Mathematics

http://www.aimspress.com/journal/Math

AIMS Mathematics, 7(2): 2370–2392.
DOI:10.3934/math.2022134
Received: 31 July 2021
Accepted: 04 November 2021
Published: 11 November 2021

Research article

An efficient modified hybrid explicit group iterative method for the time-fractional diffusion equation in two space dimensions

Fouad Mohammad Salama[1,*], **Nur Nadiah Abd Hamid**[1,*], **Norhashidah Hj. Mohd Ali**[1] **and Umair Ali**[2]

[1] School of Mathematical Sciences, Universiti Sains Malaysia, Penang 11800, Malaysia

[2] Department of Applied Mathematics and Statistics, Institute of Space Technology, P. O. Box 2750, Islamabad 44000, Pakistan

* **Correspondence:** Email: fuadmohd321@gmail.com, nurnadiah@usm.my.

Abstract: In this paper, a new modified hybrid explicit group (MHEG) iterative method is presented for the efficient and accurate numerical solution of a time-fractional diffusion equation in two space dimensions. The time fractional derivative is defined in the Caputo sense. In the proposed method, a Laplace transformation is used in the temporal domain, and, for the spatial discretization, a new finite difference scheme based on grouping strategy is considered. The unique solvability, unconditional stability and convergence are thoroughly proved by the matrix analysis method. Comparison of numerical results with analytical and other approximate solutions indicates the viability and efficiency of the proposed algorithm.

Keywords: Caputo fractional derivative; fractional diffusion equation; Laplace transform; finite difference scheme; grouping strategy; stability and convergence
Mathematics Subject Classification: 35XX, 65N12

1. Introduction

In recent years, fractional calculus has gained a great deal of attention from the research community due to its widespread applications in physics, chemistry, fluid mechanics, viscoelasticity, finance, control systems and other areas of science and engineering. The phenomena in the aforesaid fields can be modelled very successfully by means of equations containing fractional derivatives and fractional integrals; and therefore, the investigation of fractional differential equations has become a hot topic for many researchers. Fractional kinetic equations including Fokker-Planck equation, fractional diffusion equation, fractional cable equation and fractional advection-diffusion equation are proved to be powerful instruments for modelling transport dynamics in complex systems such as electrodiffusion of ions in spiny dendrites, transport of proteins molecules, movement of a solution in an aquifer, pollution of underground water, movement under the effect of optical tweezers and more [1–5]. To get more insight into the applications of fractional calculus, the interested reader can refer to the classical books in [6, 7], besides the recent books in [8, 9].

In this work, we study the following two-dimensional time-fractional diffusion equation with a non-homogeneous source term:

$$\begin{cases} {}_0^C D_t^\alpha u(x,y,t) = a_x \dfrac{\partial^2 u(x,y,t)}{\partial x^2} + a_y \dfrac{\partial^2 u(x,y,t)}{\partial y^2} + f(x,y,t),\ (x,y) \in \Omega,\ 0 < t \leq T, & (1.1) \\ u(x,y,t) = p(x,y,t), \quad (x,y) \in \partial\Omega,\ 0 < t \leq T, & (1.2) \\ u(x,y,0) = g(x,y), \quad (x,y) \in \Omega \cup \partial\Omega, & (1.3) \end{cases}$$

where $f(x,y,t)$, $p(x,y,t)$ and $g(x,y)$ are known functions, $\Omega = (0,L) \times (0,L)$ is a bounded domain in $\mathbb{R}^2$, $\partial\Omega$ is the boundary, $a_x > 0$ and $a_y > 0$ are the diffusion coefficients, and α $(0 < \alpha \leq 1)$ is the anomalous diffusion exponent. The term ${}_0^C D_t^\alpha u(x,y,t)$ stands for the Caputo fractional derivative of the function $u(x,y,t)$ that is defined as,

$${}_0^C D_t^\alpha u(x,y,t) = \begin{cases} \frac{1}{\Gamma(1-\alpha)} \int_0^t (t-\xi)^{-\alpha} \frac{\partial u(x,y,\xi)}{\partial \xi} d\xi, \ 0 < \alpha < 1, \\ \frac{\partial u(x,y,t)}{\partial t}, \ \alpha = 1. \end{cases}$$

From the above definition, it can be observed that the fractional diffusion problems (1.1)–(1.3) corresponds to the classical diffusion model as $\alpha = 1$. Fractional diffusion equation is one of the most fundamental equations in the literature, and its capability to model many problems in science and engineering is a well established fact. Generally speaking, it has been widely used for describing random walks and modelling phenomena that are governed by anomalous diffusion in various fields such as porous systems, nuclear magnetic resonance and transport in fractal geometries [10–12]. Since most fractional differential equations are difficult to handle analytically, many researchers have resorted to numerical techniques, especially the finite difference method for solving fractional diffusion problems [4, 10, 11, 13–25]. The complexity of fractional differential equations stems from the presence of non-local fractional derivatives that have the property of global dependence on time or space. This forms the principal obstacle to the development of efficient simulation algorithms in terms of CPU time and memory usage, especially for multi-dimensional problems [26–28]. To surmount this issue, techniques including fast Poisson solver [14], parallel implementation [15, 29], preconditioning [15] and multigrid method [30] were suggested in the literature. Therefore, the question of how we can remove the fractional derivative from our computations to reduce the complexity and establish efficient solution algorithms is considered a big question arising in the numerical simulations of fractional differential equations. In this regard, Ren et al. [31] have introduced a relatively new approach for solving Caputo-type fractional differential equations. In this approach, a Laplace transform technique is proposed to approximate the fractional differential equation by its corresponding integer-order differential equation, which can be solved with less effort by using a suitable numerical or analytical method (see [32, 33]). Recently in [34], a new hybrid standard point (HSP) iterative method based on a combination of the Laplace transform technique and implicit difference scheme has been proposed to solve the time-fractional diffusion problems (1.1)–(1.3). The authors showed that their method produces accurate numerical solutions with an optimal computational complexity of $O(N)$ and optimal memory requirement of $O(M_s)$, where N and M_s are the total number of time levels and spatial unknowns, respectively.

It is well known that the explicit group iterative methods [35–39] based on finite difference approximations are proposed for the numerical solutions of integer-order differential equations. From one side, explicit group methods are unconditionally stable as the conventional implicit schemes. On the other side, they reduce the number of spatial unknowns taken in the iterative process and thus lead to computationally efficient algorithms. Although the grouping methods have been successfully employed to abroad spectrum of classical differential models, very little work has been reported to deal with fractional differential equations using grouping techniques (see [40–43]). In addition, the handling of nonlinear and variable order fractional differential equations by explicit group methods, besides their parallel implementation, is another topic that is still at its infancy. Therefore, this subject needs a major development.

The main purpose of this paper is to combine the Laplace transform technique with an explicit group scheme to solve the two-dimensional time-fractional diffusion Eqs (1.1)–(1.3). The unconditional stability, convergence and solvability of the resulting method, namely the modified hybrid explicit group (MHEG) method are rigorously proved. To illustrate the efficiency and feasibility of the proposed method, the fast HSP iterative method based on the recent work in [34] is also presented.

The rest of this paper is organized as follows. In Section 2, we briefly review the existing HSP iterative method [34] for solving the problems (1.1)–(1.3). In Section 3, the MHEG iterative method is proposed, and its unconditional stability, convergence and solvability are rigorously proved in Section 4. In Section 5, numerical simulations are performed to indicate the efficiency of the proposed method and support our theoretical analysis. Finally, this work ends with a brief summary in Section 6.

2. Overview of the existing HSP iterative method

Since the Caputo fractional derivative in (1.1) is non-local and has the character of history dependence, conventional difference schemes (such as an implicit or explicit scheme with a particular discretiztion formula for the fractional derivative) for the problems (1.1)–(1.3) result in costly simulations in terms of CPU time and memory consumption. In order to surmount the computational challenge, and utilizing the Laplace transform technique, the Caputo's time fractional derivative is approximated as follows [34]:

$$ {}_0^C D_t^{\alpha} u(x,y,t) \approx \alpha \frac{\partial u(x,y,t)}{\partial t} + (1-\alpha)[u(x,y,t) - u(x,y,0)]. \tag{2.1}$$

By substituting (2.1) into (1.1), the original time-fractional diffusion Eq (1.1) is reduced to its corresponding integer-order partial differential equation (PDE) with the following initial and boundary conditions [34]:

$$\begin{cases} \dfrac{\partial u(x,y,t)}{\partial t} = A_x \dfrac{\partial^2 u(x,y,t)}{\partial x^2} + A_y \dfrac{\partial^2 u(x,y,t)}{\partial y^2} - (r-1)u(x,y,t) + (r-1)g(x,y) \\ +rf(x,y,t),\ (x,y) \in \Omega,\ 0 < t \le T, & (2.2) \\ u(x,y,t) = p(x,y,t), \quad (x,y) \in \partial\Omega,\ 0 < t \le T, & (2.3) \\ u(x,y,0) = g(x,y), \quad (x,y) \in \Omega \cup \partial\Omega, & (2.4) \end{cases}$$

where $A_x = a_x/\alpha$, $A_y = a_y/\alpha$ and $r = 1/\alpha$ are positive constants. Next, we let

$$x_i = ih,\ i = 0,1,\dots,M, \quad y_j = jh,\ j = 0,1,\dots,M, \quad t_k = k\tau,\ k = 0,1,\dots,N,$$

where $h = L/M$ and $\tau = T/N$ are the uniform space and time step sizes, respectively. We also introduce the grid functions given by

$$u_{i,j}^k = u(x_i, y_j, t_k), \quad u_{i,j}^0 = g(x_i, y_j), \quad f_{i,j}^k = f(x_i, y_j, t_k), \quad 0 \le i,j \le M,\ 0 \le k \le N.$$

Here, the notations $\delta_t U_{i,j}^{k+1}$, $\delta_x^2 U_{i,j}^{k+1}$ and $\delta_y^2 U_{i,j}^{k+1}$ are defined as follows:

$$\begin{aligned} \delta_t U_{i,j}^{k+1} &= \frac{1}{\tau}(U_{i,j}^{k+1} - U_{i,j}^k) + O(\tau), \\ \delta_x^2 U_{i,j}^{k+1} &= \frac{1}{h^2}(U_{i+1,j}^{k+1} - 2U_{i,j}^{k+1} + U_{i-1,j}^{k+1}) + O(h^2), \qquad (2.5) \\ \delta_y^2 U_{i,j}^{k+1} &= \frac{1}{h^2}(U_{i,j+1}^{k+1} - 2U_{i,j}^{k+1} + U_{i,j-1}^{k+1}) + O(h^2). \end{aligned}$$

Now, applying the difference operators in (2.5) to the approximating PDEs (2.2)–(2.4) and disregarding the truncation errors, we obtain the implicit difference scheme in the following form:

$$\begin{cases} u_{i,j}^{k+1} = \dfrac{1}{(1+(r-1)\tau + 2q_1 + 2q_2)}\Big[q_1(u_{i+1,j}^{k+1} + u_{i-1,j}^{k+1}) + q_2(u_{i,j+1}^{k+1} + u_{i,j-1}^{k+1}), \\ +u_{i,j}^k + (r-1)\tau u_{i,j}^0 + r\tau f_{i,j}^{k+1}\Big],\ 1 \le i,j \le M-1,\ 0 \le k \le N-1, & (2.6) \\ u_{i,j}^k|_{\partial\Omega} = p(x_i, y_j, t_k), \quad 0 \le k \le N, & (2.7) \\ u_{i,j}^0 = g(x_i, y_j), \quad 0 \le i,j \le M, & (2.8) \end{cases}$$

where $q_1 = A_x\tau/h^2$, $q_2 = A_y\tau/h^2$ and $u_{i,j}^k$ is the numerical approximation of the function $U_{i,j}^k$ after neglecting the truncation errors.

Considering the fact that the spatial and temporal discretizations usually lead to large and sparse systems of linear equations to be solved, iterative solvers, such as Gauss-Seidel method, are more practical and economic than the direct solvers. Here, and at each time level, the HSP method functions through the iterative evaluation of solutions at all mesh points ♦ shown in Figure 1 by using Eq (2.7) until certain convergence is achieved. Once the converged solution values $u^{k+1} = (u_{11}^{k+1}, u_{1,2}^{k+1}, \dots, u_{1,M-1}^{k+1}, \dots, u_{M-1,1}^{k+1}, u_{M-1,2}^{k+1}, \dots, u_{M-1,M-1}^{k+1})^T$ are attained, they are utilized as an initial guess for the next time level $u^{k+2} = (u_{11}^{k+2}, u_{1,2}^{k+2}, \dots, u_{1,M-1}^{k+2}, \dots, u_{M-1,1}^{k+2}, u_{M-1,2}^{k+2}, \dots, u_{M-1,M-1}^{k+2})^T$. Such process is repeated until it reaches the targeted time level. The HSP iterative method has been shown to reduce the memory requirement, computational complexity as well as the CPU time greatly when compared to the fractional standard point (FSP) iterative methods derived based on the conventional implicit difference schemes. For extra details, please refer to [34]. Seeking for more efficient solution algorithm, we propose our method in the next section.

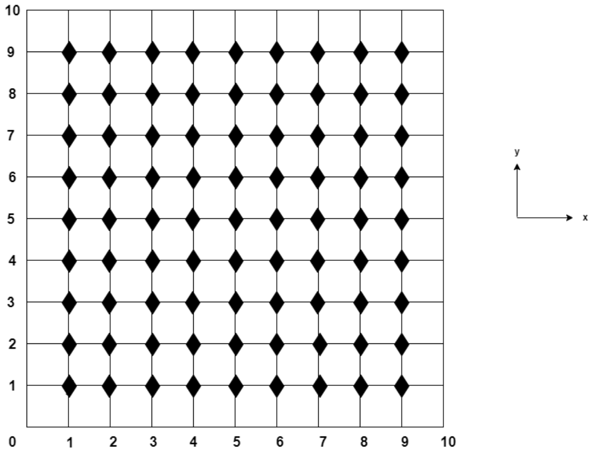


Figure 1. Distribution of solution mesh points for the HSP method when $M = 10$.

3. Formulation of the MHEG iterative method

Consider the standard mesh of points with the spatial step size $2h = 2L/M$ illustrated in Figure 2. We begin this section by discretizing the derivatives of the space variables in (2.2) around these $2h$ spaced points. Recalling the δ_t definition, the spatial difference operators δ_x^2 and δ_y^2 are defined as follows:

$$\begin{aligned}
\delta_t U_{i,j}^{k+1} &= \frac{1}{\tau}(U_{i,j}^{k+1} - U_{i,j}^{k}) + O(\tau),\\
\delta_x^2 U_{i,j}^{k+1} &= \frac{1}{h^2}(U_{i+2,j}^{k+1} - 2U_{i,j}^{k+1} + U_{i-2,j}^{k+1}) + O(h^2),\\
\delta_y^2 U_{i,j}^{k+1} &= \frac{1}{h^2}(U_{i,j+2}^{k+1} - 2U_{i,j}^{k+1} + U_{i,j-2}^{k+1}) + O(h^2).
\end{aligned} \tag{3.1}$$

Applying the difference operators in (3.1) to the approximating PDE (2.2), we obtain

$$\begin{aligned}
\frac{u_{i,j}^{k+1} - u_{i,j}^{k}}{\tau} =& A_x\left(\frac{u_{i+2,j}^{k+1} - 2u_{i,j}^{k+1} + u_{i-2,j}^{k+1}}{4h^2}\right) + A_y\left(\frac{u_{i,j+2}^{k+1} - 2u_{i,j}^{k+1} + u_{i,j-2}^{k+1}}{4h^2}\right)\\
& - (r-1)u_{i,j}^{k+1} + (r-1)u_{i,j}^{0} + rf_{i,j}^{k+1} + O(\tau + h^2).
\end{aligned} \tag{3.2}$$

figure.png

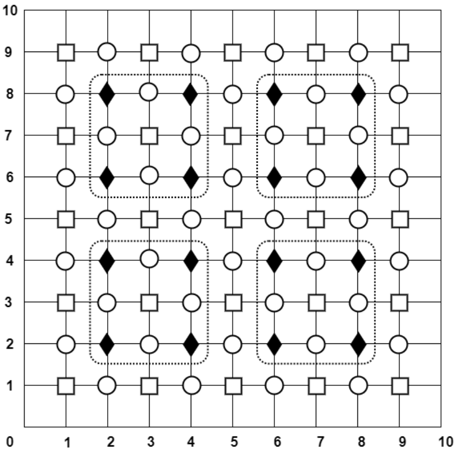


Figure 2. Distribution of solution mesh points for the MHEG method when $M = 10$.

After simplification and imposing the initial and boundary conditions, the following implicit scheme is attained:

$$\begin{cases} u_{i,j}^{k+1} = \dfrac{1}{(1+(r-1)\tau+q_1/2+q_2/2)}\Big[\dfrac{q_1}{4}\left(u_{i+2,j}^{k+1}+u_{i-2,j}^{k+1}\right)+\dfrac{q_2}{4}\left(u_{i,j+2}^{k+1}+u_{i,j-2}^{k+1}\right), \\ +u_{i,j}^{k}+(r-1)\tau u_{i,j}^{0}+r\tau f_{i,j}^{k+1}\Big],\ 2\le i,j\le M-2,\ 0\le k\le N-1, \qquad (3.3) \\ u_{i,j}^{k}|_{\partial\Omega} = p(x_i,y_j,t_k), \quad 0\le k\le N, \qquad (3.4) \\ u_{i,j}^{0} = g(x_i,y_j), \quad 0\le i,j\le M, \qquad (3.5) \end{cases}$$

where $u_{i,j}^k$ is the approximate numerical solution of $U_{i,j}^k$ after omitting the truncation error terms. The MHEG method can now be constructed by applying Eq (3.3) to a group of four mesh points of the solution domain. This will result in the following (4×4) system of equations:

$$\begin{pmatrix} D & -q_1/4 & 0 & -q_2/4 \\ -q_1/4 & D & -q_2/4 & 0 \\ 0 & -q_2/4 & D & -q_1/4 \\ -q_2/4 & 0 & -q_1/4 & D \end{pmatrix}\begin{pmatrix} u_{i,j}^{k+1} \\ u_{i+2,j}^{k+1} \\ u_{i+2,j+2}^{k+1} \\ u_{i,j+2}^{k+1} \end{pmatrix} = \begin{pmatrix} rhs_{i,j} \\ rhs_{i+2,j} \\ rhs_{i+2,j+2} \\ rhs_{i,j+2} \end{pmatrix}, \tag{3.6}$$

where

$$\begin{aligned} D =& 1+(r-1)\tau+q_1/2+q_2/2, \\ rhs_{i,j} =& \frac{q_1}{4}u_{i-2,j}^{k+1}+\frac{q_2}{4}u_{i,j-2}^{k+1}+u_{i,j}^{k}+(r-1)\tau u_{i,j}^{0}+r\tau f_{i,j}^{k+1}, \\ rhs_{i+2,j} =& \frac{q_1}{4}u_{i+4,j}^{k+1}+\frac{q_2}{4}u_{i+2,j-2}^{k+1}+u_{i+2,j}^{k}+(r-1)\tau u_{i+2,j}^{0}+r\tau f_{i+2,j}^{k+1}, \\ rhs_{i+2,j+2} =& \frac{q_1}{4}u_{i+4,j+2}^{k+1}+\frac{q_2}{4}u_{i+2,j+4}^{k+1}+u_{i+2,j+2}^{k}+(r-1)\tau u_{i+2,j+2}^{0}+r\tau f_{i+2,j+2}^{k+1}, \\ rhs_{i,j+2} =& \frac{q_1}{4}u_{i-2,j+2}^{k+1}+\frac{q_2}{4}u_{i,j+4}^{k+1}+u_{i,j+2}^{k}+(r-1)\tau u_{i,j+2}^{0}+r\tau f_{i,j+2}^{k+1}. \end{aligned}$$

The coefficients matrix in (3.6) can be inverted to get the MHEG equation given as below:

$$\begin{pmatrix} u_{i,j}^{k+1} \\ u_{i+2,j}^{k+1} \\ u_{i+2,j+2}^{k+1} \\ u_{i,j+2}^{k+1} \end{pmatrix} = \frac{1}{\eta} \begin{pmatrix} \eta_1 & \eta_2 & \eta_3 & \eta_4 \\ \eta_2 & \eta_1 & \eta_4 & \eta_3 \\ \eta_3 & \eta_4 & \eta_1 & \eta_2 \\ \eta_4 & \eta_3 & \eta_2 & \eta_1 \end{pmatrix} \begin{pmatrix} rhs_{i,j} \\ rhs_{i+2,j} \\ rhs_{i+2,j+2} \\ rhs_{i,j+2} \end{pmatrix}, \tag{3.7}$$

where

$$\begin{aligned} \eta =& \frac{1}{256}(4 + 4(r-1)\tau + q_1 + q_2)(4 + 4(r-1)\tau + 3q_1 + q_2)(4 + 4(r-1)\tau + q_1 + 3q_2) \\ & \times (4 + 4(r-1)\tau + 3q_1 + 3q_2), \\ \eta_1 =& \frac{1}{32}(2 + 2(r-1)\tau + q_1 + q_2)(16 + 16q_1 + 16q_2 + 8q_1q_2 + 3q_1^2 + 3q_2^2 + 16q_1(r-1)\tau \\ & + 16q_2(r-1)\tau + 32(r-1)\tau + 16(r-1)^2\tau^2), \\ \eta_2 =& \frac{1}{64}q_1(16 + 16q_1 + 16q_2 + 8q_1q_2 + 3q_1^2 + 5q_2^2 + 16q_1(r-1)\tau + 16q_2(r-1)\tau \\ & + 32(r-1)\tau + 16(r-1)^2\tau^2), \\ \eta_3 =& \frac{1}{16}q_1q_2(2 + 2(r-1)\tau + q_1 + q_2), \\ \eta_4 =& \frac{1}{64}q_2(16 + 16q_1 + 16q_2 + 8q_1q_2 + 5q_1^2 + 3q_2^2 + 16q_1(r-1)\tau + 16q_2(r-1)\tau \\ & + 32(r-1)\tau + 16(r-1)^2\tau^2). \end{aligned}$$

Back to Figure 2, it can be seen that the discretized solution domain comprises three different types of mesh points (♦, □, ○). Here, the MHEG method functions through the iterative evaluation of solutions at ♦ mesh points by utilizing Eq (3.7) until a certain convergence criterion is met. Afterwards, the solution values at the remaining mesh points of types □ and ○ are computed directly once in a particular sequence. The computational cost of the iterative method still depends on the number of unknowns involved in the iteration process. For the MHEG method, it is obvious that the number of mesh points taken in the iteration process is lesser than that of the prespecified HSP method, which is expected to speed up the convergence rate of the proposed method. For convenience, the MHEG method in combination with the Gauss-Seidel iterative solver is elaborated in Algorithm 1.

Algorithm 1: MHEG method utilizing the Gauss-Seidel iterative solver

```
1: Branch the mesh points of the discretized solution domain into the following categories:
     • Iterative points of type ♦.
     • Direct points of types □ and ○.
2: Arrange all the ♦ points into groups of four mesh points as depicted in Figure 2.
3: Set an initial guess for the solution at the current time level.
```

4: For each four-point group at the current time level, iterate the intermediate solutions at ♦ mesh points utilizing

$$\begin{pmatrix} \hat{u}_{i,j}^{k+1,n+1} \\ \hat{u}_{i+2,j}^{k+1,n+1} \\ \hat{u}_{i+2,j+2}^{k+1,n+1} \\ \hat{u}_{i,j+2}^{k+1,n+1} \end{pmatrix} = \frac{1}{\eta} \begin{pmatrix} \eta_1 & \eta_2 & \eta_3 & \eta_4 \\ \eta_2 & \eta_1 & \eta_4 & \eta_3 \\ \eta_3 & \eta_4 & \eta_1 & \eta_2 \\ \eta_4 & \eta_3 & \eta_2 & \eta_1 \end{pmatrix} \begin{pmatrix} rhs_{i,j} \\ rhs_{i+2,j} \\ rhs_{i+2,j+2} \\ rhs_{i,j+2} \end{pmatrix},$$

where

$$\begin{aligned} rhs_{i,j} =& \frac{q_1}{4} u_{i-2,j}^{k+1} + \frac{q_2}{4} u_{i,j-2}^{k+1} + u_{i,j}^{k} + (r-1)\tau u_{i,j}^{0} + r\tau f_{i,j}^{k+1}, \\ rhs_{i+2,j} =& \frac{q_1}{4} u_{i+4,j}^{k+1} + \frac{q_2}{4} u_{i+2,j-2}^{k+1} + u_{i+2,j}^{k} + (r-1)\tau u_{i+2,j}^{0} + r\tau f_{i+2,j}^{k+1}, \\ rhs_{i+2,j+2} =& \frac{q_1}{4} u_{i+4,j+2}^{k+1} + \frac{q_2}{4} u_{i+2,j+4}^{k+1} + u_{i+2,j+2}^{k} + (r-1)\tau u_{i+2,j+2}^{0} + r\tau f_{i+2,j+2}^{k+1}, \\ rhs_{i,j+2} =& \frac{q_1}{4} u_{i-2,j+2}^{k+1} + \frac{q_2}{4} u_{i,j+4}^{k+1} + u_{i,j+2}^{k} + (r-1)\tau u_{i,j+2}^{0} + r\tau f_{i,j+2}^{k+1}, \end{aligned}$$

and perform the Gauss-Seidel solver

$$\begin{pmatrix} u_{i,j}^{k+1,n+1} \\ u_{i+2,j}^{k+1,n+1} \\ u_{i+2,j+2}^{k+1,n+1} \\ u_{i,j+2}^{k+1,n+1} \end{pmatrix} = \omega \begin{pmatrix} \hat{u}_{i,j}^{k+1,n+1} \\ \hat{u}_{i+2,j}^{k+1,n+1} \\ \hat{u}_{i+2,j+2}^{k+1,n+1} \\ \hat{u}_{i,j+2}^{k+1,n+1} \end{pmatrix} + (1-\omega) \begin{pmatrix} u_{i,j}^{k+1,n} \\ u_{i+2,j}^{k+1,n} \\ u_{i+2,j+2}^{k+1,n} \\ u_{i,j+2}^{k+1,n} \end{pmatrix},$$

where n is the iteration number and $\omega = 1$ is the relaxation parameter of the Gauss-Seidel solver. η, η_1, η_2, η_3 and η_4 are as defined before in this section.

5: Test the convergence. If convergence is achieved, go to step 6. Otherwise, step 4 is repeated until convergence is attained.

6: The solutions on the rest of mesh points of types □ and ○ are evaluated directly once in the following manner:

- For □ points, the approximating PDE (2.2) is discretized by using skewed finite difference operators. Such difference operators are derived by rotating the standard mesh an angle 45° clockwise and applying the Taylor series expansion afterwards [45]. This will lead to the following skewed difference scheme for (2.2):

$$\begin{aligned} u_{i,j}^{k+1} =& \frac{1}{(1+(r-1)\tau + q_1 + q_2)} \Big[\frac{q_1}{2} \left(u_{i+1,j-1}^{k+1} + u_{i-1,j+1}^{k+1} \right) \\ & + \frac{q_2}{2} \left(u_{i+1,j+1}^{k+1} + u_{i-1,j-1}^{k+1} \right) + u_{i,j}^{k} + (r-1)\tau u_{i,j}^{0} + r\tau f_{i,j}^{k+1} \Big]. \end{aligned}$$

- For ○ points, the difference scheme formula defined by (2.6) is employed.

7: If the final time level is reached, go to step 8. Otherwise, adopt the solution values of the last time level as an initial guess for the next time level and go to step 4.

8: Display the numerical solutions.

4. Stability and convergence

In this section, we turn to consider the stability and convergence of the MHEG scheme defined by (3.7) by using the matrix analysis method. For the convenience of our analysis, three remarks are given at first.

Remark 4.1. Let $z \in \mathbb{N}$. The symbol $||A_{z\times z}||$ denotes the infinity norm of matrix $A = [a_{i,j}]$ which is defined as,

$$||A_{z\times z}||_{\infty} = \max_{1\le i\le z}\left\{\sum_{j=1}^{z}|a_{i,j}|\right\}.$$

Remark 4.2. $A_{z\times z}$ is called a stricatly diagonally dominant (SDD) matrix if $|a_{i,i}| > r_i(A)$, where $r_i(A)$ is the i-th deleted absolute row sum (i.e., $r_i(A) = \sum_{j\neq i,j=1}^{z}|a_{i,j}|$, $1 \le i \le z$).

Remark 4.3. Given that $A_{z\times z}$ is a SDD matrix, then $A_{z\times z}$ is non-singular and the following norm's bound does hold [44]:

$$||A^{-1}||_{\infty} \le \frac{1}{\min_{1\le i\le M}\left\{|a_{i,i}| - r_i(A)\right\}}.$$

4.1. Stability analysis

Here, we analyze the stability of the MHEG scheme (3.7). For simplicity and without loss of generality, we assume that $q_1 = q_2 = q = \tau/h^2$. Indeed, the MHEG scheme (3.7) can be represented in the following matrix form:

$$Au^{k+1} = Bu^k + Cu^0 + b, \quad 0 \le k \le N-1, \tag{4.1}$$

where $u^k$is an $\frac{(M-2)^2}{4}$-dimensional vector expressed in the block form

$$u^k = (u_1^k, u_2^k, \dots, u^k_{\frac{(M-2)^2}{16}})^T,\ u_l^k = (u_{i,j}^k, u_{i+2,j}^k, u_{i+2,j+2}^k, u_{i,j+2}^k)^T,\ 1 \le l \le \frac{(M-2)^2}{16},$$

and

$$A = \begin{pmatrix} J_1 & J_2 & & & \\ J_3 & J_1 & J_2 & & \\ & & \ddots & & \\ & & J_3 & J_1 & J_2 \\ & & & J_3 & J_1 \end{pmatrix}, B = \begin{pmatrix} H_1 & & & & \\ & H_1 & & & \\ & & \ddots & & \\ & & & H_1 & \\ & & & & H_1 \end{pmatrix}, b = \begin{pmatrix} S_1 \\ S_1 \\ \vdots \\ S_1 \\ S_1 \end{pmatrix},$$

$$C = \begin{pmatrix} P_1 & & & & \\ & P_1 & & & \\ & & \ddots & & \\ & & & P_1 & \\ & & & & P_1 \end{pmatrix}, J_1 = \begin{pmatrix} Q_1 & Q_3 & & & \\ Q_2 & Q_1 & Q_3 & & \\ & & \ddots & & \\ & & Q_2 & Q_1 & Q_3 \\ & & & Q_2 & Q_1 \end{pmatrix},$$

$$J_2 = \begin{pmatrix} Q_5 & & & & \\ & Q_5 & & & \\ & & \ddots & & \\ & & & Q_5 & \\ & & & & Q_5 \end{pmatrix}, J_3 = \begin{pmatrix} Q_4 & & & & \\ & Q_4 & & & \\ & & \ddots & & \\ & & & Q_4 & \\ & & & & Q_4 \end{pmatrix},$$

$$H_1 = \begin{pmatrix} I_4 & & & & \\ & I_4 & & & \\ & & \ddots & & \\ & & & I_4 & \\ & & & & I_4 \end{pmatrix}, P_1 = \begin{pmatrix} T_1 & & & & \\ & T_1 & & & \\ & & \ddots & & \\ & & & T_1 & \\ & & & & T_1 \end{pmatrix}, S_1 = \begin{pmatrix} G_1 \\ G_1 \\ \vdots \\ G_1 \\ G_1 \end{pmatrix},$$

$$Q_1 = \begin{pmatrix} 1 + (r-1)\tau + q & -q/4 & 0 & -q/4 \\ \frac{-q}{4} & 1 + (r-1)\tau + q & -q/4 & 0 \\ 0 & -q/4 & 1 + (r-1)\tau + q & -q/4 \\ -q/4 & 0 & -q/4 & 1 + (r-1)\tau + q \end{pmatrix},$$

$$Q_2 = \begin{pmatrix} 0 & 0 & 0 & -q/4 \\ 0 & 0 & -q/4 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, Q_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -q/4 & 0 & 0 \\ -q/4 & 0 & 0 & 0 \end{pmatrix},$$

$$Q_4 = \begin{pmatrix} 0 & -q/4 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -q/4 & 0 \end{pmatrix}, Q_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -q/4 & 0 & 0 & 0 \\ 0 & 0 & 0 & -q/4 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$T_1 = \begin{pmatrix} (r-1)\tau & 0 & 0 & 0 \\ 0 & (r-1)\tau & 0 & 0 \\ 0 & 0 & (r-1)\tau & 0 \\ 0 & 0 & 0 & (r-1)\tau \end{pmatrix},$$

$$I_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, G_1 = r\tau \begin{pmatrix} f_{i,j} \\ f_{i+2,j} \\ f_{i+2,j+2} \\ f_{i,j+2} \end{pmatrix}.$$

Taking note of the above block matrices, the matrix representation in (4.1) can be rewritten in the more elaborated general form

$$\left[A_{\frac{(M-2)^2}{4} \times \frac{(M-2)^2}{4}}\right] u^{k+1} = \left[B_{\frac{(M-2)^2}{4} \times \frac{(M-2)^2}{4}}\right] u^k + \left[C_{\frac{(M-2)^2}{4} \times \frac{(M-2)^2}{4}}\right] u^0 + b. \tag{4.2}$$

Next, we prove the following result:

Theorem 4.1. The MHEG scheme defined by (4.2) is unconditionally stable.

Proof. Suppose u^k and $\tilde{u}^k$ are respectively the exact and approximate solutions of (4.2). Then the error at time level k is expressed as $e^k = u^k - \tilde{u}^k$. From Remarks 4.2 and 4.3, it follows that A is invertible and therefore Eq (4.2) can be rewritten as

$$u^{k+1} = A^{-1}Bu^k + A^{-1}Cu^0 + A^{-1}b, \quad 0 \le k \le N-1. \tag{4.3}$$

The error function e^k satisfies (4.3) and leads to the following round-off error equation:

$$e^{k+1} = A^{-1}Be^k + A^{-1}Ce^0, \quad 0 \le k \le N-1, \tag{4.4}$$

where

$$e^{k+1} = \begin{pmatrix} e_0^{k+1} \\ e_0^{k+1} \\ \vdots \\ e_0^{k+1} \end{pmatrix}, e_0^{k+1} = \begin{pmatrix} \psi_1^{k+1} \\ \psi_2^{k+1} \\ \vdots \\ \psi_{\frac{(M-2)^2}{16}}^{k+1} \end{pmatrix}, \psi^{k+1} = \begin{pmatrix} \psi_{i,j}^{k+1} \\ \psi_{i+2,j}^{k+1} \\ \psi_{i+2,j+2}^{k+1} \\ \psi_{i,j+2}^{k+1} \end{pmatrix},$$

and $\psi_{i,j}^{k+1} = u_{i,j}^{k+1} - \tilde{u}^{k+1}$.

In order to demonstrate the stability, we shall prove that $||e^{k+1}|| \le ||e^0||$ for all $0 \le k \le N-1$. To this end, the mathematical induction will be used. For $k = 0$, we obtain

$$e^1 = A^{-1}Be^0 + A^{-1}Ce^0.$$

Using the fact that the matrix and the vector infinity norms are consistent, yields

$$\begin{aligned} ||e^1|| &= ||A^{-1}Be^0 + A^{-1}Ce^0|| \\ &\le ||A^{-1}B||||e^0|| + ||A^{-1}C||||e^0|| \\ &\le ||A^{-1}||||B||||e^0|| + ||A^{-1}||||C||||e^0|| \\ &= (||B|| + ||C||)||A^{-1}||||e^0||. \end{aligned}$$

Define $r_i(A)$ as in Remark 4.2. Then using Remark 4.3, we get

$$\begin{aligned} ||e^1|| &\le \frac{(||B|| + ||C||)}{\min_{1\le i\le M}\left\{|a_{i,i}| - r_i(A)\right\}}||e^0|| \\ & \frac{1+(r-1)\tau}{|1+(r-1)\tau+q| - (|-q/4| + |-q/4| + |-q/4| + |-q/4|)} \\ &= \frac{1+(r-1)\tau}{1+(r-1)\tau}||e^0|| = ||e^0||. \end{aligned}$$

Next, suppose that

$$||e^{s+1}|| \le ||e^0||, \quad s = 1, 2, \ldots, k-1. \tag{4.5}$$

We will show the above inequality is true for $s = k$. Noticing (4.4) and (4.5), we get

$$\begin{aligned} ||e^{k+1}|| &= ||A^{-1}Be^k + A^{-1}Ce^0|| \\ &\le ||A^{-1}||||B||||e^k|| + ||A^{-1}||||C||||e^0|| \\ &\le ||A^{-1}||||B||||e^0|| + ||A^{-1}||||C||||e^0|| \end{aligned}$$

$$
\begin{aligned}
&\leq \frac{(\|B\|+\|C\|)}{\min_{1\leq i\leq M}\left\{|a_{i,i}|-r_i(A)\right\}}\|e^0\| \\
&= \frac{1+(r-1)\tau}{|1+(r-1)\tau+q|-(|-q/4|+|-q/4|+|-q/4|+|-q/4|)} \\
&= \frac{1+(r-1)\tau}{1+(r-1)\tau}\|e^0\| = \|e^0\|.
\end{aligned}
$$

Thus, the proof is completed. □

4.2. Convergence analysis

We now analyze the convergence of the MHEG scheme (3.7) by following a similar approach to that in the previous subsection. As a start, we suppose that the truncation errors on each four-point group of mesh points at any time level are represented by the block vector of the following form:

$$
\begin{aligned}
R^{k+1} &= (R_1^{k+1}, R_2^{k+1}, \ldots, R_{\frac{(M-2)^2}{16}}^{k+1})^T, \\
R_l^{n+1} &= (R_{i,j}^{k+1}, R_{i+2,j}^{k+1}, R_{i+2,j+2}^{k+1}, R_{i,j+2}^{k+1})^T,\ 1 \leq l \leq \frac{(M-2)^2}{16}.
\end{aligned}
$$

Noticing Eq (3.2) and since i, j and k are finite, there is a positive constant C^* such that

$$\|R^{k+1}\| \leq C^*(\tau + h^2), \quad 0 \leq k \leq N-1. \tag{4.6}$$

Subtracting (4.2) from the following equation:

$$AU^{k+1} = BU^k + CU^0 + b + R^{k+1},$$

will result in error equation given by

$$AE^{k+1} = BE^k + CE^0 + R^{k+1}, \tag{4.7}$$

where

$$
E^{k+1} = \begin{pmatrix} E_0^{k+1} \\ E_0^{k+1} \\ \vdots \\ E_0^{k+1} \end{pmatrix}, E_0^{k+1} = \begin{pmatrix} \phi_1^{k+1} \\ \phi_2^{k+1} \\ \vdots \\ \phi_{\frac{(M-2)^2}{16}}^{k+1} \end{pmatrix}, \phi^{k+1} = \begin{pmatrix} \phi_{i,j}^{k+1} \\ \phi_{i+2,j}^{k+1} \\ \phi_{i+2,j+2}^{k+1} \\ \phi_{i,j+2}^{k+1} \end{pmatrix},
$$

and $\phi_{i,j}^{k+1} = U_{i,j}^{k+1} - u_{i,j}^{k+1}$.

The convergence property is given in the next theorem.

Theorem 4.2. Assume problem (1.1) has a smooth solution $u(x,y,t)$. Then, the MHEG scheme defined by (3.7) is convergent and the convergence order is $O(\tau + h^2)$.

Proof. We will use mathematical induction for the proof. For $k = 0$ and using that $E^0 = 0$, we obtain

$$E^1 = A^{-1}R^1.$$

In view of of Remark 4.3 and utilizing (4.6), yields

$$\begin{aligned}
||E^1|| &= ||A^{-1}R^1|| \\
&\leq ||A^{-1}||||R^1|| \\
&\leq \frac{1}{\min_{1\leq i\leq M}\left\{|a_{i,i}| - r_i(A)\right\}} C^*(\tau + h^2) \\
&= \frac{1}{1+(r-1)\tau} C^*(\tau + h^2) = C_0(\tau + h^2),
\end{aligned}$$

where $C_0 = C^*/(1 + (r-1)\tau)$.

$$\therefore ||E^1|| \leq C_0\left(\tau + h^2\right).$$

Now, assume that

$$E^{s+1} \leq C_s(\tau + h^2), \quad s = 1, 2, \ldots, k-1. \tag{4.8}$$

We prove the above result is true for $s = k$. Noticing (4.7) and (4.8), we get

$$\begin{aligned}
||E^{k+1}|| &= ||A^{-1}BE^k + A^{-1}R^{k+1}|| \\
&\leq ||A^{-1}||||B||||E^k|| + ||A^{-1}||||R^{k+1}|| \\
&\leq \frac{1}{\min_{1\leq i\leq M}\left\{|a_{i,i}| - r_i(A)\right\}} \left[C_{k-1}(\tau + h^2) + C^*(\tau + h^2)\right] \\
&= \frac{1}{1+(r-1)\tau}(C_{k-1} + C^*)(\tau + h^2) \\
&= C_k(\tau + h^2),
\end{aligned}$$

where $C_k = C_{k-1} + C^*$ as $\lim_{n\to\infty} \tau = 0$.

$$\therefore ||E^{n+1}|| \leq C_k\left(\tau + h^2\right), \quad 0 \leq n \leq N-1.$$

Hence, the proof is completed by induction. □

4.3. Solvability

Theorem 4.3. The MHEG scheme defined by (3.7) is uniquely solvable.

Proof. In view of (4.1), The coefficients matrix A of the MHEG scheme (3.7) is an SDD matrix. Consequently, and based on Remark 4.3, it follows that A is a non-singular matrix. This proves the existence and uniqueness of the solution of the MHEG scheme. □

5. Numerical simulations

In this section, we report on numerical simulations for (1.1)–(1.3). In this endeavor, two test problems with known exact solutions are given to demonstrate the accuracy and efficiency of the newly developed MHEG method. For comparison purposes, the fast HSP method proposed recently in [34] is applied to solve the two test problems. All simulations using the both methods are performed in Mathematica 11.3 software and run on a computer with an Intel (R) Core (TM) i7-8550U CPU and 8.00 GB of RAM. In these simulations, the Gauss-Seidel iterative solver with a stopping criterion selected to be 10^{-5} are employed to generate the corresponding numerical results. To get more insight into these results, and based on the total number of arithmetic operations to be implemented before and after the convergence process, an analysis on the computational complexity of the MHEG and HSP iterative methods is established and presented in Table 1.

Table 1. The computational complexity of the HSP [34] and the MHEG iterative methods ($\sigma = M - 1$).

Method	Per iteration	After convergence	Total operations
HSP	$13\sigma^2 * Ite$	-	$13\sigma^2 * Ite$
MHEG	$4.5(\sigma - 1)^2 * Ite$	$3.25(3\sigma^2 + 2\sigma - 1)$	$4.5(\sigma - 1)^2 * Ite + 3.25(3\sigma^2 + 2\sigma - 1)$

Test problem 5.1. We consider the following diffusion equation of fractional order:

$$ {}_0^C D_t^\alpha u(x,y,t) = \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} + \left(\frac{2t^{2-\alpha}}{\Gamma(3-\alpha)} + 2t^2\right)\sin(x)\sin(y), $$

subject to the boundary and initial conditions

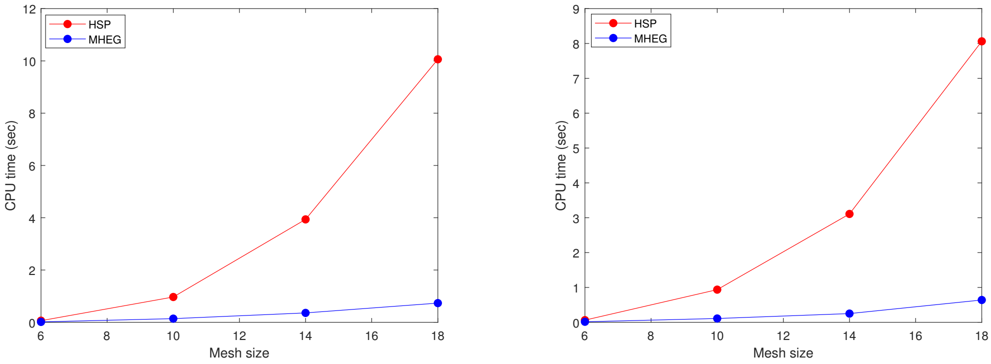
$$ u(x,y,t)|_{\partial\Omega} = t^2 \sin(x)\sin(y), \quad u(x,y,0) = 0, $$

in the spatial domain $(0,L) \times (0,L) = (0,1)^2$, $T = 1$, and the exact analytic solution is $u(x,y,t) = t^2 \sin(x)\sin(y)$.

In this test, numerical outputs including maximum absolute error $E_{max} = \max_{1\le i,j\le M-1} |u(x_i, y_j, t_N) - u_{i,j}^N|$, CPU time (in seconds), number of iterations (Ite) as well as total number of arithmetic operations are computed for different values of τ, h and α and listed in Table 2. It can be observed that the MHEG iterative method results in significantly faster simulations without compromising too much of the accuracy when compared to the HSP iterative method. To illustrate this further, Figures 3 and 4 show respectively the graphs of CPU time and total operations of both methods against various mesh sizes. Form these figures, it is apparent that the shape of the computing time plots is consistent with the shape of the total computing effort plots. This indicates that the experimental results are in good agreement with our theoretical analysis. The comparison of the exact and the numerical solutions besides the graphical error representation when $\tau = h = 1/18$ and $\alpha = 0.55$ are depicted in Figures 5 and 6, respectively. A comparison of the numerical errors of the test problems for different fractional orders shows that the accuracy of the proposed method is greatest at $\alpha = 0.55$, followed by $\alpha = 0.75$ and $\alpha = 0.95$. It is clear that the numerical solution matches well with the exact solution.

Table 2. The numerical results of the MHEG and HSP [34] iterative methods in Test problem 5.1.

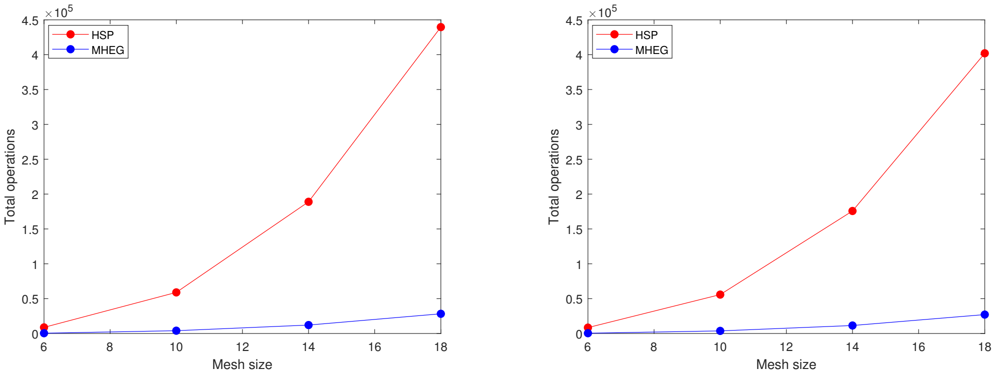
α	$\tau = h$	Method	CPU time (sec)	*Ite*	Total operations	E_{max}
0.55						
	1/6	HSP	0.062	27	8,775	1.6737E-03
		MHEG	0.015	2	417	1.7983E-03
	1/10	HSP	0.968	56	58,968	9.5405E-04
		MHEG	0.140	11	4,013	1.0764E-03
	1/14	HSP	3.937	86	188,942	6.0571E-04
		MHEG	0.359	16	12,097	7.6144E-04
	1/18	HSP	10.062	117	439,569	3.8681E-04
		MHEG	0.734	22	28,269	5.7584E-04
0.75						
	1/6	HSP	0.062	26	8,450	2.3916E-03
		MHEG	0.015	2	417	2.4750E-03
	1/10	HSP	0.937	53	55,809	1.4477E-03
		MHEG	0.109	10	3,725	1.5593E-03
	1/14	HSP	3.109	80	175,760	9.9823E-04
		MHEG	0.250	15	11,449	1.1488E-03
	1/18	HSP	8.062	107	401,999	6.9546E-04
		MHEG	0.640	21	27,117	9.0926E-04
0.95						
	1/6	HSP	0.062	25	8,125	2.9204E-03
		MHEG	0.015	2	417	2.9789E-03
	1/10	HSP	0.656	50	52,650	1.7502E-03
		MHEG	0.062	10	3,725	1.8611E-03
	1/14	HSP	2.562	75	164,775	1.1935E-03
		MHEG	0.203	14	10,801	1.3410E-03
	1/18	HSP	7.937	100	375,700	8.2842E-04
		MHEG	0.515	19	24,813	1.0461E-03



(a) $\alpha = 0.55$

(b) $\alpha = 0.75$

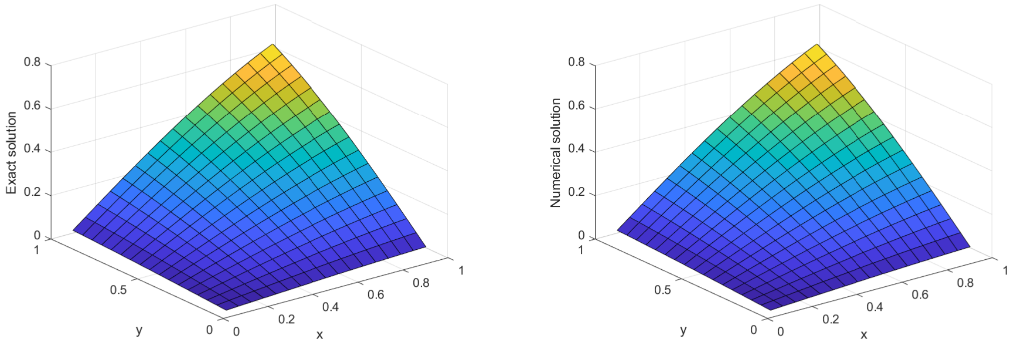
Figure 3. Graphs of CPU times for Test problem 5.1.



(a) $\alpha = 0.55$

(b) $\alpha = 0.75$

Figure 4. Graphs of Computing efforts for Test problem 5.1.



(a) Exact solution

(b) Numerical solution

Figure 5. The exact solution and the numerical solution for Test problem 5.1 when $T = 1$, $\tau = h = 1/18$ and $\alpha = 0.55$.

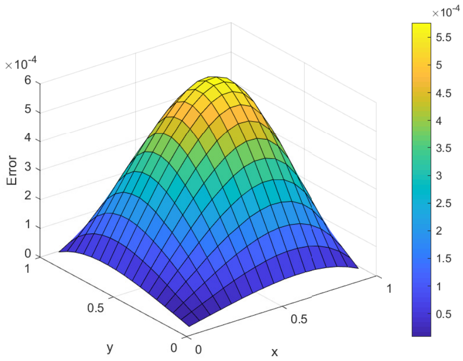


Figure 6. The maximum absolute errors of the MHEG method for the Test problem 5.1 at $T = 1$, $\tau = h = 1/18$ and $\alpha = 0.55$.

Test problem 5.2. We consider another diffusion equation of fractional order as follows:

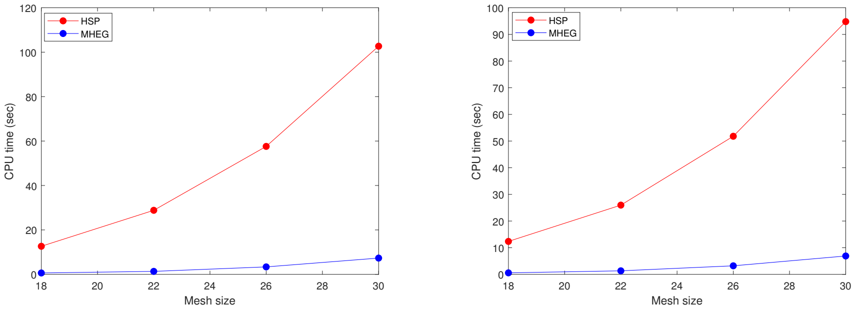
$$ {}_0^C D_t^\alpha u(x,y,t) = \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} + \left(\frac{2t^{2-\alpha}}{\Gamma(3-\alpha)} - 2t^2 \right) e^{(x+y)}, $$

with the boundary and initial conditions

$$ u(x,y,t)|_{\partial\Omega} = t^2 e^{(x+y)}, \quad u(x,y,0) = 0, $$

in the spatial domain $\Omega = (0,1)^2$, $T = 1$, and the exact analytic solution $u(x,y,t) = t^2 e^{(x+y)}$.

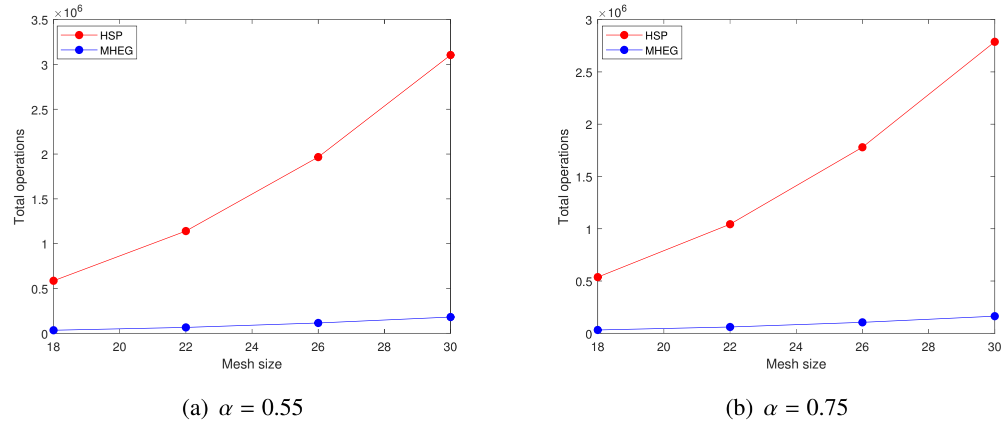
The numerical results of solving this problem for various values of τ, h and α are presented in Table 3. Figure 7 highlights the CPU time plots versus several mesh sizes. In view of Table 3 and Figure 7, it can be seen that the MHEG iterative method reduces the CPU time greatly without jeopardizing the accuracy of numerical solutions compared to the HSP iterative method. This can be attributed to the reduction in the number of executed arithmetic operations that leads to a lower computational complexity as shown in Figure 8. Figure 9 introduces the comparison between the exact and the numerical solutions, whereas Figure 10 presents the 3D plot of the maximum errors when $\tau = h = 1/30$ and $\alpha = 0.75$. Again, these demonstrate the effectiveness of the proposed MHEG method. Table 4 illustrates the experimental convergence order in the temporal direction using the formula $\rho(\tau,h) = \log_2 (E_{max}(2\tau,h)/E_{max}(\tau,h))$. It can be observed that the performance is in good agreement with the theoretical analysis.



(a) $\alpha = 0.55$

(b) $\alpha = 0.75$

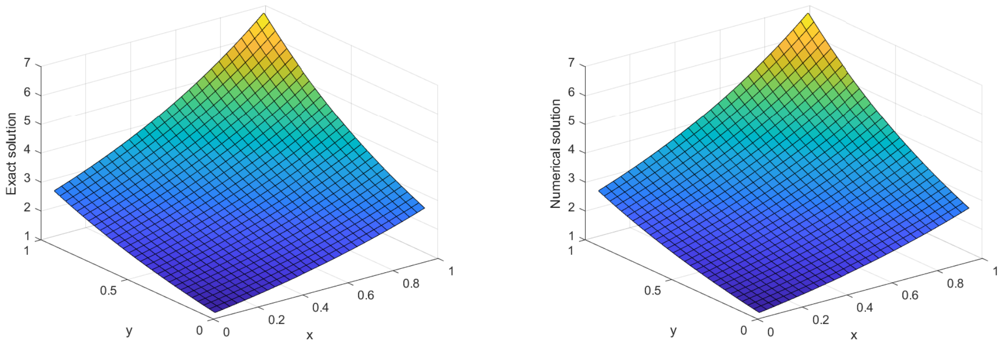
Figure 7. Graphs of CPU times for Test problem 5.2.



(a) $\alpha = 0.55$

(b) $\alpha = 0.75$

Figure 8. Graphs of Computing efforts for Test problem 5.2



(a) Exact solution

(b) Numerical solution

Figure 9. The exact solution and the numerical solution for Test problem 5.2 when $T = 1$, $\tau = h = 1/30$ and $\alpha = 0.75$.

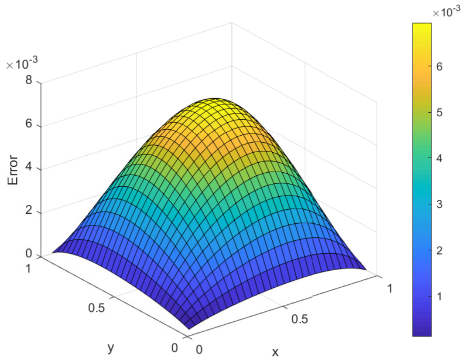


Figure 10. The maximum absolute errors of the MHEG method for the Test problem 5.2 at $T = 1$, $\tau = h = 1/30$ and $\alpha = 0.75$.

Table 3. The numerical results of the MHEG and HSP [34] iterative methods in Test problem 5.2.

α	$\tau = h$	Method	CPU time (sec)	*Ite*	Total operations	E_{max}
0.55						
	1/18	HSP	12.671	170	638,690	6.3352E-03
		MHEG	0.859	29	36,333	6.8017E-03
	1/22	HSP	31.906	221	1,266,993	5.0040E-03
		MHEG	2.359	38	72,833	5.5331E-03
	1/26	HSP	64.234	271	2,201,875	4.0351E-03
		MHEG	3.390	46	125,485	4.6340E-03
	1/30	HSP	118.625	320	3,498,560	3.2629E-03
		MHEG	7.937	55	202,425	3.8992E-03
0.75						
	1/18	HSP	12.640	156	586,092	1.0239E-02
		MHEG	0.625	27	34,029	1.0665E-02
	1/22	HSP	28.828	199	1,140,867	8.4919E-03
		MHEG	1.343	34	65,633	9.0001E-03
	1/26	HSP	57.593	242	1,966,250	7.1948E-03
		MHEG	3.359	42	115,117	7.8080E-03
	1/30	HSP	102.688	284	3,104,972	6.1807E-03
		MHEG	7.328	49	181,257	6.9414E-03
0.95						
	1/18	HSP	12.359	143	537,251	1.1950E-02
		MHEG	0.562	25	31,725	1.2403E-02
	1/22	HSP	25.968	182	1,043,406	9.7841E-03
		MHEG	1.328	31	60,233	1.0285E-02
	1/26	HSP	51.796	219	1,779,375	8.1651E-03
		MHEG	3.203	38	104,749	8.8048E-03
	1/30	HSP	94.796	255	2,787,915	6.9147E-03
		MHEG	6.890	44	163,617	7.6882E-03

Table 4. Numerical error and computational order with $\alpha = 0.55$ and $h = 1/22$.

Test problem	τ	E_{max}	Computational order
1	1/10	9.9154E-04	0.9835
	1/20	5.0149E-04	
2	1/10	1.1829E-02	0.9658
	1/20	6.0570E-03	

6. Conclusions

In this paper, a new MHEG iterative method for solving the two-dimensional diffusion equation with time fractional derivative has been developed. The unique solvability, unconditional stability and convergence are proved by the matrix analysis method. Moreover, the feasibility and efficiency of the proposed numerical method are confirmed through the computational outputs drawn from the conducted numerical simulations. The advantages of being uncomplicated, easy to implement and diminishing the amount of computational complexity indicate that the proposed method and numerical analysis in this article can be extended to solve other types of non-linear and variable order fractional differential equations. In addition, the parallel implementation of the proposed MHEG method is an interesting line to study which can be considered as a part of future work.

Acknowledgments

This research was supported by Ministry of Higher Education Malaysia, Fundamental Research Grant Scheme with Project Code: FRGS/1/2019/STG06/USM/01/1. We would like to express our gratitude to the anonymous reviewers for their valuable comments.

Conflict of interest

The authors have no conflict of interest.

References

1. H. R. Ghehsareh, A. Zaghian, S. M. Zabetzadeh, The use of local radial point interpolation method for solving two-dimensional linear fractional cable equation, *Neural Comput. Appl.*, **29** (2018), 745–754. doi: 10.1007/s00521-016-2595-y.
2. Y. L. Zhao, T. Z. Huang, X. M. Gu, W. H. Luo, A fast second-order implicit difference method for time-space fractional advection-diffusion equation, *Numer. Func. Anal. Opt.*, **41** (2020), 257–293. doi: 10.1080/01630563.2019.1627369.
3. M. Hussain, S. Haq, Weighted meshless spectral method for the solutions of multi-term time fractional advection-diffusion problems arising in heat and mass transfer, *Int. J. Heat Mass Tran.*, **129** (2019), 1305–1316. doi: 10.1016/j.ijheatmasstransfer.2018.10.039.
4. M. Abbaszadeh, A. Mohebbi, A fourth-order compact solution of the two dimensional modified anomalous fractional sub-diffusion equation with a nonlinear source term, *Comput. Math. Appl.*, **66** (2013), 1345–1359. doi: 10.1016/j.camwa.2013.08.010.
5. G. M. Zaslavsky, Chaos, fractional kinetics, and anomalous transport, *Phys. Rep.*, **371** (2002), 461–580. doi: 10.1016/S0370-1573(02)00331-9.
6. S. G. Samko, A. A. Kilbas, O. I. Marichev, *Fractional integrals and derivatives: Theory and applications*, Yverdon: Gordon and Breach, 1993.
7. I. Podlubny, *Fractional differential equations, mathematics in science and engineering*, New York: Academic Press, 1999.
8. B. Guo, X. Pu, F. Huang, *Fractional partial differential equations and their numerical solutions*, Singapore: World Scientific, 2015.
9. A. Kochubei, Y. Luchko, V. E. Tarasov, I. Petra, *Handbook of fractional calculus with applications*, Berlin: De Gruyter Grand Forks, 2019.

10. J. Y. Shen, Z. Z. Sun, R. Du, Fast finite difference schemes for time fractional diffusion equations with a weak singularity at initial time, *E. Asian J. Appl. Math.*, **8** (2018), 834–858. doi: 10.4208/eajam.010418.020718.

11. A. Chen, C. Li, A novel compact adi scheme for the time-fractional subdiffusion equation in two space dimensions, *Int. J. Comput. Math.*, **93** (2016), 889–914. doi: 10.1080/00207160.2015.1009905.

12. G. H. Gao, Z. Z. Sun, Y. N. Zhang, A finite difference scheme for fractional sub-diffusion equations on an unbounded domain using artificial boundary conditions, *J. Comput. Phys.*, **231** (2012), 2865–2879. doi: 10.1016/j.jcp.2011.12.028.

13. M. Tamsir, N. Dhiman, D. Nigam, A. Chauhan, Approximation of Caputo time-fractional diffusion equation using redefined cubic exponential B-spline collocation technique, *AIMS Mathematics*, **6** (2021), 3805–3820. doi: 10.3934/math.2021226.

14. J. Shen, X. M. Gu, Two finite difference methods based on an H2N2 interpolation for two-dimensional time fractional mixed diffusion and diffusion-wave equations, *Discrete Cont. Dyn. B*, 2021. doi: 10.3934/dcdsb.2021086.

15. X. M. Gu, Y. L. Zhao, X. L. Zhao, B. Carpentieri, Y. Y. Huang, A note on parallel preconditioning for the all-at-once solution of Riesz fractional diffusion equations, *Numer. Math. Theor. Meth. Appl.*, **14** (2021), 893–919. doi: 10.4208/nmtma.OA-2020-0020.

16. X. M. Gu, H. W. Sun, Y. L. Zhao, X. Zheng, An implicit difference scheme for time-fractional diffusion equations with a time-invariant type variable order, *Appl. Math. Lett.*, **120** (2021), 107270. doi: 10.1016/j.aml.2021.107270.

17. Y. Xu, Y. Zhang, J. Zhao, Backward difference formulae and spectral galerkin methods for the riesz space fractional diffusion equation, *Math. Comput. Simulat.*, **166** (2019), 494–507. doi: 10.1016/j.matcom.2019.07.007.

18. X. Gao, B. Yin, H. Li, Y. Liu, Tt-m FE method for a 2D nonlinear time distributed-order and space fractional diffusion equation, *Math. Comput. Simulat.*, **181** (2021), 117–137. doi: 10.1016/j.matcom.2020.09.021.

19. X. Zheng, H. Wang, An error estimate of a numerical approximation to a hidden-memory variable-order space-time fractional diffusion equation, *SIAM J. Numer. Anal.*, **58** (2020), 2492–2514. doi: 10.1137/20M132420X.

20. X. Zheng, H. Wang, A Hidden-memory variable-order time-fractional optimal control model: Analysis and approximation, *SIAM J. Control Optim.*, **59** (2021), 1851–1880. doi: 10.1137/20M1344962.

21. X. Zheng, H. Wang, Optimal-order error estimates of finite element approximations to variable-order time-fractional diffusion equations without regularity assumptions of the true solutions, *IMA J. Numer. Anal.*, **41** (2021), 1522–1545. doi: 10.1093/imanum/draa013.

22. H. Wang, X. Zheng, Wellposedness and regularity of the variable-order time-fractional diffusion equations, *J. Math. Anal. Appl.*, **475** (2019), 1778–1802. doi: 10.1016/j.jmaa.2019.03.052.

23. Y. Luchko, Initial-boundary-value problems for the one-dimensional time-fractional diffusion equation, *Fract. Calc. Appl. Anal.*, **15** (2012), 141–160. doi: 10.2478/s13540-012-0010-7.

24. B. Jin, B. Li, Z. Zhou, Numerical analysis of nonlinear subdiffusion equations, *SIAM J. Numer. Anal.*, **56** (2018), 1–23. doi: 10.1137/16M1089320.

25. X. Zheng, H. Wang, Wellposedness and regularity of a variable-order space-time fractional diffusion equation, *Anal. Appl.*, **18** (2020), 615–638. doi: 10.1142/S0219530520500013.

26. H. Fu, M. K. Ng, H. Wang, A divide-and-conquer fast finite difference method for space-time fractional partial differential equation, *Comput. Math. Appl.*, **73** (2017), 1233–1242. doi: 10.1016/j.camwa.2016.11.023.

27. F. M. Salama, N. H. M. Ali, Fast O(N) hybrid method for the solution of two dimensional time fractional cable equation, *Compusoft*, **8** (2019), 3453–3461.

28. F. M. Salama, N. H. M. Ali, Computationally efficient hybrid method for the numerical solution of the 2D time fractional advection-diffusion equation, *Int. J. Math. Eng. Manag.*, **5** (2020), 432–446. doi: 10.33889/IJMEMS.2020.5.3.036.

29. X. M. Gu, S. L. Wu, A parallel-in-time iterative algorithm for Volterra partial integro-differential problems with weakly singular kernel, *J. Comput. Phys.*, **417** (2020), 109576. doi: 10.1016/j.jcp.2020.109576.

30. X. L. Lin, M. K. Ng, H. W. Sun, A multigrid method for linear systems arising from time-dependent two-dimensional space-fractional diffusion equations, *J. Comput. Phys.*, **336** (2017), 69–86. doi: 10.1016/j.jcp.2017.02.008.

31. J. Ren, Z. Z. Sun, W. Dai, New approximations for solving the caputo-type fractional partial differential equations, *Appl. Math. Model.,* **40** (2016), 2625–2636. doi: 10.1016/j.apm.2015.10.011.

32. N. A. Khan, S. Ahmed, Finite difference method with metaheuristic orientation for exploration of time fractional partial differential equations, *Int. J. Appl. Comput. Math.*, **7** (2021), 1–22. doi: 10.1007/s40819-021-01061-y.

33. A. Ahmadian, S. Salahshour, M. Ali-Akbari, F. Ismail, D. Baleanu, A novel approach to approximate fractional derivative with uncertain conditions, *Chaos Soliton. Fract.*, **104** (2017), 68–76. doi: 10.1016/j.chaos.2017.07.026.

34. F. M. Salama, N. H. M. Ali, N. N. Abd Hamid, Fast O(N) hybrid Laplace transform-finite difference method in solving 2D time fractional diffusion equation, *J. Math. Comput. Sci.*, **23** (2021), 110–123. doi: 10.22436/jmcs.023.02.04.

35. N. H. M. Ali, L. M. Kew, New explicit group iterative methods in the solution of two dimensional hyperbolic equations, *J. Comput. Phys.*, **231** (2012), 6953–6968. doi: 10.1016/j.jcp.2012.06.025.

36. N. H. Mohd Ali, A. Mohammed Saeed, Convergence analysis of the preconditioned group splitting methods in boundary value problems, *Abstr. Appl. Anal.*, **2012** (2012), 867598. doi: 10.1155/2012/867598.

37. L. M. Kew, N. H. M. Ali, New explicit group iterative methods in the solution of three dimensional hyperbolic telegraph equations, *J. Comput. Phys.*, **294** (2015), 382–404. doi: 10.1016/j.jcp.2015.03.052.

38. A. Saudi, J. Sulaiman, Robot path planning using four point-explicit group via nine-point laplacian (4EG9L) iterative method, *Procedia Engineering*, **41** (2012), 182–188. doi: 10.1016/j.proeng.2012.07.160.

39. N. H. M. Ali, A. M. Saeed, Preconditioned modified explicit decoupled group for the solution of steady state navier-stokes equation, *Appl. Math. Inform. Sci.*, **7** (2013), 1837–1844. doi: 10.12785/amis/070522.

40. M. A. Khan, N. H. M. Ali, N. N. Abd Hamid, A new fourth-order explicit 270 group method in the solution of two-dimensional fractional rayleigh–stokes problem for a heated generalized second-grade fluid, *Adv. Diff. Equ.*, **2020** (2020), 598. doi: 10.1186/s13662-020-03061-6.

41. N. Abdi, H. Aminikhah, A. H. Sheikhani, J. Alavi, M. Taghipour, An efficient explicit decoupled group method for solving two–dimensional fractional Burgers' equation and its convergence analysis, *Adv. Math. Phys.*, **2021** (2021), 6669287. doi: 10.1155/2021/6669287.

42. N. Abdi, H. Aminikhah, A. R. Sheikhani, High-order rotated grid point iterative method for solving 2D time fractional telegraph equation and its convergence analysis, *Comp. Appl. Math.*, **40** (2021), 54. doi: 10.1007/s40314-021-01451-4.

43. F. M. Salama, N. H. M. Ali, N. N. Abd Hamid, Efficient hybrid group iterative methods in the solution of two-dimensional time fractional cable equation, *Adv. Differ. Equ.*, **2020** (2020), 257. doi: 10.1186/s13662-020-02717-7.

44. N. Moraca, Bounds for norms of the matrix inverse and the smallest singular value, *Linear Algebra Appl.*, **429** (2008), 2589–2601. doi: 10.1016/j.laa.2007.12.026.

45. A. Ali, N. H. M. Ali, On skewed grid point iterative method for solving 2d hyperbolic telegraph fractional differential equation, *Adv. Differ. Equ.*, **2019** (2019), 303. doi: 10.1186/s13662-019-2238-6.

AIMS Press

© 2022 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0)