) \equiv \sum_{i=1}^{r-2} e^{B_1^2 \Delta t} f(U_{n-i}) e^{B_2^2 \Delta t} \prod_{k=i+1,k \neq i}^{r-2} \frac{\tau + k\Delta t}{(k - i)\Delta t}, \quad 0 \leq \tau \leq \Delta t. \]

Hence the \( r \)th order cIIF schemes are
\[ U_{n+1} = e^{B_1 \Delta t} U_n e^{B_2 \Delta t} + \Delta t \left[ \alpha_1 f(U_{n+1}) + \sum_{i=0}^{r-2} \alpha_{-i} e^{(i+1)B_1 \Delta t} f(U_{n-i}) e^{(i+1)B_2 \Delta t} \right], \]  

(9)

with \( \alpha_1, \alpha_0, \alpha_{-1}, \cdots, \alpha_{-r} \) defined as

\[
\alpha_{-i} = \frac{1}{\Delta t} \int_0^{\Delta t} \prod_{k=-1, k \neq i}^{r-2} \frac{\tau + k \Delta t}{(k - i) \Delta t} \, d\tau, \quad -1 \leq i \leq r - 2.
\]

In Table 2 of [21], the values of coefficients, \( \alpha_{-i} \), for the schemes (9) with order up to four are listed. In particular, the second-order cIIF (cIIF2) scheme is of the following form:

\[ U_{n+1} = e^{B_1 \Delta t} [U_n + \frac{\Delta t}{2} f(U_n)] e^{B_2 \Delta t} + \frac{\Delta t}{2} f(U_{n+1}). \]

(10)

3. Linear stability analysis

In order to analyse the linear stability of the cIIF scheme, we use the following scalar linear test equation

\[ u_t = au - du + ru, \quad \text{with} \quad d > 0, \]

(11)

where \( a, d, \) and \( r \) respectively represent the spatial discretizations of the advection, diffusion, reaction terms. Similar to the stability analysis method in [21], we will show the boundaries of the stability regions in the complex plane for the cIIF2 scheme, a family of curves for different \( a \Delta t \) and \( \Delta t \).

Applying the cIIF2 (10) to equation (11), then substituting \( u_n = e^{i \theta} \) into the resulting equation, we get

\[
(1 - \frac{\lambda}{2}) e^{i \theta} = e^{a \Delta t} (1 + \frac{\lambda}{2}) e^{-d \Delta t},
\]

with \( \lambda = r \Delta t \). Let \( \lambda_r \) and \( \lambda_i \) be the real and imaginary parts of \( \lambda \), respectively. Thus, the equations for \( \lambda_r \) and \( \lambda_i \) are

\[
\begin{align*}
\lambda_r &= \frac{2(1 - e^{2(a-d)\Delta t})}{(1 - e^{(a-d)\Delta t})^2 + 2(1 + \cos \theta) e^{(a-d)\Delta t}}, \\
\lambda_i &= \frac{4 \sin \theta e^{(a-d)\Delta t}}{(1 - e^{(a-d)\Delta t})^2 + 2(1 + \cos \theta) e^{(a-d)\Delta t}}.
\end{align*}
\]

Figure 1. Stability regions (exterior of the closed curves) for cIIF2 scheme.
Figure 1 gives the stability regions (exterior of the closed curves) of $\lambda$ for different values of $a\Delta t$ and $d\Delta t$. Here we only plot the experimental results when $a < 0$. It can be seen from Figure 1 that: if $a\Delta t$ is fixed, the stable region becomes larger as the value of $d\Delta t$ increases. When $d\Delta t \to 0$, the stability region will coincide with the domain $\lambda_r < 0$. When $d\Delta t \to \infty$, the stability region will be the entire complex plane except for the point $(2,0)$. If $d\Delta t$ is fixed, the stable region also becomes bigger with the increase of the value of $|a|\Delta t$.

4. Numerical experiments

In this section, some numerical experiments are presented to demonstrate the effectiveness of the cIIF scheme. In the implementation of the cIIF method, the matrix exponential is computed using the "expm" method in Matlab, and the local nonlinear system is solved using the fixed point iteration method. All numerical experiments are implemented using MATLAB R2018a on a desktop with 16GB RAM, Intel (R) Core (TM) i7-4790K CPU @3.60GHz.

Example 1 Consider the following two-dimensional space-fractional ADR problem:

$$\frac{\partial u(x,y,t)}{\partial t} = -0.01 \frac{\partial u}{\partial x} - 0.01 \frac{\partial u}{\partial y} + \frac{\partial^a u}{\partial |x|^a} + \frac{\partial^\beta u}{\partial |y|^\beta} + u, \quad (x,y) \in \Omega, \quad 0 < t \leq 1,$$

$$u(x,y,t) = 0, \quad (x,y) \in \partial \Omega, \quad 0 \leq t \leq 1,$$

$$u(x,y,0) = x^2(1-x)y^2(1-y)^2, \quad (x,y) \in \Omega,$$

with $\Omega = (0,1) \times (0,1)$.

Denote $U(h_1,h_2,\Delta t)$ as the numerical solution of space grids $h_1 \times h_2$ and time grid $\Delta t$. The error in the temporal direction with sufficiently small $h_1 \times h_2$ is calculated by

$$e(\Delta t) = ||U(h_1,h_2,\Delta t) - U(h_1,h_2,\Delta t/2)||_{\infty},$$

and the error in the spatial direction with sufficiently small time step $\Delta t$ is similarly calculated by

$$e(h_1,h_2) = ||U(h_1,h_2,\Delta t) - U(h_1/2,h_2/2,\Delta t)||_{\infty}.$$

Then the corresponding convergence orders in time and space are computed by

$$order_{\Delta t} = \log_2 \left( \frac{e(\Delta t)}{e(\Delta t/2)} \right)$$

and

$$order_h = \log_2 \left( \frac{e(h_1,h_2)}{e(h_1/2,h_2/2)} \right),$$

respectively.

Table 1 displays the time errors and convergence orders of cIIF2 scheme with different time steps and $\alpha = \beta$. In Table 2, the space errors and convergence orders of cIIF2 scheme are given for different space grids and values of $\alpha = \beta$. From Tables 1-2, we conclude that the time and space convergence orders of IIF2 scheme (10) are second-order as anticipated.

Table 1. Maximum errors and time convergence orders of cIIF2 scheme for Example 1 with $M_1 = M_2 = 2^6$.

| N   | $\alpha = \beta = 1.2$ | $\alpha = \beta = 1.5$ | $\alpha = \beta = 1.8$ |
|-----|------------------------|------------------------|------------------------|
|     | $\text{error}$ | $\text{order}_{\Delta t}$ | $\text{error}$ | $\text{order}_{\Delta t}$ | $\text{error}$ | $\text{order}_{\Delta t}$ |
| $2^6$ | 3.190123E-09 | --- | 2.021076E-09 | --- | 1.274069E-09 | --- |
| $2^7$ | 7.975019E-10 | 2.0001 | 5.054059E-10 | 1.9996 | 3.192195E-09 | 1.9996 |
| $2^8$ | 1.993737E-10 | 2.0000 | 1.263914E-10 | 1.9995 | 7.981210E-11 | 1.9999 |
| $2^9$ | 4.984377E-11 | 2.0000 | 3.159775E-11 | 2.0000 | 1.995836E-11 | 1.9996 |

Table 2. Maximum errors and space convergence orders of cIIF2 scheme for Example 1 with $N = 2^6$.

| $M_1 = M_2$ | $\alpha = \beta = 1.2$ | $\alpha = \beta = 1.5$ | $\alpha = \beta = 1.8$ |
|--------------|------------------------|------------------------|------------------------|
|              | $\text{error}$ | $\text{order}_h$ | $\text{error}$ | $\text{order}_h$ | $\text{error}$ | $\text{order}_h$ |
| $2^1$        | 3.254559E-04 | --- | 4.038579E-04 | --- | 4.846894E-04 | --- |
| $2^2$        | 6.557131E-05 | 2.3113 | 8.350099E-05 | 2.2740 | 1.028057E-04 | 2.2371 |
| $2^3$        | 2.149065E-05 | 1.6094 | 1.907304E-05 | 2.1302 | 2.406567E-05 | 2.0949 |

Example 2 Consider the two-dimensional space-fractional ADR problem as follows:
\[
\begin{aligned}
\frac{\partial u(x,y,t)}{\partial t} &= -0.01 \frac{\partial u}{\partial x} - 0.01 \frac{\partial u}{\partial y} + \frac{\partial^2 u}{\partial |x|^\alpha} + \frac{\partial^2 u}{\partial |y|^\beta} + u(1 - u), \quad (x, y) \in \Omega, \quad 0 < t \leq 1, \\
\frac{\partial u(x,y,t)}{\partial t} &= 0, \quad (x, y) \in \partial \Omega, \quad 0 \leq t \leq 1, \\
\frac{\partial u(x,y,0)}{\partial t} &= x^2(1 - x)^2y^2(1 - y)^2, \quad (x, y) \in \Omega,
\end{aligned}
\]

where \(\Omega = (0,1) \times (0,1)\).

**Table 3.** Maximum errors and time convergence orders of cIIF2 scheme for Example 2 with \(M_1 = M_2 = 2^6\).

| N     | \(\alpha = \beta = 1.2\) | \(\alpha = \beta = 1.5\) | \(\alpha = \beta = 1.8\) |
|-------|----------------------------|---------------------------|---------------------------|
| \(2^6\) | 2.952336E-09 | 1.592671E-09 | 6.531311E-10 |
| \(2^7\) | 7.379599E-10 | 3.976077E-10 | 2.000000E-06 |
| \(2^8\) | 1.844822E-05 | 9.940213E-11 | 2.000000E-06 |
| \(2^9\) | 4.612037E-01 | 4.216913E-11 | 2.000000E-06 |

**Table 4.** Maximum errors and space convergence orders of cIIF2 scheme for Example 2 with \(N = 2^6\).

| \(M_1 = M_2\) | \(\alpha = \beta = 1.2\) | \(\alpha = \beta = 1.5\) | \(\alpha = \beta = 1.8\) |
|----------------|---------------------------|---------------------------|---------------------------|
| \(2^1\) | 3.252359E-04 | 4.036719E-04 | 4.845546E-04 |
| \(2^2\) | 6.553647E-05 | 8.347382E-05 | 1.027857E-04 |
| \(2^3\) | 2.148687E-05 | 1.907121E-05 | 2.406119E-05 |

Tables 3 and 4 give the accuracy order of the cIIF2 scheme in time and space directions for Example 2, respectively. From which, we can see that the accuracy order of (10) is second-order in both time and space directions.

**Table 5.** CPU(s) comparisons of IIF2, CN, and cIIF2 schemes for Example 2 with \(N = M_1 = M_2\).

| N     | IIF2 | cIIF2 | IIF2 | CN | cIIF2 | IIF2 | CN | cIIF2 |
|-------|------|-------|------|----|-------|------|----|-------|
| \(2^5\) | 0.362 | 0.002 | 0.322 | 0.152 | 0.002 | 0.471 | 0.150 | 0.002 |
| \(2^6\) | 9.894 | 0.012 | 11.389 | 1.157 | 0.012 | 13.685 | 1.165 | 0.012 |
| \(2^7\) | 658.013 | 11.107 | 850.561 | 11.644 | 0.064 | 956.955 | 13.474 | 0.064 |
| \(2^8\) | OOM | 95.857 | 103.648 | 95.857 | 1.048 | OOM | 116.208 | 0.396 |

Table 5 shows the CPU time for solving Example 2 using the IIF2 [9], CN [25], and cIIF2 schemes, where “OOM” means out of memory. As seen in Table 5, Our proposed cIIF2 method is better than the CN scheme, and much better than the IIF2 scheme in terms of CPU time.

5. **Conclusions**

Implicit integration factor (IIF) method and its high-dimensional simulation compact IIF (cIIF) are a class of effective time-stepping methods for stiff diffusion-reaction equations. In this paper, we extend the cIIF method to solve the two-dimensional space-fractional advection-diffusion-reaction (ADR) equations. The linear stability is analysed in Section 3, which shows that our cIIF method has good stability. Numerical results also demonstrate the effectiveness of our proposed method.

**Acknowledgments**

This work is supported by NSFC (61702083, 11801463 and 11701467) and the Fundamental Research Funds for the Central Universities (JBK1902028 and ZYGX2016J138).

**References**

[1] Jian H Y, Huang T Z, Zhao X L et al. 2019 Fast second-order accurate difference schemes for
time distributed-order and Riesz space fractional diffusion equations \textit{J. Appl. Anal. Comput.} \textbf{9} 1359-1392

[2] Zhu X, Nie Y, Wang J et al. 2017 A numerical approach for the Riesz space-fractional Fisher’ equation in two-dimensions \textit{Int. J. Comput. Math.} \textbf{94} 296-315

[3] Hundsdorfer W and Vermeer J 2003 \textit{Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations} (Springer-Verlag)

[4] Hauke G 2002 A simple subgrid scale stabilized method for the advection-diffusion-reaction equation \textit{Comput. Methods Appl. Mech. Engrg.} \textbf{191} 2925-2947

[5] Ropp D L and Shadid J N 2009 Stability of operator splitting methods for systems with indefinite operators: advection-diffusion-reaction systems \textit{J. Comput. Phys.} \textbf{228} 3508-3516

[6] Fournier D A, Sibert J R, Hampton J et al. 1999 An advection-diffusion-reaction model for the estimation of fish movement parameters from tagging data, with application to skipjack tuna (Katsuwonus pelamis) \textit{Can. J. Fish. Aquat. Sci.} \textbf{56} 925-938

[7] Dillon R and Othmer H G 1999 A mathematical model for outgrowth and spatial patterning of the vertebrate limb bud \textit{J. Theor. Biol.} \textbf{197} 295-330

[8] Dryer F L, Haas F M, Santner J et al. 2014 Interpreting chemical kinetics from complex reaction-advection-diffusion systems: Modeling of flow reactors and related experiments \textit{Prog. Energ. Combust.} \textbf{44} 19-39

[9] Nie Q, Zhang Y T and Zhao R 2006 Efficient semi-implicit schemes for stiff systems \textit{J. Comput. Phys.} 214 521-537

[10] Zhao S, Ovadia J, Liu X F et al. 2011 Operator splitting implicit integration factor methods for stiff reaction-diffusion-advection systems \textit{J. Comput. Phys.} \textbf{230} 5996-6009

[11] Meerschaert M M and Tadjeran C 2004 Finite difference approximations for fractional advection-dispersion flow equations \textit{J. Comput. Appl. Math.} \textbf{172} 65-77

[12] Chen S, Liu F, Turner I et al. 2018 A fast numerical method for two-dimensional Riesz space fractional diffusion equations on a convex bounded region \textit{Appl. Numer. Math.} \textbf{134} 66-80

[13] Ta C, Wang D and Nie Q 2015 An integration factor method for stochastic and stiff reaction-diffusion systems \textit{J. Comput. Phys.} \textbf{295} 505-522

[14] Ahmed S and Liu X 2019 High order integration factor methods for systems with inhomogeneous boundary conditions \textit{J. Comput. Appl. Math.} \textbf{348} 89-102

[15] Wang D, Zhang L and Nie Q 2014 Array-representation integration factor method for high-dimensional systems \textit{J. Comput. Phys.} \textbf{258} 585-600

[16] Wang D, Chen W and Nie Q 2015 Semi-implicit integration factor methods on sparse grids for high-dimensional systems \textit{J. Comput. Phys.} \textbf{292} 43-55

[17] Du Q and Zhu W 2004 Stability analysis and applications of the exponential time differencing schemes \textit{J. Comput. Math.} \textbf{22} 200-209

[18] Du Q and Zhu W 2005 Analysis and applications of the exponential time differencing schemes and their contour integration modifications \textit{BIT Numer. Math.} \textbf{45} 307-328

[19] Kassam A K and Trefethen L N 2005 Fourth-order time stepping for stiff PDEs \textit{SIAM J. Sci. Comp.} \textbf{26} 1214-1233

[20] Zhu L, Ju L and Zhao W 2016 Fast high-order compact exponential time differencing Runge-Kutta methods for second-order semilinear parabolic equations \textit{J Sci. Comput.} \textbf{67} 1043-1065

[21] Nie Q, Wan F Y M, Zhang Y T, et al. 2008 Compact integration factor methods in high spatial dimensions \textit{J. Comput. Phys.} \textbf{227} 5238-5255

[22] Liu X and Nie Q 2010 Compact integration factor methods for complex domains and adaptive mesh refinement \textit{J. Comput. Phys.} \textbf{229} 5692-5706

[23] Gu X M, Huang T Z, Ji C C et al. 2017 \textit{J. Sci. Comput.} \textbf{72} 957-985

[24] Ye H, Liu F, Anh V et al. 2015 \textit{IMA J. Appl. Math.} \textbf{80} 825-838

[25] Kadalbajoo M K and Awasthi A 2006 \textit{Appl. Math. Comput.} \textbf{182} 1430-1442