COMPUTATIONAL SCIENCE | RESEARCH ARTICLE

New group iterative schemes in the numerical solution of the two-dimensional time fractional advection-diffusion equation

Alla Tareq Balasim1,2* and Norhashidah Hj. Mohd. Ali1

Abstract: Numerical schemes based on small fixed-size grouping strategies have been successfully researched over the last few decades in solving various types of partial differential equations where they have been proven to possess the ability to increase the convergence rates of the iteration processes involved. The formulation of these strategies on fractional differential equations, however, is still at its infancy. Appropriate discretization formula will need to be derived and applied to the time and spatial fractional derivatives in order to reduce the computational complexity of the schemes. In this paper, the design of new group iterative schemes applied to the solution of the 2D time fractional advection-diffusion equation are presented and discussed in detail. The Caputo fractional derivative is used in the discretization of the fractional group schemes in combination with the Crank–Nicolson difference approximations on the standard grid. Numerical experiments are conducted to determine the effectiveness of the proposed group methods with regard to execution times, number of iterations, and computational complexity. The stability and convergence properties are also presented using a matrix method with mathematical induction. The numerical results will be proven to agree with the theoretical claims.

ABOUT THE AUTHORS

Alla Tareq Balasim is a lecturer at the Department of Mathematics Almustansiria University, Iraq. He received his Master in Mathematics from the Department of Mathematics, Baghdad University, Iraq. He is currently a PhD student and his research interests are numerical differential equations and dynamical system.

Norhashidah Hj. Mohd. Ali is a Professor at the School of Mathematical Sciences, Universiti Sains Malaysia. She received her PhD in Industrial Computing from Universiti Kebangsaan Malaysia. Her areas of expertise include numerical differential equations and parallel numerical algorithms. She has over 25 years teaching and research experiences and has written more than 100 articles in these areas.

PUBLIC INTEREST STATEMENT

Many phenomena in science and engineering can be modelled as two dimensional fractional advection-diffusion equations satisfying appropriate initial and boundary conditions. With the advent of computing technology, effective numerical methods have been extensively formulated in solving these equations due to their simplicity and accuracy. In particular, finite difference formulas, are oftenly used to numerically discretize the differential equations to produce sparse systems of linear equations which may be suitably solved by iterative solvers. However, iterative solvers have the disadvantage of being too slow to converge which may increase the computation timings. To overcome this problem, suitable small fixed-size grouping strategies are applied to the mesh points of the solution domain following the discretization of the differential equation which results in schemes with faster convergence and lower computational complexity with comparable accuracies. This is very useful to engineers and scientists who are involved in time consuming simulation modeling processes.
1. Introduction

Fractional calculus, which is the calculus of integrals and derivatives in random order, dates back as far as the more popular integer calculus, and has been gaining significant interest over the past few years with its history and development being explored in detail by Oldham and Spanier (1974), Miller and Ross (1993), Samko, Kilbas, and Marichev (1993) and Podlubny (1998). Fractional differential equations (FDEs) can be used to model many problems in a wide field of applications. They are defined as equations that utilize fractional derivatives and considered as powerful tools that can describe the memory and hereditary characteristics of various materials. Several researchers have explored the use of FDEs in the fields of chemistry (Gorenflo, Mainardi, Moretti, Pagnini, & Paradisi, 2002; Seki, Wojcik, & Tachiya, 2003), physics (Henry & Wearne, 2000; Metzler, Barkai, & Klafter, 1999; Metzler & Klafter, 2000; Wyss, 1986) and other scientific and engineering spheres (Baeumer, Benson, Meerschaert, & Wheatcraft, 2001; Benson, Wheatcraft, & Meerschaert, 2000; Cushman & Ginn, 2000; Mehdinejadiani, Naseri, Jafari, Ghanbarzadeh, & Baleanu, 2013). Since there are at most no exact solutions to the majority of fractional differential equations, it is necessary to resort to approximation and numerical methods (Abdelkawy, Zaky, Bhrawy, & Baleanu, 2015; Balasim & Ali, 2015; Baleanu, Agheli, & Al Qurashi, 2016; Bhrawy & Baleanu, 2013). Over the past decade, there has been an influx of numerical methods development for solving various types of FDEs (Agrawal, 2008; Ali, Abdullah, & Mohyud-Din, 2017; Chen, Deng, & Wu, 2013; Chen & Liu, 2008; Chen, Liu, Anh, & Turner, 2013; Chen, Liu, & Burrage, 2008; Leonenko, Meerschaert, & Sikorskii, 2013; Li, Zeng, & Liu, 2012; Liu, Zhuang, Anh, Turner, & Burrage, 2007; Shen, Liu, & Anh, 2011; Sousa & Li, 2015; Su, Wang, & Wang, 2013; Uddin & Haq, 2011; Zhang, Huang, Feng, & Wei, 2013; Zheng, Li, & Zhao, 2010; Zhuang, Gu, Liu, Turner, & Yarlagadda, 2011; Zhuang, Liu, Anh, & Turner, 2009).

Chen et al. (2008) used implicit and explicit difference techniques to solve time fractional reaction-diffusion equations, while Liu et al. (2007) proposed for these techniques to be employed in solving the space–time fractional advection-dispersion equation by replacing the first-order time derivative by the Caputo fractional derivative, and the first- and second-order space derivatives by the Riemman–Liouville fractional derivatives. The use of radial basis function (RBF) approximation method was discussed in Uddin and Haq (2011) to solve the time fractional advection-dispersion equation in a bounded domain. The application of finite element method was also seen in Zheng et al. (2010), in solving the space fractional advection-diffusion equation under non-homogeneous initial boundary conditions. In Chen et al. (2011), a numerical method with first-order temporal accuracy and second-order spatial accuracy was established for solving the variable order Galilei advection-diffusion equation with a nonlinear source term, while Zhuang et al. (2009) used the explicit and implicit Euler approximation to solve a variable order fractional advection-diffusion equation with a nonlinear source term. A new numerical solution for the 2D fractional advection-dispersion equation with variable coefficients in a finite field was also introduced by Chen and Liu (2008). In 2011, Shen et al. (2011) proposed the explicit and implicit finite difference approximations for the space-time Riesz-Caputo fractional advection-diffusion equation where the implicit scheme was proven to be unconditionally stable, while the explicit scheme exhibits a conditionally stable property. The development of an implicit meshless method was also seen in Zhuang et al. (2011) in solving the time-dependent fractional advection-diffusion equation where a discretized system of equations was obtained using the moving least squares (MLS) meshless shape functions.

In general, finding numerical solutions to FDEs using iterative finite difference schemes is not a straightforward process and can be a challenging task due to several reasons. Firstly, all the earlier solutions have to be saved if the current solution is to be computed, making the calculations even more complex and costly in terms of CPU usage time in cases where traditional point implicit difference approaches are used. Secondly, although the point implicit techniques are stable, a significant
amount of CPU usage time is required when many unknowns are involved. In recent decades, group-
ing strategies have been proven to possess characteristics that are able to reduce the spectral radius
of the generated matrix resulting from the finite difference discretization of the differential equation,
and therefore increase the convergence rates of the iterative algorithms. They have been shown
to reduce the computational timings compared to their point-wise counterparts in solving several
types of partial differential equation (Ali & Kew, 2012; Evans & Yousif, 1986; Kew & Ali, 2015;
Ng & Ali, 2008; Othman & Abdullah, 2000; Tan, Ali, & Lai, 2012; Yousif & Evans, 1995). These group
methods are also easy to implement and suitable to be implemented on parallel computers due to
their explicit nature. However, till date these strategies have not been tested on solving FDEs, par-
ticularly for the 2D cases. Therefore, in this study, the formulation of new group iterative methods is
presented in solving the following 2D time fractional advection diffusion equation

$$\frac{\partial^\alpha u}{\partial t^\alpha} = a_x \frac{\partial^2 u}{\partial x^2} + a_y \frac{\partial^2 u}{\partial y^2} - b_x \frac{\partial u}{\partial x} - b_y \frac{\partial u}{\partial y} + f(x, y, t),$$

(1)

where $0 < \alpha < 1$, $a_x$, $a_y$, $b_x$, $b_y$ are positive constants and $f(x, y, t)$ is nonhomogeneous term sub-
jected to the following initial and Dirichlet boundary conditions

$$u(x, y, 0) = g(x, y)$$

$$u(0, y, t) = g_1(y, t) \quad u(1, y, t) = g_2(y, t)$$

$$u(x, 0, t) = g_3(x, t) \quad u(x, 1, t) = g_4(x, t)$$

This equation plays an important role in describing transport dynamics in complex systems which
are governed by anomalous diffusion and non-exponential relaxation patterns (Zhuang, Gu, Liu,
Turner, & Yarlagadda, 2011). This paper is outlined as follows. Section 2 presents the proposed group
iterative methods obtained from the Crank–Nicolson difference approximation followed by the sta-
tional derivatives are available in Podlubny (1998).

2. Standard approximation schemes for fractional advection-diffusion equations
The Caputo fractional derivative, $D^\alpha$, of the order-$\alpha$ is expressed as follows (Li, Qian, & Chen, (2011):

$$D^\alpha = \frac{1}{\Gamma(m - \alpha)} \int_0^t \frac{f^m(t)}{(x - t)^{\alpha-m+1}} dt, \quad m - 1 < \alpha < m, \quad m \in N, \quad x > 0,$$

(2)

where $\Gamma(\cdot)$ is the Euler Gamma function. Further details on the definitions and properties of frac-
tional derivatives are available in Podlubny (1998).

We need to apply appropriate finite difference approximations to the time and space derivatives of
(1), let $h > 0$ be the space step and $k > 0$ be the time step. The domain is assumed to be uniform in
both $x$ and $y$ directions. Define $x_i = ih$, $y_j = jh$, $i, j = 0, 1, ..., n$, and a mesh size of $h = \frac{1}{n}$ where $n$
is an arbitrary positive integer and $t_k = kr$, $k = 0, 1, ..., l$. Various approximations formulas could be
obtained for (1) at the point of $(x_i, y_j, t_k)$. By taking the average of the central difference approxima-
tions to the left side of (1) at the points $(i, j, k + 1)$ and $(i, j, k)$, the Caputo time fractional approxima-
tion (2) can be transformed to the following form (Karatay, Kale, & Bayramoglu, 2013)

$$\frac{\partial^\alpha u(x, y, t)}{\partial t^\alpha} = w_i u_i^k + \sum_{s=0}^{k-1} [w_{k-s+1} - w_{k-s}] u_s^k - w_i u_i^0 + \sigma (u_i^{k+1} - u_i^k) + O(t^{1-\alpha}),$$

(3)

where $\sigma = \frac{1}{\Gamma(1-\alpha)}$, $w_i = \sigma(s + \frac{1}{2})^{1-\alpha} - (s - \frac{1}{2})^{1-\alpha}$.

Using (3) in combination with the second-order Crank-Nicolson difference scheme for the right
side of (1) will result in the following fractional standard point (FSP) formula

---
\[ w_i u^k + \sum_{s=1}^{k+1} \left[ w_{k+s-1} - w_{k-s} \right] u_{ij}^k - w_i u_0^k + \sigma \frac{\left( \frac{u_{ij}^k - u_{ij}^{k-1}}{2h} \right)}{h^2} \]

\[ = \frac{\sigma}{2} \left( \frac{u_{ij}^{k-1} - 2u_{ij}^k + u_{ij}^{k+1}}{h^2} \right) + \frac{b_j}{2} \left( \frac{u_{ij}^{k-1} - u_{ij}^{k+1}}{2h} \right) \]

\[ - \frac{b_j}{2} \left( \frac{u_{ij}^{k-1} - u_{ij}^{k+1}}{2h} \right) + f_{ij}^{k+\frac{1}{2}} + O(\varepsilon^{2-\alpha}) + (\Delta x)^2 + (\Delta y)^2. \]

Equation (4) can be simplified to become as follows:

\[ (1 + s_x + s_y) u_{ij}^{k+1} = \left( \frac{s_x}{2} + \frac{c_x}{4} \right) u_{ij}^{k+1} + \left( \frac{s_y}{2} + \frac{c_y}{4} \right) u_{ij}^{k+1} + \left( \frac{s_x}{2} - \frac{c_x}{4} \right) u_{ij}^{k-1} + \left( \frac{s_y}{2} - \frac{c_y}{4} \right) u_{ij}^{k-1} \]

\[ + \left( \frac{s_x}{2} - \frac{c_x}{4} \right) u_{ij}^{k+1} + \left( 1 - 2^{1-\alpha} w_i^k - s_x - s_y \right) u_{ij}^{k+1} + \left( \frac{s_x}{2} + \frac{c_x}{4} \right) u_{ij}^{k-1} + (\Delta x)^2 + \frac{4}{3} \sum_{s=1}^{k+1} \left[ w_{k+s} - w_{k-s+1} \right] u_{ij}^{k+1} + m_{ij}^{k+1/2} \]

**Figure 1. Computational molecule of the standard point approximation.**
where \( m_0 = 2^{1-a} t^a (2 - a) \), \( w^*_n = \left( (s + 1/2)^{1-a} - (s - 1/2)^{1-a} \right) \), \( s_x = a_m, s_y = a_{m_0}, c_x = b_{m_0}, c_y = b_{m_0} \).

Figure 1 shows the computational molecule for Equation (5).

### 2.1. Fractional explicit group method

In formulating the fractional explicit group (FEG) method, (5) is applied to any group of four points in the solution domain to generate a \( 4 \times 4 \) system of equation as follows

\[
\begin{pmatrix} D & -a_1 & 0 & -b_1 \\ -a_2 & D & -b_1 & 0 \\ 0 & -b_2 & D & -a_2 \\ -b_2 & 0 & -a_1 & D \end{pmatrix} \begin{pmatrix} u_{i,j} \\ u_{i+1,j} \\ u_{i+1,j+1} \\ u_{i,j+1} \end{pmatrix} = \begin{pmatrix} \text{rhs}_{ij} \\ \text{rhs}_{i+1,j} \\ \text{rhs}_{i+1,j+1} \\ \text{rhs}_{ij+1} \end{pmatrix},
\]

(6)

where \( D = 1 + s_x + s_y, a_1 = \left( \frac{s_x}{2} - \frac{s_y}{4} \right), b_1 = \left( \frac{s_x}{2} - \frac{s_y}{4} \right), a_2 = \left( \frac{s_x}{2} + \frac{s_y}{4} \right), b_2 = \left( \frac{s_x}{2} + \frac{s_y}{4} \right), R = (1 - 2^{1-a} w^*_n - s_x - s_y),

\[
\text{rhs}_{ij} = a_1 (u^k_{i+1,j} + u^k_{i,j+1}) + b_2 (u^k_{i+1,j-1} + u^k_{i,j-1}) + a_1 u^k_{i+1,j} + b_2 u^k_{i,j+1} \\
+ 2^{1-a} w^*_n u^0_{i,j} + Ru^k_{i,j} + \sum_{s=1}^{k-1} 2^{1-a} \left[ w^*_k - w^*_s \right] u^s_{i,j} + m_{o}^{k+1/2},
\]

\[
\text{rhs}_{i+1,j} = a_1 (u^k_{i+2,j} + u^k_{i,j+1}) + b_2 (u^k_{i+1,j-1} + u^k_{i,j-1}) + b_1 u^k_{i+1,j+1} + a_2 u^k_{i,j} \\
+ 2^{1-a} w^*_n u^0_{i+1,j} + Ru^k_{i+1,j} + \sum_{s=1}^{k-1} 2^{1-a} \left[ w^*_k - w^*_s \right] u^s_{i+1,j} + m_{o}^{k+1/2},
\]

\[
\text{rhs}_{i+1,j+1} = a_1 (u^k_{i+2,j+1} + u^k_{i,j+2}) + b_1 (u^k_{i+1,j+1} + u^k_{i+1,j}) + b_2 u^k_{i,j+1} + a_2 u^k_{i,j+1} \\
+ 2^{1-a} w^*_n u^0_{i+1,j+1} + Ru^k_{i+1,j+1} + \sum_{s=1}^{k-1} 2^{1-a} \left[ w^*_k - w^*_s \right] u^s_{i+1,j+1} + m_{o}^{k+1/2},
\]

\[
\text{rhs}_{ij+1} = a_2 (u^k_{i,j+1} + u^k_{i-1,j}) + b_1 (u^k_{i+1,j+2} + u^k_{i,j+2}) + a_1 u^k_{i+1,j+1} + b_2 u^k_{i,j} \\
+ 2^{1-a} w^*_n u^0_{i,j+1} + Ru^k_{i,j+1} + \sum_{s=1}^{k-1} 2^{1-a} \left[ w^*_k - w^*_s \right] u^s_{i,j+1} + m_{o}^{k+1/2}.
\]

Mathematical software can be used to easily invert (6) to obtain the FEG formula.
In this new method, we consider the nodal points with grid size spacing of even \( n \), the solutions at the ungrouped points near the boundaries are computed using (5).

Note that if \( n \) is even, there will be ungrouped points near the upper and right sides of the boundary.

2h-spaced points. Using the Caputo time fractional approximation (3) at the left-hand side of (1) and tional formula is generated through the application specific finite difference approximations with

\[
\begin{aligned}
  &r_1 = \frac{1}{16} \left( c_x - 3s_x \right) \left( c_x - c_y \right) \left( 1 + s_x + s_y \right), \\
  &r_2 = -\frac{1}{64} \left( c_x + c_y \right) \left( c_x + c_y \right) \left( 1 + s_x + s_y \right), \\
  &r_3 = \frac{1}{8} \left( c_x + 2s_x \right) \left( c_y + 2s_y \right) \left( 1 + s_x + s_y \right), \\
  &r_4 = \frac{1}{8} \left( c_x + 2s_x \right) \left( c_y - 2s_y \right) \left( 1 + s_x + s_y \right), \\
  &r_5 = \frac{1}{64} \left( c_y + 2s_y \right) \left( -c_x + c_y \right) \left( 1 + s_x + s_y \right), \\
  &r_6 = -\frac{1}{8} \left( c_x - 2s_x \right) \left( c_y + 2s_y \right) \left( 1 + s_x + s_y \right), \\
  &r_7 = \frac{1}{8} \left( c_x - 2s_x \right) \left( 1 + s_x + s_y \right) \left( c_y + 2s_y \right), \\
  &r_8 = \frac{1}{64} \left( c_y - 2s_y \right) \left( -c_y + c_x \right) \left( 1 + s_x + s_y \right), \\
  &r_9 = -\frac{1}{8} \left( c_x - 2s_x \right) \left( 1 + s_x + s_y \right) \left( c_y - 2s_y \right).
\end{aligned}
\]

Figure 2 shows the construction of blocks of four points in the solution domain for the case \( n = 10 \).

Note that if \( n \) is even, there will be ungrouped points near the upper and right sides of the boundary. The FEG method proceeds with the iterative evaluation of solutions in these blocks of four points using Equation (7) throughout the whole solution domain until convergence is achieved. For the case of even \( n \), the solutions at the ungrouped points near the boundaries are computed using (5).

2.2. The fractional modified explicit group method

In this new method, we consider the nodal points with grid size spacing \( 2h = \frac{\Delta}{n} \). The standard fractional formula is generated through the application specific finite difference approximations with 2h-spaced points. Using the Caputo time fractional approximation (3) at the left-hand side of (1) and the second order Crank–Nicolson difference scheme with 2h-spaced points at the right hand side of (1), the following approximation formula is obtained:

\[
\begin{aligned}
  &w_k u^k + \sum_{s=1}^{k-1} \left[ w_{k-s} - w_{k-s-1} \right] u^s_j - w_k u^0_j + \sigma \frac{(u^0_j - u^1_j)}{2^{\alpha + 1}} \\
  &= \frac{a}{2} \left( \frac{u^0_{i+1,j} - 2u^1_{i+1,j} + u^2_{i+1,j}}{4h^2} + \frac{u^0_{i-1,j} - 2u^1_{i-1,j} + u^2_{i-1,j}}{4h^2} \right) \\
  &+ \frac{a}{2} \left( \frac{u^0_{i+1,j} - 2u^1_{i+1,j} + u^2_{i+1,j}}{4h^2} + \frac{u^0_{i-1,j} - 2u^1_{i-1,j} + u^2_{i-1,j}}{4h^2} \right) - \frac{b}{2} \left( \frac{u^0_{i,j+1} - 2u^1_{i,j+1} + u^2_{i,j+1}}{4h} + \frac{u^0_{i,j-1} - 2u^1_{i,j-1} + u^2_{i,j-1}}{4h} \right) \\
  &- \frac{b}{2} \left( \frac{u^0_{i+1,j} - 2u^1_{i+1,j} + u^2_{i+1,j}}{4h} + \frac{u^0_{i-1,j} - 2u^1_{i-1,j} + u^2_{i-1,j}}{4h} \right) + f^{k+1/2}_{i,j} + O(r^{2-\alpha} + (\Delta x)^2 + (\Delta y)^2).
\end{aligned}
\]

On simplification, we obtain the following
\[
\left( 1 + \frac{S_x + S_y}{4} \right) u^{k+1}_{ij} = \left( \frac{S_x}{8} + \frac{c_x}{8} \right) u^{k+1}_{i-2,j} + \left( \frac{S_y}{8} - \frac{c_y}{8} \right) u^{k+1}_{i+2,j} + \left( \frac{S_y}{8} + \frac{c_y}{8} \right) u^{k+1}_{i+2,j} + \left( \frac{S_x}{8} - \frac{c_x}{8} \right) u^{k+1}_{i-2,j} + (1 - 2^{1-a} w_1^* - \frac{S_x + S_y}{4}) u^{k}_{ij} + \left( \frac{S_x}{2} + c_x \right) u^{k-2}_{i-2,j} + (\frac{S_y}{2} - \frac{c_y}{8}) u^{k}_{i+2,j} + 2^{1-a} w_k^* u^{k}_{ij} + 2^{1-a} \sum_{s=1}^{k-1} [w_{k-s}^* - w_{k-s+1}^*] u^{s}_{ij} + m_{0ij}^{k+1/2}.
\]

Applying (9) to any group of four points \((i,j), (i+2,j), (i+2,j+2)\) and \((i,j+2)\) in the solution domain will result in the following \(4 \times 4\) system

\[
\begin{pmatrix}
D_1 & -a_{11} & 0 & -b_{11} \\
-a_{22} & D_1 & -a_{22} & 0 \\
0 & -b_{22} & D_2 & -a_{22} \\
-b_{11} & 0 & -a_{11} & D_2
\end{pmatrix}
\begin{pmatrix}
u_{ij} \\
u_{i+2,j} \\
u_{i+2,j+2} \\
u_{i,j+2}
\end{pmatrix}
= \begin{pmatrix}
\text{rhs}_{ij} \\
\text{rhs}_{i+2,j} \\
\text{rhs}_{i+2,j+2} \\
\text{rhs}_{i,j+2}
\end{pmatrix},
\]

where, \(D_1 = (1 + \frac{S_x + S_y}{4}), a_{11} = (\frac{S_x}{8} - \frac{c_x}{8}), b_{11} = (\frac{S_y}{8} - \frac{c_y}{8}), a_{22} = (\frac{S_x}{8} + \frac{c_x}{8}), b_{22} = (\frac{S_y}{8} + \frac{c_y}{8}), Q = (1 - 2^{1-a} w_1^* - \frac{S_x + S_y}{4})

\[
\text{rhs}_{ij} = a_{22}(u^{k}_{i-2,j} + u^{k}_{i+2,j}) + b_{22}(u^{k}_{i-2,j} + u^{k}_{i+2,j}) + a_{11} u^{k}_{i+2,j} + b_{11} u^{k}_{i+2,j} + 2^{1-a} w_k^* u^{k}_{ij} + Qu^{k}_{ij} + \sum_{s=1}^{k-1} 2^{1-a} [w_{k-s}^* - w_{k-s+1}^*] u^{s}_{ij} + m_{0ij}^{k+1/2},
\]

\[
\text{rhs}_{i+2,j} = a_{11}(u^{k}_{i+2,j} + u^{k}_{i+4,j}) + b_{11}(u^{k}_{i+2,j} + u^{k}_{i+4,j}) + a_{22} u^{k}_{i+2,j} + b_{22} u^{k}_{i+2,j} + 2^{1-a} w_k^* u^{k}_{i+2,j} + Qu^{k}_{i+2,j} + \sum_{s=1}^{k-1} 2^{1-a} [w_{k-s}^* - w_{k-s+1}^*] u^{s}_{i+2,j} + m_{0i+2,j}^{k+1/2},
\]

\[
\text{rhs}_{i+2,j+2} = a_{11}(u^{k}_{i+2,j+2} + u^{k}_{i+4,j+2}) + b_{11}(u^{k}_{i+2,j+2} + u^{k}_{i+4,j+2}) + a_{22} u^{k}_{i+2,j+2} + b_{22} u^{k}_{i+2,j+2} + 2^{1-a} w_k^* u^{k}_{i+2,j+2} + Qu^{k}_{i+2,j+2} + \sum_{s=1}^{k-1} 2^{1-a} [w_{k-s}^* - w_{k-s+1}^*] u^{s}_{i+2,j+2} + m_{0i+2,j+2}^{k+1/2},
\]

\[
\text{rhs}_{i,j+2} = a_{22}(u^{k}_{i-2,j+2} + u^{k}_{i-2,j+2}) + b_{22}(u^{k}_{i-2,j+2} + u^{k}_{i-2,j+2}) + a_{11} u^{k}_{i-2,j+2} + b_{11} u^{k}_{i-2,j+2} + 2^{1-a} w_k^* u^{k}_{i-2,j+2} + Qu^{k}_{i-2,j+2} + \sum_{s=1}^{k-1} 2^{1-a} [w_{k-s}^* - w_{k-s+1}^*] u^{s}_{i-2,j+2} + m_{0i-2,j+2}^{k+1/2}.
\]

The four-point FMEG equation below is obtained by inverting (10), as follows

\[
\begin{pmatrix}
u^{k+1}_{ij} \\
u^{k+1}_{i+2,j} \\
u^{k+1}_{i+2,j+2} \\
u^{k+1}_{i,j+2}
\end{pmatrix}
= \frac{1}{\text{const}} \begin{pmatrix} r_{11} & r_{22} & r_{33} & r_{44} \\
r_{55} & r_{11} & r_{44} & r_{66} \\
r_{77} & r_{88} & r_{11} & r_{55} \\
r_{88} & r_{99} & r_{22} & r_{11}
\end{pmatrix}
\begin{pmatrix}
\text{rhs}_{ij} \\
\text{rhs}_{i+2,j} \\
\text{rhs}_{i+2,j+2} \\
\text{rhs}_{i,j+2}
\end{pmatrix}. 
\]
Figure 3. Grouping of the points for the FMEG method (n = 10).

\[
\text{cons} = \frac{1}{4096}(4096 + c_x^6 + c_y^6 + 4096s_x + 1408s^2_x + 192s^3_x + 9s^4_x + 4096s_y \\
+ 3072s_x s_y + 704s^2_x s_y + 48s^3_x s_y + 1408s^2_y + 704s^3_x y + 78s^2_y + 192s^3_y \\
+ 48s_x s_y^2 + 9s_y^3 + 2c_x c_y^2(64 + 5s^2_x + 32s_y + 3s_y^2 + 8s_y(4 + s_y)) \\
+ 2c_x(64 - c_y^2 + 3s_y^2 + 32s_y + 5s_y^2 + 8s_y(4 + s_y)),
\]

\[
r_{11} = \frac{1}{256}(4 + s_x + s_y)(64 + c_x^2 + c_y^2 + 32s_x + 3s_x^2 + 32s_y + 8s_x s_y + 3s_y^2),
\]

\[
r_{22} = -\frac{1}{512}(c_x - s_x)(64 + c_x^2 - c_y^2 + 32s_x + 3s_x^2 + 32s_y + 8s_x s_y + 5s_y^2),
\]

\[
r_{33} = -\frac{1}{512}(c_x - s_x)(64 + c_x^2 - c_y^2 + 32s_x + 3s_x^2 + 32s_y + 8s_x s_y + 5s_y^2),
\]

\[
r_{44} = -\frac{1}{512}(c_y - s_y)(64 - c_y^2 + c_x^2 + 32s_x + 5s_x^2 + 32s_y + 8s_x s_y + 3s_y^2),
\]

\[
r_{55} = \frac{1}{512}(c_x + s_x)(64 + c_x^2 - c_y^2 + 32s_x + 3s_x^2 + 32s_y + 8s_x s_y + 5s_y^2),
\]

\[
r_{66} = -\frac{1}{128}(c_x + s_x)(c_y - s_y)(4 + s_x + s_y),
\]

\[
r_{77} = \frac{1}{128}(c_x + s_x)(c_y + s_y)(4 + s_x + s_y),
\]

\[
r_{88} = \frac{1}{512}(c_y + s_y)(64 - c_x^2 + c_y^2 + 32s_x + 5s_x^2 + 32s_y + 8s_x s_y + 3s_y^2),
\]

\[
r_{99} = -\frac{1}{128}(c_x - s_x)(c_y + s_y)(4 + s_x + s_y).
\]

To use this method, the grid points in the solution domain are divided into three types of points, denoted by the symbols □, △ and ● in alternate ordering as shown in Figure 3. Note that the evaluation of (11) require points of type ● only. Thus, we can construct the FMEG method by generating the iterations on this type of points only, followed by the evaluation of solutions directly once on points of type □ and △. The FMEG algorithm can then be summarized as follows:

1. Divide the grid points into three types □, △ and ● in alternate order as depicted in Figure 3.
2. Set the initial guess for the iterations.
(3) Evaluate the solutions at points • using Equation (11) iteratively at the time level \( k + 1 \).

(4) Step 5 is performed if the iteration converges. Otherwise, Step 3 is repeated until a convergence is attained.

(5) The steps below are performed directly once for points □ and △ in Figure 3 at the time level \( k + 1 \):

(a) For type □ points, the rotated \( h \) – spaced five-point approximation formula derived by rotating the \( x – y \) axis clockwise at 45° was used (Tan, Ali, & Lai, 2012). This approximation formula was applied to the right side of Equation (1), in combination with (3) being applied to the left side of (1), to obtain the following rotated C-N formula:

\[
\left( 1 + \frac{S_x + S_y}{2} \right) U^{k+1}_{ij} = \frac{S_x}{4} + \frac{C_x - C_y}{8} U_{i-1,j+1}^{k} + \left( \frac{S_x}{4} - \frac{C_x + C_y}{8} \right) U_{i+1,j}^{k} + \left( \frac{S_y}{4} + \frac{C_x - C_y}{8} \right) U_{i,j-1}^{k} + \left( \frac{S_y}{4} + \frac{C_x + C_y}{8} \right) U_{i,j+1}^{k} + \left( 1 - 2^{1-s} w_1 - \frac{S_x + S_y}{2} \right) U_{ij}^{k} + \left( \frac{S_x}{4} + \frac{C_x - C_y}{8} \right) U_{i-1,j+1}^{k} + \left( \frac{S_y}{4} + \frac{C_x + C_y}{8} \right) U_{i+1,j}^{k} + \left( \frac{S_y}{4} + \frac{C_x - C_y}{8} \right) U_{i,j-1}^{k} + \left( \frac{S_y}{4} + \frac{C_x + C_y}{8} \right) U_{i,j+1}^{k} + 2^{1-s} \sum_{s=1}^{k-1} \left[ w_{k-s} - w_{k-s+1} \right] U_{ij}^{k} + m_{ij}^{k+1/2}.
\]

(b) For type △ points, the typical \( h \) – spaced formula (5) is used. Note that formula (5) involves points of type △ only.

### 3. Stability analysis

A scheme is considered to be stable if the errors cease to increase with the passing of time, and gradually become inconsequential as the computation progresses. Even though the spacing is different, Equation (5) gives rise to both the FEG and FMEG methods. Therefore, the stability of both methods can be analyzed in similar ways. Here, we will show the stability of the FMEG scheme using eigenvalues of the generated matrices with mathematical induction.

Form (10) we obtain

\[
AU^k = BU
\]

\[
AU^{k+1} = BU^k - C_1 U^k + \sum_{s=1}^{k-1} C_{k-s} U^s + C_j U^0 + b \quad k > 0
\]

\[(12)\]

where

\[
A = \begin{pmatrix}
R_1 & R_2 & 0 & 0 \\
R_1 & R_2 & 0 & \vdots \\
0 & R_3 & R_1 & R_2 \\
0 & 0 & R_3 & R_1 \\
\end{pmatrix}, \quad B = \begin{pmatrix}
S_1 & S_2 & 0 & 0 \\
S_3 & S_1 & S_2 & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & S_3 & S_1 & S_2 \\
0 & 0 & S_3 & S_1 \\
\end{pmatrix}, \quad b = \begin{pmatrix}
W_1 \\
W_1 \\
\vdots \\
W_1 \\
\end{pmatrix}
\]

\[
R_1 = \begin{pmatrix}
G_1 & G_3 & G_5 \\
G_2 & G_1 & G_3 \\
\vdots & \vdots & \ddots \\
G_2 & G_1 & G_3 \\
\end{pmatrix}, \quad R_2 = \begin{pmatrix}
G_5 \\
G_5 \\
\vdots \\
G_5 \\
\end{pmatrix}, \quad R_3 = \begin{pmatrix}
G_6 & G_4 \\
G_4 & \ddots \\
\end{pmatrix}
\]
\[
S_1 = \begin{pmatrix}
H_1 & H_3 & H_3 \\
H_2 & H_1 & H_3 \\
& H_2 & H_1 \\
\end{pmatrix}, \quad
S_2 = \begin{pmatrix}
H_4 & H_5 \\
& H_5 \\
\end{pmatrix}, \quad
S_3 = \begin{pmatrix}
H_4 & H_4 \\
\end{pmatrix}
\]

\[
C_k = \begin{pmatrix}
M_k & M_k \\
& M_k \\
& & M_k \\
\end{pmatrix}, \quad
C_{k-s} = \begin{pmatrix}
M_{k-s} & M_{k-s} \\
& M_{k-s} \\
& & M_{k-s} \\
\end{pmatrix}, \quad s = 1, \ldots, k - 1
\]

\[
C_1 = \begin{pmatrix}
M_1 & M_1 \\
& M_1 \\
& & M_1 \\
\end{pmatrix}
\]

\[
W_1 = \begin{pmatrix}
L_1 \\
L_1 \\
\end{pmatrix}
\]

\[
G_1 = \begin{pmatrix}
D_1 & -a_{11} & 0 & -b_{11} \\
-a_{22} & D_1 & -b_{11} & 0 \\
0 & -b_{22} & D_1 & -a_{22} \\
-b_{22} & 0 & -a_{11} & D_1 \\
\end{pmatrix}
\]

\[
G_2 = \begin{pmatrix}
0 & 0 & 0 & -b_{22} \\
0 & 0 & -b_{22} & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

\[
G_3 = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & -b_{11} & 0 & 0 \\
-b_{11} & 0 & 0 & 0 \\
\end{pmatrix}
\]

\[
G_4 = \begin{pmatrix}
0 & -a_{22} & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & -a_{22} & 0 \\
\end{pmatrix}
\]

\[
G_5 = \begin{pmatrix}
0 & 0 & 0 & 0 \\
-a_{11} & 0 & 0 & 0 \\
0 & 0 & 0 & -a_{11} \\
0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

\[
H_1 = \begin{pmatrix}
Q & a_{11} & 0 & b_{11} \\
a_{22} & Q & b_{11} & 0 \\
0 & b_{22} & Q & a_{22} \\
b_{22} & 0 & a_{11} & Q \\
\end{pmatrix}
\]

\[
H_2 = \begin{pmatrix}
0 & 0 & 0 & b_{22} \\
0 & 0 & b_{22} & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
\end{pmatrix}
\]

\[
H_3 = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & b_{11} & 0 & 0 \\
b_{11} & 0 & 0 & 0 \\
\end{pmatrix}
\]

\[
H_4 = \begin{pmatrix}
0 & a_{22} & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & a_{22} & 0 \\
\end{pmatrix}
\]

\[
H_5 = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & a_{11} & 0 \\
0 & 0 & 0 & a_{11} \\
\end{pmatrix}
\]

\[
M_{k-s} = \frac{2^{1-s}}{t^s} \begin{pmatrix}
(w^* - w^*_{-s}) & 0 & 0 & 0 \\
0 & (w^* - w^*_{-s}) & 0 & 0 \\
0 & 0 & (w^* - w^*_{-s}) & 0 \\
0 & 0 & 0 & (w^* - w^*_{-s}) \\
\end{pmatrix}
\]

\[
M_k = \frac{2^{1-s}}{t^s} \begin{pmatrix}
w^* & 0 & 0 & 0 \\
0 & w^* & 0 & 0 \\
0 & 0 & w^* & 0 \\
0 & 0 & 0 & w^* \\
\end{pmatrix}
\]

\[
M_1 = \frac{2^{1-s}}{t^s} \begin{pmatrix}
f_{i,j} & f_{i+2,j} & f_{i+2,j+2} & f_{i,j+2} \\
\end{pmatrix}, \quad i,j = 2, 6, \ldots, n - 2.
\]

\[
L_1 = 2^{1-s} \Gamma[2 - \alpha] \begin{pmatrix}
f_{i,j} \\
f_{i+2,j} \\
f_{i+2,j+2} \\
f_{i,j+2} \\
\end{pmatrix}, \quad i,j = 2, 6, \ldots, n - 2.
\]
The following lemma is important to prove the stability.

**Lemma 1** [Karatay, Kale, & Bayramoglu, 2013] In (12), the coefficients \( w'_s, s = 1, 2, \ldots, k \) satisfy the following 1 - \( w'_{s-1} > w'_{s-1} \), s = 1, 2, ..., k, \( 2 - \sum_{s=1}^{k-1} [w'_{s-1} - w'_{s-1}] + w'_{s} = w'_{s}, s = 1, 2, \ldots, k. \)

For simplicity, we assume \( s_j = s_j = s = \frac{\gamma^2 - \gamma^2}{n^2}, \) \( c_j = c_j = C = \frac{\gamma^2 - \gamma^2}{h} \), \( D_1 = \frac{1}{2} - \frac{1}{2} s_j \) and \( Q = \frac{1}{2} - 2^{1-s} w_1 - \frac{1}{2} \).

**Theorem 1** If \( \left( \frac{1}{s_j} - \frac{1}{2} - \frac{s_j}{n^2} + \frac{\sqrt{c_s + d_s}}{4} \right) > 0 \) then the FMEG scheme (10) is stable.

**Proof** To prove the stability of (10), we suppose that \( U_{ij}^k, i, j = 1, 2, \ldots, n, k = 1, 2, \ldots, l \) are the approximate solutions to the exact solution \( U_{ij}^k \) of (1), the error \( \epsilon_{ij}^k = U_{ij}^k - U_{ij}^k \) be the error at time level \( k. \) From (12), the error satisfies

\[
AE^1 = BE
\]

\[
AE^{k+1} = BE^k - C_1E^k + \sum_{s=1}^{k-1} C_s U^k + C_kE^0 \quad k > 0
\]

where

\[
E^{k+1} = \begin{pmatrix}
E_1^{k+1} \\
E_2^{k+1} \\
\vdots \\
E_m^{k+1}
\end{pmatrix},
\]

\[
E_1^{k+1} = \begin{pmatrix}
E_{11}^{k+1} \\
E_{12}^{k+1} \\
\vdots \\
E_{1m}^{k+1}
\end{pmatrix},
\]

\[
E_2^{k+1} = \begin{pmatrix}
E_{21}^{k+1} \\
E_{22}^{k+1} \\
\vdots \\
E_{2m}^{k+1}
\end{pmatrix},
\]

\[
E_{ij}^{k+1} = \begin{pmatrix}
E_{i1}^{k+1} \\
E_{i2}^{k+1} \\
\vdots \\
E_{im}^{k+1}
\end{pmatrix},
\]

\[
E_{ij}^{k+1} = \begin{pmatrix}
E_{ij}^{k+1} \\
E_{ij}^{k+1} \\
\vdots \\
E_{ij}^{k+1}
\end{pmatrix},
\]

\[
i = 2, 6, \ldots, n - 2, j = 2, 6, \ldots, n - 2
\]

From the above equations, the following are obtained:

\[
A = G_4 + G_2 + G_1 + G_3 + G_5
\]

\[
B = H_4 + H_2 + H_1 + H_3 + H_5
\]

\[
C_1 = M_1
\]

\[
C_{k-5} = M_{k-5}
\]

\[
C_k = M_k
\]

It is worthy to note that, from (14), the eigenvalues of the matrices \( A, B, C_1, C_{k-5} \) and \( C_k \) are \( a_k, b_k, c_1, c_{k-5} \) and \( c_k \) respectively, where

| Table 1. Number of various types of mesh points for the point and explicit group methods |
|-----------------------------------------------|
| **Point types** | **Number of points** |
|-----------------|----------------------|
| **FSP** | **FE1** | **FMEG** |
| Iterative group points | \( m^2 \) | \( (m - 1)^2 \) | \( \frac{(m-1)^2}{4} \) |
| Iterative ungrouped points | \(-\) | \( 2m-1 \) | \(-\) |
| Total iterative points | \( m^2 \) | \( m^2 \) | \( \frac{(m-1)^2}{4} \) |
| Direct h-spaced rotated points | \(-\) | \(-\) | \( \frac{(m+1)^2}{4} \) |
| Direct h-spaced standard points | \(-\) | \(-\) | \( \frac{m^2+1}{4} \) |
| Total direct points | \(-\) | \(-\) | \( \frac{3m^2+2m-1}{4} \) |
\[ a_k = \left\{ \frac{1}{t^2} + \frac{3}{2} \cdot \frac{1}{t^2} + \frac{S}{2} \cdot \frac{1}{4} \left( \frac{4}{t^2} + 2S - \sqrt{-C^2 + S^2} \right), \frac{1}{4} \left( \frac{4}{t^2} + 2S + \sqrt{-C^2 + S^2} \right) \right\} \]

\[ b_k = \left\{ \frac{1}{t^2} - \frac{S}{2} \cdot \frac{1}{t^2} - \frac{1}{2} \cdot \frac{1}{4} \left( \frac{4}{t^2} - 2S - \sqrt{-C^2 + S^2} \right), \frac{1}{4} \left( \frac{4}{t^2} - 2S + \sqrt{-C^2 + S^2} \right) \right\} \]

\[ c_1 = \frac{2^{1-s} - w}{t^2, \ s = 1, \ldots, k} \]

\[ c_k = \frac{2^{1-s} - w}{t^2, \ s = 1, \ldots, k} \]

For \( k = 0 \)

\[ E^1 = A^{-1} B E^0 \]

\[ \|E^1\| \leq \rho(A^{-1} B) \|E^0\| = \left\| \frac{1}{t^2} - \frac{S}{2} + \frac{\sqrt{-C^2 + S^2}}{4} \right\| \cdot \|E^0\| \]

\[ \therefore \|E^1\| \leq \|E^0\| \]

Supposing that: \( \|E^s\| \leq \|E^0\|, \ s = 1, \ldots, k. \)

Need to prove this inequality holds for \( s = k + 1. \)

\[ E^{k+1} = A^{-1}(B - C_1)E^k + \sum_{s=1}^{k-1} A^{-1}C_{k-s}E^s + A^{-1}C_kE^0 \]

\[ \leq \rho(A^{-1}(B - C_1)) \|E^0\| + \sum_{s=1}^{k-1} \rho(A^{-1}C_{k-s}) \|E^0\| + \rho(A^{-1}C_k) \|E^0\| \]

Using Lemma 1, we get

\[ \|E^{k+1}\| \leq \left\| \frac{1}{t^2} - \frac{S}{2} - \frac{2^{1-s} - w}{t^2} + \frac{\sqrt{-C^2 + S^2}}{4} \right\| \cdot \|E^0\| + \frac{2^{1-s} - w}{t^2} \|E^0\| \]

\[ \therefore \|E^{k+1}\| \leq \|E^0\| \]

Therefore, under the conditions \( \left( \frac{1}{t^2} - \frac{S}{2} - \frac{2^{1-s} - w}{t^2} + \frac{\sqrt{-C^2 + S^2}}{4} \right) > 0 \) the FMEG iterative scheme (10) is stable.
Table 3. Comparison of the number of iterations, Execution times, average, and maximum error for different time step and mesh size at $\alpha = 0.75$ and $\alpha = 0.95$

| Example 1 | $\alpha = 0.75$ | | $\alpha = 0.95$ | |
|-----------|-----------------|-----------------|-----------------|-----------------|
| h         | $\Delta t$     | Method | Time  | Ite.  | Ave     | Max     | Time  | Ite.  | Ave     | Max     |
| 1/6       | 1/100           | FSP    | 9.18  | 11    | 6.48E-4 | 1.03E-3 | 6.16  | 8     | 6.12E-4 | 9.76E-4 |
|           |                 | FEG    | 7.25  | 8     | 6.45E-4 | 1.029E-3| 5.49  | 6     | 6.120E-4| 9.763E-4|
|           |                 | FMEG   | 1.13  | 2     | 2.06E-3 | 3.375E-3| 1.15  | 2     | 1.986E-3| 3.252E-3|
| 1/10      | 1/350           | FSP    | 350.37| 11    | 2.480E-4| 4.409E-4| 195.31| 6     | 2.581E-4| 4.664E-4|
|           |                 | FEG    | 367.80| 8     | 2.09E-4 | 3.807E-4| 172.19| 5     | 2.031E-4| 3.681E-4|
|           |                 | FMEG   | 54.55 | 4     | 6.89E-4 | 1.27E-3 | 54.30 | 4     | 6.709E-4| 1.240E-3|
| 1/14      | 1/850           | FSP    | 3924.31| 10   | 2.172E-4| 4.353E-4| 2032.86| 5     | 2.506E-4| 5.002E-4|
|           |                 | FEG    | 2941.31| 7    | 1.346E-4| 4.353E-4| 2024.52| 5     | 9.522E-5| 1.801E-4|
|           |                 | FMEG   | 700.02| 4     | 3.402E-4| 6.539E-4| 604.66| 3     | 3.317E-4| 6.372E-4|
| 1/18      | 1/1620          | FSP    | 2223.3| 9     | 2.923E-4| 6.114E-4| 1222.43| 5     | 1.599E-4| 3.242E-4|
|           |                 | FEG    | 17268.0| 6    | 1.613E-4| 3.212E-4| 10499.4| 4     | 7.361E-5| 1.471E-4|
|           |                 | FMEG   | 4330.99| 4    | 2.208E-4| 4.015E-4| 3786.52| 3     | 1.966E-4| 3.887E-4|

Figure 4. A comparison of the numerical solution of the example 1 at $\alpha = 0.75$, $y = 1/6$, $t = 1$ between, (A) FSP and FMEG (B) FEG and FMEG.

Figure 5. A comparison of the numerical solution of the example 2 at $\alpha = 0.75$, $y = 1/6$, $t = 1$ between, (A) FSP and FMEG (B) FEG and FMEG.
Figure 6. Experimental results for the mesh size against the Time at $\alpha = 0.75$, (A) Example 1, (B) Example 2.

Figure 7. Experimental results for the mesh size against the Ite. at $\alpha = 0.75$, (A) Example 1, (B) Example 2.

Table 4. Comparison of the number of iterations, Execution times, average and maximum error for different time step and mesh size at $\alpha = 0.75$ and $\alpha = 0.95$

| Example 2 | $\alpha = 0.75$ | $\alpha = 0.95$ |
|-----------|----------------|----------------|
| $h$       | $\Delta t$     | Method | Time | Ite. | Ave | Max | Time | Ite. | Ave | Max |
| 1/6       | 1/100          | FSP    | 7.81 | 9    | 4.320E-4 | 6.690E-3 | 5.2 | 6 | 4.326E-4 | 6.702E-4 |
| 1/10      | 1/350          | FEG    | 6.53 | 7    | 4.359E-4 | 6.759E-4 | 4.83 | 5 | 4.372E-4 | 6.781E-4 |
|           |                | FMEG   | 1.17 | 2    | 1.364E-3 | 2.276E-3 | 1.32 | 2 | 1.369E-3 | 2.283E-3 |
|           |                | FSP    | 294.69 | 9 | 1.173E-4 | 1.974E-4 | 169.2 | 5 | 1.126E-4 | 1.835E-4 |
| 1/10      | 1/300          | FEG    | 230.39 | 6 | 1.269E-4 | 2.192E-4 | 155.22 | 4 | 1.334E-4 | 2.238E-4 |
|           |                | FMEG   | 55.13 | 4 | 5.190E-4 | 9.475E-4 | 8.4 | 3 | 5.205E-4 | 9.504E-4 |
|           |                | FSP    | 3,267.46 | 8 | 1.577E-5 | 3.441E-5 | 1.772.2 | 4 | 1.321E-5 | 3.995E-5 |
| 1/14      | 1/850          | FEG    | 2,621.1 | 6 | 5.451E-5 | 9.851E-5 | 1,718.99 | 4 | 6.546E-5 | 1.202E-4 |
|           |                | FMEG   | 690.99 | 4 | 2.643E-4 | 4.999E-4 | 596.96 | 3 | 2.652E-4 | 5.017E-4 |
|           |                | FSP    | 17,647.9 | 7 | 5.607E-5 | 1.264E-4 | 10,034.0 | 4 | 1.780E-5 | 4.500E-5 |
| 1/18      | 1/1620         | FEG    | 13,574.3 | 5 | 4.239E-6 | 1.010E-5 | 9,708.83 | 3 | 1.494E-5 | 3.291E-5 |
|           |                | FMEG   | 4,351.54 | 4 | 1.584E-4 | 3.057E-4 | 3,764.74 | 3 | 1.591E-4 | 3.072E-4 |
Let us denote the truncation error at \(t_k\) into (12) produces:

\[
A^e_1 = R^\frac{1}{2} \\
A^e_1 + Be^k - C_1e^k + \sum_{i=1}^{k-1} C_{k-i}e^{k-i} + R^{k+\frac{1}{2}}.
\]

Using mathematical induction to prove the above theorem, set \(\|C_0^{-1}\| = 1\) for \(k = 0\)

\[
A^e_1 = R^\frac{1}{2} \\
\|e^k\| \leq \rho(A^{-1})\|R\| \leq \|C_0^{-1}\| (\tau^{2-s} + (\Delta x)^2 + (\Delta y)^2).
\]

Assume that \(\|e^k\| \leq \|C_{k-1}^{-1}\|\|R^{k+\frac{1}{2}}\|\)

Need to prove it hold for \(s = k + 1\)

Table 5. Total computing operations involved for the point and grouping methods

| \(h\) | \(\Delta t\) | Method | \(\alpha = 0.75\) | \(\alpha = 0.95\) | \(\alpha = 0.75\) | \(\alpha = 0.95\) |
|---|---|---|---|---|---|---|
| 1/6 | 1/100 | FEG | 196,625 | 143,000 | 160,875 | 107,250 |
|   |   | FMEG | 23,002 | 23,002 | 23,002 | 23,002 |
|   |   | FSP | 19,698,000 | 13,374,800 | 18,743,200 | 11,245,900 |
| 1/10 | 1/350 | FEG | 1,000,090 | 1,200,100 | 800,068 |
|   |   | FMEG | 341,462 | 341,462 | 301,934 |
|   |   | FSP | 1,766,910 | 1,551,970 | 1,766,910 | 1,551,970 |
| 1/14 | 1/850 | FEG | 7,062,140 | 5,044,390 | 6,053,260 | 4,035,510 |
|   |   | FMEG | 1,766,910 | 1,551,970 | 1,766,910 | 1,551,970 |
|   |   | FSP | 7,062,140 | 5,044,390 | 6,053,260 | 4,035,510 |
| 1/18 | 1/1620 | FEG | 19,698,000 | 13,374,800 | 18,743,200 | 11,245,900 |
|   |   | FMEG | 5,828,950 | 5,196,200 | 6,672,190 | 5,836,220 |

4. Convergence analysis

**Theorem 2**\ The finite difference FMEG scheme (10) is convergent and the following estimate hold

\[
\|e^{k+1}\| \leq C^e_0 (\tau^{2-s} + (\Delta x)^2 + (\Delta y)^2) \quad \text{if} \quad \frac{1}{\tau} - \frac{s}{2} - \frac{1}{2} - \frac{C^e_0}{4} > 0.
\]

Let us denote the truncation error at \((x_i, y_j, t_{k+1})\) by \(R^{k+1}\). From (8), we have,\n
\[
\|R^{k+1/2}\| \leq C(\tau^{2-s} + (\Delta x)^2 + (\Delta y)^2)
\]

Define \(\eta^{k+1} = U(x_i, y_j, t_{k+1}) - u^{k+1}_{i,j} = 1, 2, \ldots, n, \quad k = 1, 2, \ldots, l\) and \(e^{k+1} = (e^{k+1}_1, e^{k+1}_2, \ldots, e^{k+1}_l)^T\),

using \(e^0 = 0\), where \(e^{k+1}_1 = \begin{bmatrix} n_1 & \cdots & n_k \\ n_{k+1} & \cdots & n_{k+1} \\ \vdots & \ddots & \vdots \\ n_{m-1} & \cdots & n_{m-1} \end{bmatrix}, \quad e^{k+1}_2 = \begin{bmatrix} n_1^{k+1} & \cdots & n_1^{k+1} \\ \vdots & \ddots & \vdots \\ n_{m-1}^{k+1} & \cdots & n_{m-1}^{k+1} \end{bmatrix}, \quad (i, j) = 2, 6, \ldots, n - 2\). Substitution into (12) produces:

\[
\begin{align*}
Ae^1 &= R^\frac{1}{2} \\
Ae^{k+1} &= Be^k - C_1e^k + \sum_{i=1}^{k-1} C_{k-i}e^{k-i} + R^{k+\frac{1}{2}}.
\end{align*}
\]
\[ Ae^{k+1} = (B - C_1) e^k + \sum_{s=1}^{k-1} C_{k-s} e^{s+1} + R^{k+\frac{1}{2}} \]

\[ \| e^{k+1} \| \leq \rho(A^{-1}(B - C_1)) \| e^k \| + \sum_{s=1}^{k-1} \rho(A^{-1} C_{k-s}) \| e^{s+1} \| + \rho(A^{-1}) \| R^{k+\frac{1}{2}} \| \]

\[ \| e^{k+1} \| \leq \left( \frac{1}{\rho} \right) \left( \frac{1}{\rho} \right) W_1 \| C_{k-1} \| R^{k+\frac{1}{2}} \| \]

\[ = \left( \frac{21-a}{2t} \right) \left( \frac{(2-a) \sigma^2 + (\sigma x)^2 + (\sigma y)^2}{2(1-a)} \right) \]

Since \( \lim_{k \to \infty} \left( \frac{k}{(k+\frac{1}{2})} \right) = \frac{1}{1-a} \)

5. Computational complexity

This section presents an analysis of the computational complexity with regard to the three techniques described for solving (1), which is the number of arithmetic operations per iteration. For simplicity, we assume \( n = m + 1 \), where \( n \) is an even mesh size, then the ungrouped points will be found close to the right/upper boundaries. Internal mesh points have two main types, namely, the iteration points that participate in the iteration process and the direct points that are directly calculated once using the rotated and standard difference formulas following the attainment of the iteration convergence. Table 1 lists the number of internal mesh points for the three earlier methods, whereas Table 2 provides a summary of the number of arithmetic operations that are needed for each iteration and the direct solution following the convergence, not only for the explicit group methods, but also for the fractional standard point (FSP) method.

6. Numerical experiments and discussion of results

Several numerical examples are presented in this section to prove the effectiveness of the fractional explicit group methods in solving the 2-D TFADDE (1) with a Dirichlet boundary condition. A computer with Windows 7 Professional and Mathematica software having a Core i7 GHz and 4 GB of RAM was used for conducting the experiments.

Example 1 The time fractional initial boundary value problem below was considered (Zhuang, Gu, Liu, Turner, & Yarlagadda, 2011)

\[ \frac{\partial^a u}{\partial t^a} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - \frac{\partial u}{\partial x} - \frac{\partial u}{\partial y} + 0.1(3 + 4t^2 \sin y) \]

where \( \Omega = \{ (x,y) | 0 \leq x \leq 1, 0 \leq y \leq 1 \} \) is the solution domain with the exact solution being \( t^{2-a} e^{2y} \).

Example 2 The following time fractional advection-diffusion equation was also considered (Mohebbi & Abbaszadeh, 2013)

\[ \frac{\partial u}{\partial t^a} = \frac{\partial^a u}{\partial x^a} + \frac{\partial^2 u}{\partial y^2} - \frac{\partial u}{\partial x} - \frac{\partial u}{\partial y} + \frac{t^{2-a}(\sin x \sin y)}{(2-a)} + t(\cos x + \sin x + \cos y + \sin y) \]

with the initial and boundary conditions are given as \( u(x,y,0) = \Omega(0,y,t) = t \sin y, u(1,y,t) = t(\sin 1 + \sin y), u(x,0,t) = t \sin x, u(x,1,t) = t(\sin x + \sin 1), 0 < t < 1, 0 < x, < 1 \).

Different mesh sizes of 6, 10, 14, and 18 and various time steps, which satisfy the stability conditions, with a fixed relaxation factor (Gauss-Seidel relaxation scheme) of 1.0 were used to run the experiments. During all the experiments, the norm for the convergence criteria was \( e^{-1} \), while error tolerance was set at \( e = 10^{-5} \). Numerical results were obtained using of the methods described in...
Section 2 for different values of $\alpha$. The number of iterations, the execution time, and the error analysis are presented in Tables 3 and 4 and Figures 4–7 for the fractional point method and the fractional group methods. Figure 6 shows the execution times for various mesh sizes for both Examples 1 & 2. It’s clear that when fractional group methods are used, the results are just as accurate as the FSP method. From the results obtained, it is observed that the execution timings were reduced by as much as 20 and 80 % of the FSP method when the FEG and FMEG methods were used respectively. In contrast to the other tested methods, the fractional group methods, specifically the FMEG method, took at least times to compute the solutions. As shown in Tables 3 & 4 and Figure 6 the time taken for FMEG to be executed was only approximately 15.6–32.5% of the FEG method, and was 12.3–23.9% of the FSP method. To gauge the computational complexity, obtaining an approximation of the amount of calculations involved for each method is necessary. The estimation for the amount of computational work involved was determined by arithmetic operations that were performed for a single iteration (as outlined in Section 5). With the help of Tables 1 & 2, and based on the assumption that an approximately equal amount of time was needed for adding, subtracting, multiplying, dividing and assigning, a summary of the total number of operations involved in the iterative methods is presented in Table 5. This shows that the FMEG requires the least number of operations, thus confirming the theoretical complexity analysis.

7. Conclusion

This paper presented the development and formulation of new fractional explicit group iterative methods for solving the 2D-TFADE. The C-N difference schemes with $h$ spacing and $2h$ spacing gave rise to the FEG and FMEG methods, respectively. The stability and convergence of the proposed methods were analyzed using the matrix form with mathematical induction. Through the numerical experiments, the FMEG method stood out amongst all the other tested methods when it comes to its execution time and number of iterations, as it requires the least number of operation counts. It is noted that in terms of accuracy, the FMEG method is just as good as the FSP method and the FEG method. This work confirms the suitability and feasibility of the grouping strategies in solving the 2D time fractional advection-diffusion equation. The implementation of similar group methods on solving other fractional differential equations will be reported soon.

Funding

This work was supported by the Research University Individual (RUI) under (grant number 1001/PMTHS/8011016).

Author details

Alla Tareq Balasim1,2
E-mail: alkhazrejy@yahoo.com
ORCID ID: http://orcid.org/0000-0002-6533-1703
Norhashidah Hj. Mohd. Ali1
E-mail: shidah.ali@usm.my

1 School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia.
2 Department of Mathematics, College of Basic Education, University of Al-mustansiria, Baghdad, Iraq.

Citation information

Cite this article as: New group iterative schemes in the numerical solution of the two-dimensional time fractional advection-diffusion equation, Alla Tareq Balasim & Norhashidah Hj. Mohd. Ali, Cogent Mathematics (2017), 4: 1412241.

References

Abdelkawy, M., Zaky, M., Bhrawy, A., & Baleanu, D. (2015). Numerical simulation of time variable fractional order mobile-immobile advection-dispersion model. Romanian Reports in Physics, 67(3), 1–19.

Agrawal, O. P. (2008). A general finite element formulation for fractional variational problems. Journal of Mathematical Analysis and Applications, 337(1), 1–12.

Ali, N. H. M., & Kew, L. M. (2012). New explicit group iterative methods in the solution of two dimensional hyperbolic equations. Journal of Computational Physics, 231(20), 6953–6968.

Ali, U., Abdullah, F. A., & Mohyud-Din, S. T. (2017). Modified implicit fractional difference scheme for 2D modified anomalous fractional sub-diffusion equation. Advances in Difference Equations, 2017(1), 185.

Baeumer, B., Benson, D. A., Meerschaert, M. M., & Wheatcraft, S. W. (2001). Subordinated advection-dispersion equation for contaminant transport. Water Resources Research, 37(6), 1543–1550.

Bolásim, A. T., & Ali, N. H. M. (2015). A rotated Crank–Nicolson iterative method for the solution of two-dimensional time-fractional diffusion equation. Indian Journal of Science and Technology, 8(12).

Baleanu, D., Agheli, B., & Al Quraishi, M. M. (2016). Fractional advection differential equation within caputo and caputo-fabrizio derivatives. Advances in Mechanical Engineering, 8(12), 1–8.

Benson, D. A., Wheatcraft, S. W., & Meerschaert, M. M. (2000). Application of a fractional advection-dispersion equation. Water Resources Research, 36(6), 1403–1412.

Bhrawy, A., & Baleanu, D. (2013). A spectral legendre-gauss-lobatto collocation method for a space-fractional advection diffusion equations with variable coefficients. Reports on Mathematical Physics, 72(2), 219–233.

Chen, C.-M., Liu, F., Anh, V., & Turner, I. (2011). Numerical simulation for the variable-order galilei invariant advection diffusion equation with a nonlinear source term. Applied Mathematics and Computation, 217(12), 5729–5742.
Chen, C.-M., Liu, F., & Burrage, K. (2008). Finite difference methods and a fourier analysis for the fractional reaction-subdiffusion equation. Applied Mathematics and Computation, 198(2), 754–769.

Chen, M., Deng, W., & Wu, Y. (2013). Superlinearly convergent algorithms for the two-dimensional space-time coputo-riesz fractional diffusion equation. Applied Numerical Mathematics, 70, 22–41.

Chen, S., & Liu, F. (2008). Adi-euler and extrapolation methods for the two-dimensional fractional advection-dispersion equation. Journal of Applied Mathematics and Computing, 26(1–2), 295–311.

Cushman, J. H., & Ginn, T. R. (2000). Fractional advection-dispersion equation: A classical mass balance with convolution-fickian flux. Water Resources Research, 36(12), 3763–3766.

Evans, D., & Yousif, W. (1986). Explicit group iterative methods for solving elliptic partial differential equations in 3-space dimensions. International Journal of Computer Mathematics, 18(3–4), 323–340.

Gorenflo, R., Mainardi, F., Moretti, D., Pagnini, G., & Paradisi, P. (2002). Discrete random walk models for space-time fractional diffusion. Chemical Physics, 284(1), 521–541.

Henry, B. I., & Wearne, S. L. (2010). Fractional reaction-diffusion. Physica A: Statistical Mechanics and its Applications, 376(3), 448–455.

Kararay, I., Kole, N., & Bayramoglu, S. R. (2013). A new difference scheme for time fractional heat equations based on the crank-nicholson method. Fractional Calculus and Applied Analysis, 16(4), 892–910.

Kew, L. M., & Ali, N. H. M. (2015). New explicit group iterative methods in the solution of three dimensional hyperbolic telegraph equations. Journal of Computational Physics, 294, 382–404.

Leonenko, N. N., Meerschaert, M. M., & Sikorskii, A. (2013). Fractional Pearson diffusions. Journal of Mathematical Analysis and Applications, 403(2), 532–546.

Li, C., Qian, D., & Chen, Y. (2011). On Riemann–Liouville and Caputo derivatives. Discrete Dynamics in Nature and Society, 2011.

Li, C., Zeng, F., & Liu, F. (2012). Spectral approximations to the fractional integral and derivative. Fractional Calculus and Applied Analysis, 15(3), 383–406.

Liu, F., Zhang, P., Anh, V., Turner, I., & Burrage, K. (2007). Stability and convergence of the difference methods for the space-time fractional advection-diffusion equation. Applied Mathematics and Computation, 191(1), 12–20.

Mehdinejadian, B., Nasri, A. A., Jafari, H., Ghanbarzadeh, A., & Boroujeni, D. (2013). A mathematical model for simulation of a water table profile between two parallel subsurface drains using fractional derivatives. Computers & Mathematics with Applications, 66(5), 785–794.

Metzler, R., Borkai, E., & Klafter, J. (1999). Anomalous diffusion and relaxation close to thermal equilibrium: A fractional Fokker-Planck equation approach. Physical Review Letters, 82(18), 3563.

Metzler, R., & Klafter, J. (2000). Boundry value problems for fractional diffusion equations. Physico A: Statistical Mechanics and its Applications, 278(1), 107–125.

Miller, K. S., & Ross, B. (1993). An introduction to the fractional calculus and fractional differential equations. New York, NY: Wiley.

Mohd. Ali, N. H. M., & Saeedi, A. (2008). Performance analysis of explicit group parallel algorithms for distributed memory multicomputer. Parallel Computing, 34(6), 427–440.

Oldham, K. B., & Spanier, J. (1974). The fractional calculus. New York, NY: Academic Press.

Othman, M., & Abdulhadi, A. (2000). An efficient four points modified explicit group poison solver. International Journal of Computer Mathematics, 76(2), 203–217.

Podlubny, I. (1998). Fractional differential equations: An introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications (Vol. 198). New York, NY: Academic Press.

Samko, S. G., Kilbas, A. A., & Marichev, O. I. (1993). Fractional integrals and derivatives. London: Taylor & Francis.

Seki, K., Wójcik, M., & Tachiya, M. (2003). Fractional reaction-diffusion equation. The Journal of Chemical Physics, 119(4), 2165–2170.

Shen, S., Liu, F., & Anh, V. (2011). Numerical approximations and solution techniques for the space-time riesz-caputo fractional advection-diffusion equation. Numerical Algorithms, 56(3), 383–403.

Sousa, E. & Li, C. (2015). A weighted finite difference method for the fractional diffusion equation based on the Riemann–Liouville derivative. Applied Numerical Mathematics, 90, 22–37.

Su, L., Wang, W., & Wang, H. (2011). A characteristic difference method for the transient fractional convection-diffusion equations. Applied Numerical Mathematics, 61(8), 946–960.

Tan, K. B., Ali, N. H. M., & Loi, C.-H. (2012). Parallel block interface domain decomposition methods for the 2d convection-diffusion equation. International Journal of Computer Mathematics, 89(12), 1704–1723.

Uddin, M., & Haq, S. (2011). Rbf approximation method for time fractional partial differential equations. Communications in Nonlinear Science and Numerical Simulation, 16(11), 4208–4214.

Wyss, W. (1986). The fractional diffusion equation. Journal of Mathematical Physics, 27(11), 2782–2785.

Yousif, W., & Evans, D. J. (1995). Explicit de-coupled group iterative methods and their parallel implementations. Parallel Algorithms and Applications, 7(1–2), 53–71.

Zhang, X., Huang, P., Feng, X., & Wei, L. (2013). Finite element method for two-dimensional time-fractional tricomi-type equations. Numerical Methods for Partial Differential Equations, 29(4), 1081–1096.

Zheng, Y., Li, C., & Zhao, Z. (2010). A note on the finite element method for the space-fractional advection diffusion equation. Computers & Mathematics with Applications, 59(5), 1718–1726.

Zhuang, P., Guo, Y., Liu, F., Turner, J., & Yarlagadda, P. (2011). Time dependent fractional advection-diffusion equations by an implicit mls meshless method. International Journal for Numerical Methods in Engineering, 88(13), 1346–1362.
