AIMS Mathematics, 7(9): 15854–15880.
DOI: 10.3934/math.2022868
Received: 15 April 2022
Revised: 11 June 2022
Accepted: 19 June 2022
Published: 27 June 2022

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

Research article

Fast hybrid explicit group methods for solving 2D fractional advection-diffusion equation

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

[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 recent years, fractional partial differential equations (FPDEs) have been viewed as powerful mathematical tools for describing ample phenomena in various scientific disciplines and have been extensively researched. In this article, the hybrid explicit group (HEG) method and the modified hybrid explicit group (MHEG) method are proposed to solve the 2D advection-diffusion problem involving fractional-order derivative of Caputo-type in the temporal direction. The considered problem models transport processes occurring in real-world complex systems. The hybrid grouping methods are developed based upon a Laplace transformation technique with a pair of explicit group finite difference approximations constructed on different grid spacings. The proposed methods are beneficial in reducing the computational burden resulting from the nonlocality of fractional-order differential operator. The theoretical investigation of stability and convergence properties is conducted by utilizing the matrix norm analysis. The improved performance of the proposed methods against a recent competitive method in terms of central processing unit (CPU) time, iterations number and computational cost is illustrated by several numerical experiments.

Keywords: Caputo-type fractional derivative; fractional advection-diffusion problem; finite difference method; Laplace transform; explicit group methods; stability and convergence
Mathematics Subject Classification: 35XX, 65N12

1. Introduction

In the past few years, fractional calculus (FC) has acquired utmost importance as one of the most hot topics of scientific research, and its applications have been widely observed in physics, chemistry, bioscience, signal processing, financial markets, continuum mechanics, control theory, chaotic

systems, rheology, electrical engineering and so forth.

As a matter of fact, numerous researchers from various scientific and engineering fields have recently focused their efforts on the theory and applications of FC. Among the most important features of fractional-order derivatives are the universal and historical dependence properties, which make them attractive for modelling a wide range of complex real-world phenomena. For instance, Javaid et al. [1] established a mathematical model under the Riemann-Liouville fractional derivative to describe the mechanical behavior of a Burgers fluid through rotating cyllinders. Anwar et al. [2] used the Caputo-Fabrizio fractional approach to investigate the thermal properties of the unsteady magnetohydrodynamic flow of an Oldroyd-B fluid. Xu et al. [3] proposed a financial crisis contagion model based on the Caputo-type fractional derivative to analyze the Hopf bifurcation dynamical phenomenon. Ahmed et al. [4] presented a new fractional order Darwinian particle swarm optimization defined in the Grunwald-Letnikov sense. The authors showed that the fractional order Darwinian particle swarm optimization surpasses the traditional particle swarm optimization technique because it estimates the electrical parameters of photovoltaic cells precisely. In addition, many fractional-order epidemiological models analyzing various pandemics and health issues such as COVID-19 outbreak [5, 6], Ebola virus [7], dengue fever [8], and childhood infections [9] can be found in the literature.

FPDEs have been a subject of significant importance as a tool of FC. FPDEs are considered as mathematical extensions of integer-order partial differential equations and have drawn the attention of many researchers in recent years. The integer-order time derivative in the classical advection-diffusion model can be replaced with a fractional-order derivative which leads to the so-called time-fractional advection-diffusion equation (TFADE). The aforementioned equation has the ability to well describe transport processes occurring in complex systems and controlled by anomalous diffusion, which justify its successful usage for modeling various phenomena including heat transfer processes, describing biological systems, air pollution and others [10, 11].

In this work, we point our attention to the TFADE in two dimensional space with the following general form:

$$\begin{cases} {}_0^C D_t^\gamma u(x,y,t) = k_1 \dfrac{\partial^2 u(x,y,t)}{\partial x^2} + k_2 \dfrac{\partial^2 u(x,y,t)}{\partial y^2} - v_1 \dfrac{\partial u(x,y,t)}{\partial x} \\ \qquad\qquad - v_2 \dfrac{\partial u(x,y,t)}{\partial y} + f(x,y,t),\ (x,y) \in \Omega \subset \mathbb{R}^2, \\ u(x,y,0) = g(x,y),\ (x,y) \in \Omega, \\ u(x,y,t) = h(x,y,t),\ (x,y) \in \partial\Omega,\ 0 < t \le T, \end{cases} \tag{1.1}$$

where k_1, k_2 are the diffusion coefficients while v_1 and v_2 are the advection coefficients. $f(x,y,t)$, $g(x,y)$ and $h(x,y,t)$ are given functions. Ω and $\partial\Omega$ denote bounded space domain and its boundary, respectively. ${}_0^C D_t^\gamma u(x,y,t)$, $0 < \gamma \le 1$ represents the fractional-order derivative of Caputo-type which is given by

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

The advection-diffusion model is utilized to illustrate the description of several quantities such as heat, mass and energy, which makes it an applicable tool for modeling various types of real-life

phenomena. The list of applications includes pollutant transport in rivers and atmosphere, water transport in soils, dispersion of disseminated salt and materials in underground water and estuaries, heat transfer processes and fluid flow phenomena, see [12] and the other cited references. The fluid flow phenomenon is one of the most attractive areas of research due to its wide spectrum of applications in different sciences such as physics, biology, medicine and engineering. A substantial class of fluids known as nanofluids was introduced for the first time in 1995 by Choi and Eastman [13]. The nanofluid was obtained by disseminating ultra-fine nanoparticles in a conventional fluid such as oil and water to enhance the thermos-physical characteristics of fluids. Later, an improved version of nanofluid called hybrid nanofluid was introduced by adding two distinct nanoparticles that have higher thermal conductivity to continuous phase liquid. In recent years, the applications of nanofluids/hybrid nanofluids have witnessed exponential growth in various disciplines that range from pharmaceutical processes, microelectronics, fuel cells and hybrid-powered engines to engine cooling. As a result, many researchers have constructed differential equations-based mathematical models to account for the flow nature of nanofluids, see [14–19]. FC is a generalization of classical calculus where the orders of differential and integral operators are extended from the set of integer numbers to the set of real and even complex numbers. In the last few years, the utilization of FPDEs for modeling numerous problems has thrived in diverse fields such as viscoelastic materials, economic processes, control problems, biological systems and image processing [20]. Therefore, fractional modeling is of utmost importance, particularly for problems in which memoy has a significant role. Fractional advection-diffusion equations have been successfully applied for modelling problems in hydrology, chemistry, entropy and engineering [21]. Hence, it is worthwhile to solve fractional advection-diffusion problems. Since equations that contain fractional derivatives are mostly difficult to handle analytically, numerical techniques have been widely employed to solve the advection-diffusion problems of fractional orders. For more details, the interested reader may refer to [10, 22–32].

Numerical schemes based on the discretization of differential operators with fractional and integer orders are the vast majority in literature, see [33] and the references therein. These numerical solution algorithms such as finite difference discretizations implemented for time FPDEs will result in large and sparse linear systems to be solved. In such a case, iterative solvers are more efficient in solving these linear systems due to the sparsity of the coefficients matrix. However, numerical simulations of fractional-order differential models introduce considerable challenges because of the non-locality of the time evolution process. For instance, numerical schemes with iterative solvers for time FPDEs require total computational cost of $O(N^2)$ and storage requirement of $O(NM_s)$, where N and M_s are the temporal steps and spatial grid points, respectively. This ultimately leads to time and memory-intensive simulations, particularly for high-dimensional fractional problems [34–37]. Therefore, the development of fast and accurate numerical methods becomes more and more essential. As a matter of fact, several techniques including parallel computing [38], preconditioning technology [39] and short memory principle [40] have been suggested to improve the computational efficiency of numerical methods in solving FPDEs. This shows that the constant development of numerical algorithms with computational merits is of great importance in the literature. Recently in [41], a Laplace transformation approach is utilized for approximating the fractional-order derivative of Caputo-type and converting the TFADE (1.1) into its corresponding partial differential equation (PDE), and then an implicit difference scheme with an iterative solver is proposed to solve the obtained PDE with less computing effort. In

details, the authors proved that their method, namely hybrid standard point (HSP) iterative method has a linear computational workload of $O(N)$ and an effective storage of $O(M_S)$. This study aims to introduce new enhanced numerical iterative methods for solving Eq (1.1) that are superior in terms of computational efficiency to the HSP method presented in [41].

In recent years, explicit group difference schemes in conjunction with iterative solvers have caught the attention of researchers for solving several types of PDEs that are used to describe various phenomena in the fields of science and engineering [42–46]. The ability of grouping methods to generate fast simulations with low computing effort makes them a superb choice for solving FPDEs [47–51].

In this article, we propose fast and accurate hybrid group iterative methods for handling the TFADE of Dirichlet-type boundary conditions (1.1). The developed methods are based upon the Laplace transform technique presented in [52] and a pair of explicit group finite difference approximations constructed on different grid spacings. The main merits of our methods include that they require less iteration count, computational workload as well as CPU timing in comparison to the HSP iterative methods suggested in [41]. Moreover, our methods can be implemented on parallel computers. It is well understood that the non locality of fractional-order differential operator is inherited by its discretiztions. Therefore, many researchers strive for the design of fast and accurate numerical algorithms to treat the computational complexity in solving FPDEs, which assert the significance of these formulations in the literature, see [53–56]. In this line of thought, we propose our work.

In the literature, several techniques such as the Fourier transform method and the exponential-sum approximation technique can be utilized to reduce the computational complexity of solving time-fractional problems from $O(N^2)$ of direct methods to $O(N\log(N))$ and $O(N\log^2(N))$, respectively [57–59]. In other words, the techniques described here can reduce the computational workload from a quadratic expression to an almost linear expression with respect to N. Salama and Ali [41] developed a fast hybrid method with linear computational complexity of $O(N)$ for solving the TFADE. The main goal of this paper is to construct new numerical algorithms for solving the TFADE that outperform the HSP method [41] in aspects of CPU timing, iteration count, and computational workload. To the best of our knowledge, the development of efficient numerical schemes which are faster than $O(N)$ methods is quite scarce in the literature, and herein lies the motivation of this work.

To summarize, we study the TFADE in two space dimensions (1.1) to account for its numerical solutions. For the purpose of less computational complexity, we develop two new hybrid group iterative methods to solve the mentioned equation. In addition, we provide a detailed and rigorous analysis to demonstrate the unconditional stability and convergence properties with regard to arbitrary step sizes. Furthermore, numerical simulations are implemented to highlight the applicability and efficiency of the established solution algorithms. Overall, we show that our methods provide a reliable and efficient tool for simulating the TFADE. As far as we know, there are no such similar works for the problem (1.1) in literature.

The content of the article is outlined as follows. In the next section, the existing HSP method [41] for the numerical solution of Eq (1.1) is briefly described. Section 3 thoroughly explains the construction of the proposed numerical methods. In Section 4, the theoretical aspects of the unconditional stability together with the convergence properties are investigated via the technique of matrix norm. In Section 5, numerical experiments are conducted to report on the applicability,

accuracy and efficiency of the proposed methods. Finally, the current work is briefly concluded in Section 6.

2. Existing HSP method

In this section, the HSP method presented in [41] is briefly described. Because fractional differential operators are non-local, numerical solutions of all historical time steps must be stored in order to simulate the physical problem at the current time step. This results in numerical simulations with enormous computational costs and storage requirements. To optimize the computational complexity, the Laplace transformation technique proposed in [52] was utilized to approximate the fractional differential operator of Caputo-type and reduce the TFADE (1.1) into its corresponding PDE. Afterwards, the approximating PDE can be solved with less effort to generate numerical solutions that are close to the exact solutions of the TFADE.

The Laplace transform of the Caputo time-fractional derivative is given as [52]

$$\begin{aligned} L\{{}_0^C D_t^\alpha u(x,y,t)\} &= s^\alpha u(x,y,s) - s^{\alpha-1}u(x,y,0) \\ &= s^\alpha[u(x,y,s) - s^{-1}u(x,y,0)], \end{aligned} \tag{2.1}$$

where $u(x, y, s)$ is the Laplace transform of $u(x, y, t)$. In this work, we have $0 < \alpha < 1$, so the term s^α is linearized as [52]

$$s^\alpha \approx \alpha s^1 + (1-\alpha)s^0 = \alpha s + (1-\alpha). \tag{2.2}$$

Substituting (2.2) into (2.1), we get

$$\begin{aligned} L\{{}_0^C D_t^\alpha u(x,y,t)\} &\approx [\alpha s + (1-\alpha)][u(x,y,s) - s^{-1}u(x,y,0)] \\ &= \alpha s[u(x,y,s) - s^{-1}u(x,y,0)] + (1-\alpha)[u(x,y,s) - s^{-1}u(x,y,0)]. \end{aligned} \tag{2.3}$$

By applying the inverse Laplace transform, the follwing expression is obtained

$${}_0^C D_t^\alpha u(x,y,t) \approx \alpha\frac{\partial u(x,y,t)}{\partial t} + (1-\alpha)[u(x,y,t) - u(x,y,0)]. \tag{2.4}$$

By considering Eq (2.4), the TFADE (1.1) is approximated by the following PDE

$$\begin{cases} \dfrac{\partial u(x,y,t)}{\partial t} = K_1\dfrac{\partial^2 u(x,y,t)}{\partial x^2} + K_2\dfrac{\partial^2 u(x,y,t)}{\partial y^2} - V_1\dfrac{\partial u(x,y,t)}{\partial x} - V_2\dfrac{\partial u(x,y,t)}{\partial y} \\ \qquad\qquad - (\mathfrak{L}-1)u(x,y,t) + (\mathfrak{L}-1)g(x,y) + \mathfrak{L}f(x,y,t),\ (x,y) \in \Omega \subset \mathbb{R}^2, \\ u(x,y,0) = g(x,y),\ (x,y) \in \Omega, \\ u(x,y,t) = h(x,y,t),\ (x,y) \in \partial\Omega,\ 0 < t \le T, \end{cases} \tag{2.5}$$

where $K_1 = \frac{k_1}{\gamma}$, $K_2 = \frac{k_2}{\gamma}$, $V_1 = \frac{v_1}{\gamma}$, $V_2 = \frac{V_2}{\gamma}$, $\mathfrak{L} = \frac{1}{\gamma}$ and $\Omega = [0, L] \times [0, L]$. For the discretiztion of the above equation, we define $\tau = \frac{T}{N}$ as the time increment and $h = \frac{L}{M}$ as the space step size in both x and y coordinates, where N and M are positive integers. Then we define uniform space and time partitions $x_i = ih$, $y_j = jh$, $0 \le i, j \le M$ and $t^n = n\tau$, $0 \le n \le N$. Let $u_{i,j}^n$ be the numerical solution at the

mesh point (x_i, y_j, t^n). It is illustrated in [41] that a backward and central differences in time and space, respectively, give the rise of the following implicit difference scheme for the PDE (2.5):

$$\begin{aligned} u_{i,j}^{n+1} = \frac{1}{(1+(\mathfrak{L}-1)\tau+2\mathfrak{M}_1+2\mathfrak{M}_2)}\Big[&(\mathfrak{M}_1-\frac{\mathfrak{N}_1}{2})u_{i+1,j}^{n+1}+(\mathfrak{M}_1+\frac{\mathfrak{N}_1}{2})u_{i-1,j}^{n+1} \\ &+(\mathfrak{M}_2-\frac{\mathfrak{N}_2}{2})u_{i,j+1}^{n+1}+(\mathfrak{M}_2+\frac{\mathfrak{N}_2}{2})u_{i,j-1}^{n+1}+u_{i,j}^{n}+(\mathfrak{L}-1)\tau u_{i,j}^{0}+\mathfrak{L}\tau f_{i,j}^{n+1}\Big], \end{aligned} \tag{2.6}$$

where $\mathfrak{M}_1 = \frac{K_1\tau}{h^2}$, $\mathfrak{M}_2 = \frac{K_2\tau}{h^2}$, $\mathfrak{N}_1 = \frac{V_1\tau}{h}$ and $\mathfrak{N}_2 = \frac{V_2\tau}{h}$. This scheme has been proven to be stable without restriction conditions and accurate with convergence order of $O(\tau + h^2)$. By using an iterative solver for the difference scheme (2.6), the HSP iterative method proceeds by generating iterations at each time step on all mesh points utilizing Eq (2.6) until predetermined convergence criteria are met, before moving to the following time step. The iteration process carries on as far as the targeted time level is not reached.

The aforementioned solution algorithm is beneficial in producing accurate numerical solutions for Eq (1.1) while reducing the computational effort significantly when compared to the conventional finite difference schemes. For more details, we can see [41]. In order to further improve the computational efficiency, the next section reports on the proposed HEG and MHEG methods.

3. Construction of the hybrid group methods

3.1. The four-point hybrid explicit group (HEG) method

Consider the approximation formula (2.6) demonstrated in the previous section. Figure 1 highlights the computational molecule of the HEG method. It can be seen that the mesh points are arranged into four-point blocks to facilitate the formulation of the HEG method. The Eq (2.6) can be applied to any four-point block depicted in Figure 1 which leads to a (4×4) system written as,

$$\begin{pmatrix} q_1 & -q_2 & 0 & -q_4 \\ -q_3 & q_1 & -q_4 & 0 \\ 0 & -q_5 & q_1 & -q_3 \\ -q_5 & 0 & -q_2 & q_1 \end{pmatrix} \begin{pmatrix} u_{i,j}^{n+1} \\ u_{i+1,j}^{n+1} \\ u_{i+1,j+1}^{n+1} \\ u_{i,j+1}^{n+1} \end{pmatrix} = \begin{pmatrix} \mathfrak{R}_{i,j} \\ \mathfrak{R}_{i+1,j} \\ \mathfrak{R}_{i+1,j+1} \\ \mathfrak{R}_{i,j+1} \end{pmatrix}, \tag{3.1}$$

where

$$\begin{aligned} q_1 &= (1+(\mathfrak{L}-1)\tau+2\mathfrak{M}_1+2\mathfrak{M}_2), \qquad q_2 = (\mathfrak{M}_1-\frac{\mathfrak{N}_1}{2}), \\ q_3 &= (\mathfrak{M}_1+\frac{\mathfrak{N}_1}{2}), \qquad q_4 = (\mathfrak{M}_2-\frac{\mathfrak{N}_2}{2}), \qquad q_5 = (\mathfrak{M}_2+\frac{\mathfrak{N}_2}{2}), \\ \mathfrak{R}_{i,j} &= q_3u_{i-1,j}^{n+1}+q_5u_{i,j-1}^{n+1}+u_{i,j}^{n}+(\mathfrak{L}-1)\tau u_{i,j}^{0}+\mathfrak{L}\tau f_{i,j}^{n+1}, \\ \mathfrak{R}_{i+1,j} &= q_2u_{i+2,j}^{n+1}+q_5u_{i+1,j-1}^{n+1}+u_{i+1,j}^{n}+(\mathfrak{L}-1)\tau u_{i+1,j}^{0}+\mathfrak{L}\tau f_{i+1,j}^{n+1}, \\ \mathfrak{R}_{i+1,j+1} &= q_2u_{i+2,j+1}^{n+1}+q_4u_{i+1,j+2}^{n+1}+u_{i+1,j+1}^{n}+(\mathfrak{L}-1)\tau u_{i+1,j+1}^{0}+\mathfrak{L}\tau f_{i+1,j+1}^{n+1}, \\ \mathfrak{R}_{i,j+1} &= q_3u_{i-1,j+1}^{n+1}+q_4u_{i,j+2}^{n+1}+u_{i,j+1}^{n}+(\mathfrak{L}-1)\tau u_{i,j+1}^{0}+\mathfrak{L}\tau f_{i,j+1}^{n+1}. \end{aligned}$$

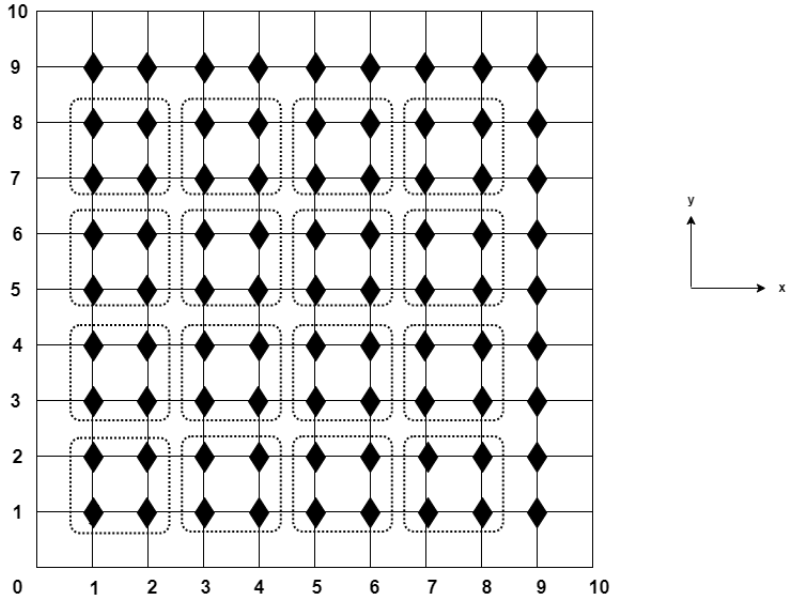


Figure 1. Mesh points included in the 2D domain of the HEG method with $M = 10$.

The last system can be rewritten in the following form,

$$\begin{pmatrix} u_{i,j}^{n+1} \\ u_{i+1,j}^{n+1} \\ u_{i+1,j+1}^{n+1} \\ u_{i,j+1}^{n+1} \end{pmatrix} = \frac{1}{\mathcal{I}} \begin{pmatrix} \mathcal{I}_1 & \mathcal{I}_2 & \mathcal{I}_3 & \mathcal{I}_4 \\ \mathcal{I}_5 & \mathcal{I}_1 & \mathcal{I}_4 & \mathcal{I}_6 \\ \mathcal{I}_7 & \mathcal{I}_8 & \mathcal{I}_1 & \mathcal{I}_5 \\ \mathcal{I}_8 & \mathcal{I}_9 & \mathcal{I}_2 & \mathcal{I}_1 \end{pmatrix} \begin{pmatrix} \Re_{i,j} \\ \Re_{i+1,j} \\ \Re_{i+1,j+1} \\ \Re_{i,j+1} \end{pmatrix}, \tag{3.2}$$

where

$$\begin{aligned}
\mathcal{I} =&(1 + (\mathfrak{L} - 1)\tau + 2\mathfrak{M}_1 + 2\mathfrak{M}_2)^4 - 2(1 + (\mathfrak{L} - 1)\tau + 2\mathfrak{M}_1 + 2\mathfrak{M}_2)^2(\mathfrak{M}_1 - \frac{\mathfrak{N}_1}{2})(\mathfrak{M}_1 + \frac{\mathfrak{N}_1}{2}) \\
&- 2(1 + (\mathfrak{L} - 1)\tau + 2\mathfrak{M}_1 + 2\mathfrak{M}_2)^2(\mathfrak{M}_2 - \frac{\mathfrak{N}_2}{2})(\mathfrak{M}_2 + \frac{\mathfrak{N}_2}{2}) + (\mathfrak{M}_1 - \frac{\mathfrak{N}_1}{2})^2(\mathfrak{M}_1 + \frac{\mathfrak{N}_1}{2})^2 \\
&- 2(\mathfrak{M}_1 - \frac{\mathfrak{N}_1}{2})(\mathfrak{M}_1 + \frac{\mathfrak{N}_1}{2})(\mathfrak{M}_2 - \frac{\mathfrak{N}_2}{2})(\mathfrak{M}_2 + \frac{\mathfrak{N}_2}{2}) + (\mathfrak{M}_2 - \frac{\mathfrak{N}_2}{2})^2(\mathfrak{M}_2 + \frac{\mathfrak{N}_2}{2})^2, \\
\mathcal{I}_1 =&\frac{1}{4}(1 + (\mathfrak{L} - 1)\tau + 2\mathfrak{M}_1 + 2\mathfrak{M}_2)(\mathfrak{N}_1^2 + \mathfrak{N}_2^2 + 12\mathfrak{M}_1^2 + 12\mathfrak{M}_2^2 + 32\mathfrak{M}_1\mathfrak{M}_2 + 16\mathfrak{M}_1 + 16\mathfrak{M}_2 \\
&+ 16(\mathfrak{L} - 1)\tau\mathfrak{M}_1 + 16(\mathfrak{L} - 1)\tau\mathfrak{M}_2 + 4(\mathfrak{L} - 1)^2\tau^2 + 8(\mathfrak{L} - 1)\tau + 4), \\
\mathcal{I}_2 =&\frac{-1}{8}(\mathfrak{N}_1 - 2\mathfrak{M}_1)(\mathfrak{N}_1^2 - \mathfrak{N}_2^2 + 12\mathfrak{M}_1^2 + 20\mathfrak{M}_2^2 + 32\mathfrak{M}_1\mathfrak{M}_2 + 16\mathfrak{M}_1 + 16\mathfrak{M}_2 + 16(\mathfrak{L} - 1)\tau\mathfrak{M}_1 \\
&+ 16(\mathfrak{L} - 1)\tau\mathfrak{M}_2 + 4(\mathfrak{L} - 1)^2\tau^2 + 8(\mathfrak{L} - 1)\tau + 4), \\
\mathcal{I}_3 =&\frac{1}{2}(1 + (\mathfrak{L} - 1)\tau + 2\mathfrak{M}_1 + 2\mathfrak{M}_2)(\mathfrak{N}_1 - 2\mathfrak{M}_1)(\mathfrak{N}_2 - 2\mathfrak{M}_2), \\
\mathcal{I}_4 =&\frac{1}{8}(\mathfrak{N}_2 - 2\mathfrak{M}_2)(\mathfrak{N}_1^2 - \mathfrak{N}_2^2 - 20\mathfrak{M}_1^2 - 12\mathfrak{M}_2^2 - 32\mathfrak{M}_1\mathfrak{M}_2 - 16\mathfrak{M}_1 - 16\mathfrak{M}_2 - 16(\mathfrak{L} - 1)\tau\mathfrak{M}_1 \\
&- 16(\mathfrak{L} - 1)\tau\mathfrak{M}_2 - 4(\mathfrak{L} - 1)^2\tau^2 - 8(\mathfrak{L} - 1)\tau - 4), \\
\mathcal{I}_5 =&\frac{1}{8}(\mathfrak{N}_2 + 2\mathfrak{M}_2)(\mathfrak{N}_1^2 - \mathfrak{N}_2^2 + 12\mathfrak{M}_1^2 + 20\mathfrak{M}_2^2 + 32\mathfrak{M}_1\mathfrak{M}_2 + 16\mathfrak{M}_1 + 16\mathfrak{M}_2 + 16(\mathfrak{L} - 1)\tau\mathfrak{M}_1
\end{aligned}$$

$$+ 16(\mathfrak{L}-1)\tau\mathfrak{M}_2 + 4(\mathfrak{L}-1)^2\tau^2 + 8(\mathfrak{L}-1)\tau + 4),$$

$$\mathcal{I}_6 = \frac{-1}{2}(1 + (\mathfrak{L}-1)\tau + 2\mathfrak{M}_1 + 2\mathfrak{M}_2)(\mathfrak{N}_1 + 2\mathfrak{M}_2)(\mathfrak{N}_2 - 2\mathfrak{M}_2),$$

$$\mathcal{I}_7 = \frac{1}{2}(1 + (\mathfrak{L}-1)\tau + 2\mathfrak{M}_1 + 2\mathfrak{M}_2)(\mathfrak{N}_1 + 2\mathfrak{M}_1)(\mathfrak{N}_2 + 2\mathfrak{M}_2),$$

$$\mathcal{I}_8 = \frac{-1}{8}(\mathfrak{N}_2 + 2\mathfrak{M}_2)(\mathfrak{N}_1^2 - \mathfrak{N}_2^2 - 20\mathfrak{M}_1^2 - 12\mathfrak{M}_2^2 - 32\mathfrak{M}_1\mathfrak{M}_2 - 16\mathfrak{M}_1 - 16\mathfrak{M}_2 - 16(\mathfrak{L}-1)\tau\mathfrak{M}_1$$
$$- 16(\mathfrak{L}-1)\tau\mathfrak{M}_2 - 4(\mathfrak{L}-1)^2\tau^2 - 8(\mathfrak{L}-1)\tau - 4),$$

$$\mathcal{I}_9 = \frac{-1}{2}(1 + (\mathfrak{L}-1)\tau + 2\mathfrak{M}_1 + 2\mathfrak{M}_2)(\mathfrak{N}_1 - 2\mathfrak{M}_1)(\mathfrak{N}_2 + 2\mathfrak{M}_2).$$

Assuming M is even, the application of the HEG method entails the iterative evaluation of solutions at any time level on both grouped and non-grouped mesh points. More specifically, the solutions on each group of four points are iterated by utilizing Eq (3.2) while the solutions on the residual non-grouped mesh points are iterated with the help of Eq (2.6) until a predetermined convergence test is achieved. Thereafter, the attained solutions are used as an initial approximate in order to initiate the iterative process at the following time level. As far as the targeted time step is not reached, the iteration process is repeated.

3.2. The four-point modified hybrid explicit group (MHEG) method

For the formulation of the MHEG method, a new implicit difference approximation based on a uniform mesh with spatial step size $2h = \frac{2L}{M}$ is derived. Applying backward difference in time and central difference approximations for the remaining derivatives in Eq (2.5), the $2h$-spaced fully discrete scheme can be formulated as written as,

$$\begin{aligned}\frac{U_{i,j}^{n+1} - U_{i,j}^{n}}{\tau} &= K_1\left(\frac{U_{i+2,j}^{n+1} - 2U_{i,j}^{n+1} + U_{i-2,j}^{n+1}}{h^2}\right) + K_2\left(\frac{U_{i,j+2}^{n+1} - 2U_{i,j}^{n+1} + U_{i,j-2}^{n+1}}{h^2}\right)\\ &\quad - V_1\left(\frac{U_{i+2,j}^{n+1} - U_{i-2,j}^{n+1}}{2h}\right) - V_2\left(\frac{U_{i,j+2}^{n+1} - U_{i,j-2}^{n+1}}{2h}\right) - (\mathfrak{L}-1)U_{i,j}^{n+1}\\ &\quad + (\mathfrak{L}-1)U_{i,j}^{0} + rf_{i,j}^{n+1} + O(\tau + h^2),\end{aligned} \tag{3.3}$$

where $U_{i,j}^{k}$ is the exact solution of Eq (2.5) at the location point (i, j, k). Neglecting the remainder $O(\tau+h^2)$ in (3.3) and replacing $U_{i,j}^{n}$ by the relevant numerical approximation $u_{i,j}^{n}$, the following implicit difference scheme with $2h$ spacing is obtained:

$$\begin{aligned}u_{i,j}^{n+1} = \frac{1}{(1 + (\mathfrak{L}-1)\tau + \mathfrak{M}_1/2 + \mathfrak{M}_2/2)}\Big[&(\frac{\mathfrak{M}_1}{4} - \frac{\mathfrak{N}_1}{4})u_{i+2,j}^{n+1} + (\frac{\mathfrak{M}_1}{4} + \frac{\mathfrak{N}_1}{4})u_{i-2,j}^{n+1}\\ &+ (\frac{\mathfrak{M}_2}{4} - \frac{\mathfrak{N}_2}{4})u_{i,j+2}^{n+1} + (\frac{\mathfrak{M}_2}{4} + \frac{\mathfrak{N}_2}{4})u_{i,j-2}^{n+1} + u_{i,j}^{n} + (\mathfrak{L}-1)\tau u_{i,j}^{0} + \mathfrak{L}\tau f_{i,j}^{n+1}\Big].\end{aligned} \tag{3.4}$$

The MHEG method is formulated by applying Eq (3.4) to each four-point group illustrated in Figure 2 which results in the followinf system represented in matrix form,

$$\begin{pmatrix} p_1 & -p_2 & 0 & -p_4 \\ -p_3 & p_1 & -p_4 & 0 \\ 0 & -p_5 & p_1 & -p_3 \\ -p_5 & 0 & -p_2 & p_1 \end{pmatrix} \begin{pmatrix} u_{i,j}^{n+1} \\ u_{i+2,j}^{n+1} \\ u_{i+2,j+2}^{n+1} \\ u_{i,j+2}^{n+1} \end{pmatrix} = \begin{pmatrix} \mathfrak{R}_{i,j}^{*} \\ \mathfrak{R}_{i+2,j}^{*} \\ \mathfrak{R}_{i+2,j+2}^{*} \\ \mathfrak{R}_{i,j+2}^{*} \end{pmatrix}, \tag{3.5}$$

where

$$\begin{aligned} p_1 &= (1 + (\mathfrak{L} - 1)\tau + \mathfrak{M}_1/2 + \mathfrak{M}_2/2), \qquad p_2 = (\frac{\mathfrak{M}_1}{4} - \frac{\mathfrak{N}_1}{4}), \\ p_3 &= (\frac{\mathfrak{M}_1}{4} + \frac{\mathfrak{N}_1}{4}), \qquad p_4 = (\frac{\mathfrak{M}_2}{4} - \frac{\mathfrak{N}_2}{4}), \qquad p_5 = (\frac{\mathfrak{M}_2}{4} + \frac{\mathfrak{N}_2}{4}), \\ \mathfrak{R}_{i,j}^{*} &= p_3 u_{i-2,j}^{n+1} + p_5 u_{i,j-2}^{n+1} + u_{i,j}^{n} + (\mathfrak{L} - 1)\tau u_{i,j}^{0} + \mathfrak{L}\tau f_{i,j}^{n+1}, \\ \mathfrak{R}_{i+2,j}^{*} &= p_2 u_{i+4,j}^{n+1} + p_5 u_{i+2,j-2}^{n+1} + u_{i+2,j}^{n} + (\mathfrak{L} - 1)\tau u_{i+2,j}^{0} + \mathfrak{L}\tau f_{i+2,j}^{n+1}, \\ \mathfrak{R}_{i+2,j+2}^{*} &= p_2 u_{i+4,j+2}^{n+1} + p_4 u_{i+2,j+4}^{n+1} + u_{i+2,j+2}^{n} + (\mathfrak{L} - 1)\tau u_{i+2,j+2}^{0} + \mathfrak{L}\tau f_{i+2,j+2}^{n+1}, \\ \mathfrak{R}_{i,j+2}^{*} &= p_3 u_{i-2,j+2}^{n+1} + p_4 u_{i,j+4}^{n+1} + u_{i,j+2}^{n} + (\mathfrak{L} - 1)\tau u_{i,j+2}^{0} + \mathfrak{L}\tau f_{i,j+2}^{n+1}. \end{aligned}$$

The last system in Eq (3.5) can be rewritten as,

$$\begin{pmatrix} u_{i,j}^{n+1} \\ u_{i+2,j}^{n+1} \\ u_{i+2,j+2}^{n+1} \\ u_{i,j+2}^{n+1} \end{pmatrix} = \frac{1}{\mathcal{J}} \begin{pmatrix} \mathcal{J}_1 & \mathcal{J}_2 & \mathcal{J}_3 & \mathcal{J}_4 \\ \mathcal{J}_5 & \mathcal{J}_1 & \mathcal{J}_4 & \mathcal{J}_6 \\ \mathcal{J}_7 & \mathcal{J}_8 & \mathcal{J}_1 & \mathcal{J}_5 \\ \mathcal{J}_8 & \mathcal{J}_9 & \mathcal{J}_2 & \mathcal{J}_1 \end{pmatrix} \begin{pmatrix} \mathfrak{R}_{i,j}^{*} \\ \mathfrak{R}_{i+2,j}^{*} \\ \mathfrak{R}_{i+2,j+2}^{*} \\ \mathfrak{R}_{i,j+2}^{*} \end{pmatrix}, \tag{3.6}$$

where

$$\begin{aligned} \mathcal{J} =&(1 + (\mathfrak{L} - 1)\tau + \frac{\mathfrak{M}_1}{2} + \frac{\mathfrak{M}_2}{2})^4 - 2(1 + (\mathfrak{L} - 1)\tau + \frac{\mathfrak{M}_1}{2} + \frac{\mathfrak{M}_2}{2})^2(\frac{\mathfrak{M}_1}{4} - \frac{\mathfrak{N}_1}{4})(\frac{\mathfrak{M}_1}{4} + \frac{\mathfrak{N}_1}{4}) \\ &- 2(1 + (\mathfrak{L} - 1)\tau + \frac{\mathfrak{M}_1}{2} + \frac{\mathfrak{M}_2}{2})^2(\frac{\mathfrak{M}_2}{4} - \frac{\mathfrak{N}_2}{4})(\frac{\mathfrak{M}_2}{4} + \frac{\mathfrak{N}_2}{4}) + (\frac{\mathfrak{M}_1}{4} - \frac{\mathfrak{N}_1}{4})^2(\frac{\mathfrak{M}_1}{4} + \frac{\mathfrak{N}_1}{4})^2 \\ &- 2(\frac{\mathfrak{M}_1}{4} - \frac{\mathfrak{N}_1}{4})(\frac{\mathfrak{M}_1}{4} + \frac{\mathfrak{N}_1}{4})(\frac{\mathfrak{M}_2}{4} - \frac{\mathfrak{N}_2}{4})(\frac{\mathfrak{M}_2}{4} + \frac{\mathfrak{N}_2}{4}) + (\frac{\mathfrak{M}_2}{4} - \frac{\mathfrak{N}_2}{4})^2(\frac{\mathfrak{M}_2}{4} + \frac{\mathfrak{N}_2}{4})^2, \\ \mathcal{J}_1 =&\frac{1}{32}(2 + 2(\mathfrak{L} - 1)\tau + \mathfrak{M}_1 + \mathfrak{M}_2)(\mathfrak{N}_1^2 + \mathfrak{N}_2^2 + 3\mathfrak{M}_1^2 + 3\mathfrak{M}_2^2 + 8\mathfrak{M}_1\mathfrak{M}_2 + 16\mathfrak{M}_1 + 16\mathfrak{M}_2 \\ &+ 16(\mathfrak{L} - 1)\tau\mathfrak{M}_1 + 16(\mathfrak{L} - 1)\tau\mathfrak{M}_2 + 16(\mathfrak{L} - 1)^2\tau^2 + 32(\mathfrak{L} - 1)\tau + 16), \\ \mathcal{J}_2 =&\frac{-1}{64}(\mathfrak{N}_1 - \mathfrak{M}_1)(\mathfrak{N}_1^2 - \mathfrak{N}_2^2 + 3\mathfrak{M}_1^2 + 5\mathfrak{M}_2^2 + 8\mathfrak{M}_1\mathfrak{M}_2 + 16\mathfrak{M}_1 + 16\mathfrak{M}_2 + 16(\mathfrak{L} - 1)\tau\mathfrak{M}_1 \\ &+ 16(\mathfrak{L} - 1)\tau\mathfrak{M}_2 + 16(\mathfrak{L} - 1)^2\tau^2 + 32(\mathfrak{L} - 1)\tau + 16), \\ \mathcal{J}_3 =&\frac{1}{16}(2 + 2(\mathfrak{L} - 1)\tau + \mathfrak{M}_1 + \mathfrak{M}_2)(\mathfrak{N}_1 - \mathfrak{M}_1)(\mathfrak{N}_2 - \mathfrak{M}_2), \\ \mathcal{J}_4 =&\frac{1}{64}(\mathfrak{N}_2 - \mathfrak{M}_2)(\mathfrak{N}_1^2 - \mathfrak{N}_2^2 - 5\mathfrak{M}_1^2 - 3\mathfrak{M}_2^2 - 8\mathfrak{M}_1\mathfrak{M}_2 - 16\mathfrak{M}_1 - 16\mathfrak{M}_2 - 16(\mathfrak{L} - 1)\tau\mathfrak{M}_1 \\ &- 16(\mathfrak{L} - 1)\tau\mathfrak{M}_2 - 16(\mathfrak{L} - 1)^2\tau^2 - 32(\mathfrak{L} - 1)\tau - 16), \end{aligned}$$

$$\mathcal{J}_5 = \frac{1}{64}(\mathfrak{N}_1 + \mathfrak{M}_1)(\mathfrak{N}_1^2 - \mathfrak{N}_2^2 + 3\mathfrak{M}_1^2 + 5\mathfrak{M}_2^2 + 8\mathfrak{M}_1\mathfrak{M}_2 + 16\mathfrak{M}_1 + 16\mathfrak{M}_2 + 16(\mathfrak{L} - 1)\tau\mathfrak{M}_1 + 16(\mathfrak{L} - 1)\tau\mathfrak{M}_2 + 16(\mathfrak{L} - 1)^2\tau^2 + 32(\mathfrak{L} - 1)\tau + 16),$$

$$\mathcal{J}_6 = \frac{-1}{16}(2 + 2(\mathfrak{L} - 1)\tau + \mathfrak{M}_1 + \mathfrak{M}_2)(\mathfrak{N}_1 + \mathfrak{M}_1)(\mathfrak{N}_2 - \mathfrak{M}_2),$$

$$\mathcal{J}_7 = \frac{1}{16}(2 + 2(\mathfrak{L} - 1)\tau + \mathfrak{M}_1 + \mathfrak{M}_2)(\mathfrak{N}_1 + \mathfrak{M}_1)(\mathfrak{N}_2 + \mathfrak{M}_2),$$

$$\mathcal{J}_8 = \frac{-1}{64}(\mathfrak{N}_2 + \mathfrak{M}_2)(\mathfrak{N}_1^2 - \mathfrak{N}_2^2 - 5\mathfrak{M}_1^2 - 3\mathfrak{M}_2^2 - 8\mathfrak{M}_1\mathfrak{M}_2 - 16\mathfrak{M}_1 - 16\mathfrak{M}_2 - 16(\mathfrak{L} - 1)\tau\mathfrak{M}_1 - 16(\mathfrak{L} - 1)\tau\mathfrak{M}_2 - 16(\mathfrak{L} - 1)^2\tau^2 - 32(\mathfrak{L} - 1)\tau - 16),$$

$$\mathcal{J}_9 = \frac{-1}{16}(2 + 2(\mathfrak{L} - 1)\tau + \mathfrak{M}_1 + \mathfrak{M}_2)(\mathfrak{N}_1 - \mathfrak{M}_1)(\mathfrak{N}_2 + \mathfrak{M}_2).$$

Taking note of Figure 2, the mesh nodes that we find the solution values at are formed of three kinds of points, namely diamond-shaped ♦, square-shaped □ and circle-shaped ○ mesh points. One can easily verify that the evaluation of Eq (3.6) can be performed only on ♦ points. Thus, the MHEG method is implemented by generating iterations at each time step on the ♦ points until certain convergence is attained. Thereafter, the solution values on the residual mesh nodes are evaluated directly once by utilizing the standard and rotated point approximation formulas. The MHEG solution algorithm is outlined in Algorithm 1. As regards the implementation of the proposed methods, the application of Eqs (3.2) and (4.3) to the mesh points of the solution domain will result in large and sparse systems of linear equations. All linear systems will be solved by using the Gauss-Seidel iterative scheme.

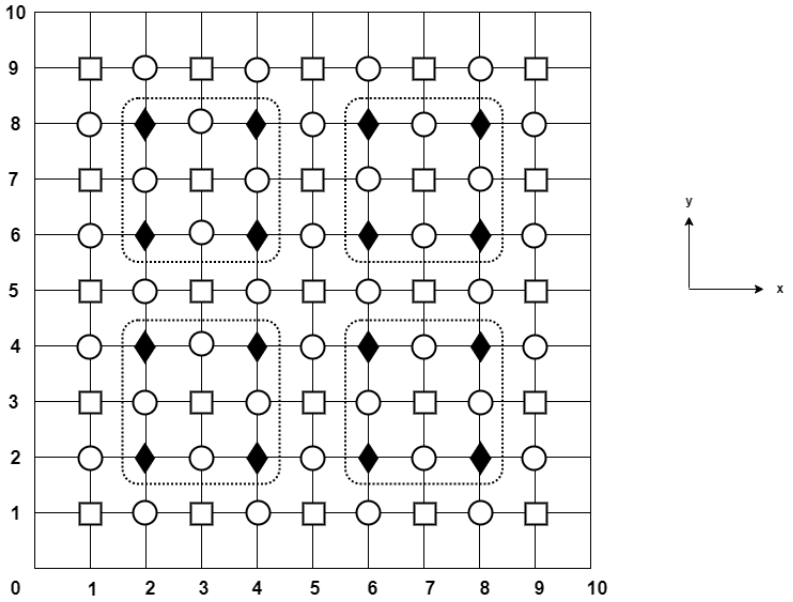


Figure 2. Mesh points included in the 2D domain of the MHEG method with $M = 10$.

Algorithm 1: The four-point MHEG iterative method

(1) Classify the mesh nodes into three different subclasses of ♦, □ and ○ as depicted in Figure 2.
(2) Arrange all the mesh points of type ♦ into four-point groups.
(3) Set an initial guess for the numerical solution at the present time step.
(4) At each group of four points, iterate the intermediate solutions at ♦ points using

$$\begin{pmatrix} \hat{u}_{i,j}^{n+1,k+1} \\ \hat{u}_{i+2,j}^{n+1,k+1} \\ \hat{u}_{i+2,j+2}^{n+1,k+1} \\ \hat{u}_{i,j+2}^{n+1,k+1} \end{pmatrix} = \frac{1}{\mathcal{J}} \begin{pmatrix} \mathcal{J}_1 & \mathcal{J}_2 & \mathcal{J}_3 & \mathcal{J}_4 \\ \mathcal{J}_5 & \mathcal{J}_1 & \mathcal{J}_4 & \mathcal{J}_6 \\ \mathcal{J}_7 & \mathcal{J}_8 & \mathcal{J}_1 & \mathcal{J}_5 \\ \mathcal{J}_8 & \mathcal{J}_9 & \mathcal{J}_2 & \mathcal{J}_1 \end{pmatrix} \begin{pmatrix} \mathfrak{R}_{i,j}^{*} \\ \mathfrak{R}_{i+2,j}^{*} \\ \mathfrak{R}_{i+2,j+2}^{*} \\ \mathfrak{R}_{i,j+2}^{*} \end{pmatrix},$$

where $\mathfrak{R}_{i,j}^{*}, \mathfrak{R}_{i+2,j}^{*}, \mathfrak{R}_{i+2,j+2}^{*} and \mathfrak{R}_{i,j+2}^{*}$ are as specified before. Perform the Gauss-Seidel solver

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

where ω is the relaxation factor and k is the iteration number.
(5) Check the convergence in the previous step. If the computed numerical solutions do converge, go to the next step. Otherwise, go back to step 3.
(6) Compute the remaining solution values directly once with the following order:
   (a) For the □ mesh points, a new discretization scheme based on finite difference approximations for Eq (2.6) is derived on rotated (skewed) mesh. Such rotated or skewed mesh can be established by the 45° clockwise rotation of the $x-y$ axes [60]. Here, the solutions on the □ mesh points are evaluated directly once using the rotated difference scheme given as follows:

$$\begin{aligned} u_{i,j}^{n+1} =& \frac{1}{(1+(\mathfrak{L}-1)\tau+\mathfrak{M}_1+\mathfrak{M}_2)}\Big[(\frac{\mathfrak{M}_1}{2}-\frac{\mathfrak{R}_1}{4}+\frac{\mathfrak{R}_2}{4})u_{i+1,j-1}^{n+1}+(\frac{\mathfrak{M}_1}{2} \\ &+\frac{\mathfrak{R}_1}{4}-\frac{\mathfrak{R}_2}{4})u_{i-1,j+1}^{n+1}+(\frac{\mathfrak{M}_2}{2}-\frac{\mathfrak{R}_1}{4}-\frac{\mathfrak{R}_2}{4})u_{i+1,j+1}^{n+1}+(\frac{\mathfrak{M}_2}{2} \\ &+\frac{\mathfrak{R}_1}{4}+\frac{\mathfrak{R}_2}{4})u_{i-1,j-1}^{n+1}+u_{i,j}^{n}+(\mathfrak{L}-1)\tau u_{i,j}^{0}+\mathfrak{L}\tau f_{i,j}^{n+1}\Big]. \end{aligned} \tag{3.7}$$

   (b) For the residual ○ mesh points, the implicit difference scheme (2.6) is utilized.

4. Stability and convergence analyses

Here, it is worth pointing out that the HEG (3.2) and MHEG (3.6) equations are derived from the same implicit difference scheme but with different spatial step sizes h and $2h$, respectively. Hence, the investigation of stability and convergence properties of the HEG and MHEG schemes could be analyzed in an analogous fashion. Here, special attention is dedicated to study the theoretical analysis of the MHEG scheme (3.6) via the technique of matrix norm. To this end, some useful remarks are introduced as follows:

Remark 4.1 ([61]). *$\mathfrak{A}_{M\times M} = [\mathcal{A}_{i,j}]_{M\times M}$ is a strictly diagonally dominant (SDD) matrix if $|\mathcal{A}_{i,i}| > \mathcal{R}_i(\mathfrak{A})$, $1 \le i \le M$ in which $\mathcal{R}_i(\mathfrak{A})$ is i-th deleted absolute row sum.*

Remark 4.2 ([62]). *If $\mathfrak{A}_{M\times M}$ is a SDD matrix, then it is invertible and the infinity norm of matrix inverse has the following upper bound:*

$$\|\mathfrak{A}^{-1}\|_\infty \le \frac{1}{\min_{1\le i\le M}\left\{|\mathcal{A}_{i,i}| - \mathcal{R}_i(\mathfrak{A})\right\}}.$$

4.1. Stability analysis

This subsection investigates the stability of the MHEG scheme defined in Eq (3.6). To simplify our discussion, we assume that $\mathfrak{M}_1 = \mathfrak{M}_2 = \mathfrak{M} = \frac{\tau}{h^2}$ and $\mathfrak{N}_1 = \mathfrak{N}_2 = \mathfrak{N} = \frac{\tau}{h}$. With this, the matrix representation of Eq (3.6) is given by

$$\mathfrak{A}u^{n+1} = \mathfrak{B}u^n + \mathfrak{C}u^0 + b, \quad 0 \le n \le N-1, \tag{4.1}$$

where u^n is an $\frac{(M-2)^2}{4}$-dimensional block vector represented as

$$u^n = (u_1^n, u_2^n, \ldots, u^n_{\frac{(M-2)^2}{16}})^T,\ u_p^n = (u_{i,j}^n, u_{i+2,j}^n, u_{i+2,j+2}^n, u_{i,j+2}^n)^T,\ 1 \le p \le \frac{(M-2)^2}{16},$$

and

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

$$\mathfrak{C} = \begin{pmatrix} P_1 & & & & \\ & P_1 & & & \\ & & \ddots & & \\ & & & P_1 & \\ & & & & P_1 \end{pmatrix}, b = \begin{pmatrix} D_1 \\ D_1 \\ \vdots \\ D_1 \\ D_1 \end{pmatrix},$$

$$J_1 = \begin{pmatrix} Q_1 & Q_3 & & & \\ Q_2 & Q_1 & Q_3 & & \\ & \ddots & & & \\ & & Q_2 & Q_1 & Q_3 \\ & & & Q_2 & Q_1 \end{pmatrix}, J_2 = \begin{pmatrix} Q_5 & & & & \\ & Q_5 & & & \\ & & \ddots & & \\ & & & Q_5 & \\ & & & & Q_5 \end{pmatrix},$$

$$J_3 = \begin{pmatrix} Q_4 & & & & \\ & Q_4 & & & \\ & & \ddots & & \\ & & & Q_4 & \\ & & & & Q_4 \end{pmatrix}, H_1 = \begin{pmatrix} I_4 & & & & \\ & I_4 & & & \\ & & \ddots & & \\ & & & I_4 & \\ & & & & I_4 \end{pmatrix},$$

$$P_1 = \begin{pmatrix} T_1 & & & & \\ & T_1 & & & \\ & & \ddots & & \\ & & & T_1 & \\ & & & & T_1 \end{pmatrix}, D_1 = \begin{pmatrix} G_1 \\ G_1 \\ \vdots \\ G_1 \\ G_1 \end{pmatrix},$$

$$Q_1 = \begin{pmatrix} n_{11} & -n_{22} & 0 & -n_{22} \\ -n_{33} & n_{11} & -n_{22} & 0 \\ 0 & -n_{33} & n_{11} & -n_{33} \\ -n_{33} & 0 & -n_{22} & n_{11} \end{pmatrix}, Q_2 = \begin{pmatrix} 0 & 0 & 0 & -n_{33} \\ 0 & 0 & -n_{33} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix},$$

$$Q_3 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & -n_{22} & 0 & 0 \\ -n_{22} & 0 & 0 & 0 \end{pmatrix}, Q_4 = \begin{pmatrix} 0 & -n_{33} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & -n_{33} & 0 \end{pmatrix},$$

$$Q_5 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ -n_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & -n_{22} \\ 0 & 0 & 0 & 0 \end{pmatrix}, I_4 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix},$$

$$T_1 = \begin{pmatrix} (\mathfrak{L}-1)\tau & 0 & 0 & 0 \\ 0 & (\mathfrak{L}-1)\tau & 0 & 0 \\ 0 & 0 & (\mathfrak{L}-1)\tau & 0 \\ 0 & 0 & 0 & (\mathfrak{L}-1)\tau \end{pmatrix}, G_1 = \mathfrak{L}\tau \begin{pmatrix} f_{i,j} \\ f_{i+2,j} \\ f_{i+2,j+2} \\ f_{i,j+2} \end{pmatrix},$$

with

$$n_{11} = 1 + (\mathfrak{L}-1)\tau + \mathfrak{M}, \quad n_{22} = (\frac{\mathfrak{M}}{4} - \frac{\mathfrak{N}}{4}), \quad n_{33} = (\frac{\mathfrak{M}}{4} + \frac{\mathfrak{N}}{4}).$$

The next theorem is about the stability of this scheme.

Theorem 4.3. *The MHEG scheme given by Eq (3.6) is unconditionally stable.*

Proof. Suppose the exact and numerical solutions of Eq (4.1) are denoted by u^n and $\hat{u}^n$, respectively. Let $\varepsilon^n = u^n - \hat{u}^n$ be the error defined at the time level n. From Remarks 4.1 and 4.2, we realize that A is non-singular and thus Eq (4.1) can be written as

$$u^{n+1} = \mathfrak{A}^{-1}\mathfrak{B}u^n + A^{-1}\mathfrak{C}u^0 + \mathfrak{A}^{-1}b, \quad 0 \le n \le N-1. \tag{4.2}$$

From (4.2), we get the round-off error equation written as,

$$\varepsilon^{n+1} = \mathfrak{A}^{-1}\mathfrak{B}\varepsilon^n + \mathfrak{A}^{-1}\mathfrak{C}\varepsilon^0, \quad 0 \le n \le N-1, \tag{4.3}$$

where

$$\varepsilon^{n+1} = \begin{pmatrix} \varepsilon_0^{n+1} \\ \varepsilon_0^{n+1} \\ \vdots \\ \varepsilon_0^{n+1} \end{pmatrix}, \varepsilon_0^{n+1} = \begin{pmatrix} \psi_1^{n+1} \\ \psi_2^{n+1} \\ \vdots \\ \psi_{\frac{(M-2)^2}{16}}^{n+1} \end{pmatrix}, \psi^{n+1} = \begin{pmatrix} \psi_{i,j}^{n+1} \\ \psi_{i+2,j}^{n+1} \\ \psi_{i+2,j+2}^{n+1} \\ \psi_{i,j+2}^{n+1} \end{pmatrix},$$

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

In order to prove the stability, we employ mathematical induction to show that $\|\varepsilon^{n+1}\| \leq \|\varepsilon^0\|$ for all $0 \leq n \leq N-1$.

For $n = 0$, we get

$$\varepsilon^1 = \mathfrak{A}^{-1}\mathfrak{B}\varepsilon^0 + \mathfrak{A}^{-1}\mathfrak{C}\varepsilon^0.$$

As the matrix infinity norm $\|\mathfrak{A}\|$ and vector infinity norm $\|\varepsilon\|$ are consistent, we get

$$\begin{aligned} \|\varepsilon^1\| &= \|\mathfrak{A}^{-1}\mathfrak{B}\varepsilon^0 + \mathfrak{A}^{-1}\mathfrak{C}\varepsilon^0\| \\ &\leq \|\mathfrak{A}^{-1}\mathfrak{B}\|\|\varepsilon^0\| + \|\mathfrak{A}^{-1}\mathfrak{C}\|\|\varepsilon^0\| \\ &\leq \|\mathfrak{A}^{-1}\|\|\mathfrak{B}\|\|\varepsilon^0\| + \|\mathfrak{A}^{-1}\|\|\mathfrak{C}\|\|\varepsilon^0\| \\ &= (\|\mathfrak{B}\| + \|\mathfrak{C}\|)\|\mathfrak{A}^{-1}\|\|\varepsilon^0\|. \end{aligned}$$

Here $\mathfrak{A}$ is SDD matrix. Making use of Remarks 4.1 and 4.2, we obtain

$$\begin{aligned} \|\varepsilon^1\| &\leq \frac{(\|\mathfrak{B}\| + \|\mathfrak{C}\|)}{\min_{1\leq i\leq M}\left\{|\mathcal{A}_{i,i}| - \mathcal{R}_i(\mathfrak{A})\right\}}\|\varepsilon^0\| \\ &\quad \frac{1 + (\mathfrak{L}-1)\tau}{|1 + (\mathfrak{L}-1)\tau + \mathfrak{M}| - (|-n_{22}| + |-n_{22}| + |-n_{33}| + |-n_{33}|)} \\ &= \frac{1 + (\mathfrak{L}-1)\tau}{1 + (\mathfrak{L}-1)\tau}\|\varepsilon^0\| = \|\varepsilon^0\|. \end{aligned}$$

$$\therefore \|\varepsilon^1\| \leq \|\varepsilon^0\|.$$

Next, suppose that

$$\|\varepsilon^{s+1}\| \leq \|\varepsilon^0\|, \quad s = 1, 2, \ldots, n-1. \tag{4.4}$$

We will prove the above inequality for $s = n$. From (4.3) and (4.4), we have

$$\begin{aligned} \|\varepsilon^{n+1}\| &= \|\mathfrak{A}^{-1}\mathfrak{B}\varepsilon^n + \mathfrak{A}^{-1}\mathfrak{C}\varepsilon^0\| \\ &\leq \|\mathfrak{A}^{-1}\|\|\mathfrak{B}\|\|\varepsilon^n\| + \|\mathfrak{A}^{-1}\|\|\mathfrak{C}\|\|\varepsilon^0\| \\ &\leq \|\mathfrak{A}^{-1}\|\|\mathfrak{B}\|\|\varepsilon^0\| + \|\mathfrak{A}^{-1}\|\|\mathfrak{C}\|\|\varepsilon^0\| \\ &\leq \frac{(\|\mathfrak{B}\| + \|\mathfrak{C}\|)}{\min_{1\leq i\leq M}\left\{|\mathcal{A}_{i,i}| - \mathcal{R}_i(\mathfrak{A})\right\}}\|\varepsilon^0\| \\ &= \frac{1 + (\mathfrak{L}-1)\tau}{|1 + (\mathfrak{L}-1)\tau + \mathfrak{M}| - (|-n_{22}| + |-n_{22}| + |-n_{33}| + |-n_{33}|)} \\ &= \frac{1 + (\mathfrak{L}-1)\tau}{1 + (\mathfrak{L}-1)\tau}\|\varepsilon^0\| = \|\varepsilon^0\|. \end{aligned}$$

$$\therefore \|\varepsilon^{n+1}\| \leq \|\varepsilon^0\|.$$

This completes the proof.

4.2. Convergence analysis

Here, we will prove the convergence of the MHEG scheme (3.6). At any time level, let the truncation error on each group of four mesh points be expressed in the following form of the block vector:

$$R^{n+1} = (R_1^{n+1}, R_2^{n+1}, \ldots, R_{\frac{(M-2)^2}{16}}^{n+1})^T,$$

$$R_p^{n+1} = (R_{i,j}^{n+1}, R_{i+2,j}^{n+1}, R_{i+2,j+2}^{n+1}, R_{i,j+2}^{n+1})^T,\ 1 \le p \le \frac{(M-2)^2}{16}.$$

Then from (3.3), there is a positive constant C^* such that

$$\|R^{n+1}\| \le C^*(\tau + h^2), \quad 0 \le n \le N-1. \tag{4.5}$$

The subtraction of Eq (4.1) from the equation that generate the exact solution of Eq (2.5),

$$\mathfrak{A}U^{n+1} = \mathfrak{B}U^n + \mathfrak{C}U^0 + b + R^{n+1},$$

will lead to the error equation of the following form:

$$\mathfrak{A}\xi^{n+1} = \mathfrak{B}\xi^n + \mathfrak{C}\xi^0 + R^{n+1}, \tag{4.6}$$

where

$$\xi^{n+1} = \begin{pmatrix} \xi_0^{n+1} \\ \xi_0^{n+1} \\ \vdots \\ \xi_0^{n+1} \end{pmatrix},\ \xi_0^{n+1} = \begin{pmatrix} \phi_1^{n+1} \\ \phi_2^{n+1} \\ \vdots \\ \phi_{\frac{(M-2)^2}{16}}^{n+1} \end{pmatrix},\ \phi^{n+1} = \begin{pmatrix} \phi_{i,j}^{n+1} \\ \phi_{i+2,j}^{n+1} \\ \phi_{i+2,j+2}^{n+1} \\ \phi_{i,j+2}^{n+1} \end{pmatrix},$$

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

Theorem 4.4. *The MHEG scheme defined in (3.6) is convergent, and the following estimate* $\|\xi^{n+1}\| \le C_n(\tau + h^2)$ *does hold.*

Proof. Mathematical induction will be used for the proof. For $n = 0$ and utilizing that $\xi^0 = 0$, we get

$$\xi^1 = \mathfrak{A}^{-1}R^1.$$

Noticing Remark 4.2 and using (4.5), then

$$\|\xi^1\| = \|\mathfrak{A}^{-1}R^1\| \le \|\mathfrak{A}^{-1}\|\|R^1\| \le \frac{1}{\min_{1\le i\le M}\left\{|\mathcal{A}_{i,i}| - \mathcal{R}_i(\mathfrak{A})\right\}} C^*(\tau + h^2)$$

$$= \frac{1}{1 + (\mathfrak{L} - 1)\tau} C^*(\tau + h^2) = C_0(\tau + h^2),$$

where $C_0 = C^*/(1 + (\mathfrak{L} - 1)\tau)$.

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

Now, suppose that

$$\xi^{s+1} \le C_s(\tau + h^2), \quad s = 1, 2, \ldots, n-1. \tag{4.7}$$

We show that the last result does hold for $s = n$. From (4.6) and (4.7), we have

$$\begin{aligned}
\|\xi^{n+1}\| &= \|\mathfrak{A}^{-1}\mathfrak{B}\xi^{n} + \mathfrak{A}^{-1}R^{n+1}\| \\
&\leq \|\mathfrak{A}^{-1}\|\|\mathfrak{B}\|\|\xi^{n}\| + \|\mathfrak{A}^{-1}\|\|R^{n+1}\| \\
&\leq \frac{1}{\min_{1\leq i\leq M}\left\{|\mathcal{A}_{i,i}| - \mathcal{R}_i(\mathfrak{A})\right\}}\left[C_{n-1}(\tau + h^2) + C^*(\tau + h^2)\right] \\
&= \frac{1}{1 + (\mathfrak{Q} - 1)\tau}(C_{n-1} + C^*)(\tau + h^2) \\
&= C_n(\tau + h^2),
\end{aligned}$$

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

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

5. Numerical results

Some numerical simulations are provided in this part in order to illustrate the performance of the numerical solution algorithms in dealing with the TFADE (1.1). For the sake of comparison, we test the existing HSP method developed by Salama and Ali [41] together with the proposed HEG and MHEG methods. The three methods are solved by Gauss-Seidel iterative solver and implemented utilizing Mathematica 11.3 on a laptop with the configuration: Intel (R) Core (TM) i7-8550U and 8GB of RAM. In all experiments, the infinity norm l_∞ along with error tolerance $\epsilon = 10^{-5}$ are used for the stopping criterion. It is well known that the computational complexity of an iterative algorithm is mostly influenced by the number of iterations needed to attain convergence. As a result, an analysis of the computational cost based on the count of arithmetic operations performed per iteration is presented in Table 1. With this, the comparison between the tested methods are demonstrated in terms of elapsed CPU time (Sec), average iteration number (η), maximum absolute error ($Max\ Err$) and total arithmetic operations (TAO).

Table 1. The total arithmetic operations (TAO) of the HSP, HEG and MHEG methods ($\rho = M - 1$).

Method	*Per iteration*	*After convergence*	*TAO*
HSP	$15\rho^2 * \eta$	-	$15\rho^2\eta$
HEG	$\left(18(\rho-1)^2 + 15(2\rho-1)\right)\eta$	-	$\left(18(\rho-1)^2 + 15(2\rho-1)\right)\eta$
MHEG	$4.5(\rho-1)^2\eta$	$3.75(3\rho^2 + 2\rho - 1)$	$4.5(\rho-1)^2\eta + 3.75(3\rho^2 + 2\rho - 1)$

The following two numerical examples are considered:

Example 5.1 ([63])**.**

$$\begin{aligned}
{}_0^C D_t^{\gamma} u(x,y,t) = {} & \frac{\partial^2 u(x,y,t)}{\partial x^2} + \frac{\partial^2 u(x,y,t)}{\partial y^2} - \frac{\partial u(x,y,t)}{\partial x} - \frac{\partial u(x,y,t)}{\partial y} \\
& + t(\cos(x) + \sin(x) + \cos(y) + \sin(y)) + \frac{t^{1-\gamma}(\sin(x) + \sin(y))}{\Gamma(2-\gamma)},
\end{aligned}$$

with the initial and boundary conditions,

$$\begin{aligned} &u(x, y, 0) = 0, \quad 0 \le x, y, \le 1, \\ &u(0, y, t) = t\sin(y), \quad u(1, y, t) = t(\sin(1) + \sin(y)), \quad 0 < t \le 1, \\ &u(x, 0, t) = t\sin(x), \quad u(x, 1, t) = t(\sin(x) + \sin(1)), \quad 0 < t \le 1, \end{aligned}$$

and the exact analytical solution is

$$u(x, y, t) = t(\sin(x) + \sin(y)).$$

Example 5.2 ([28])**.**

$$\begin{aligned} {}_0^C D_t^\gamma u(x, y, t) = &\frac{\partial^2 u(x, y, t)}{\partial x^2} + \frac{\partial^2 u(x, y, t)}{\partial y^2} - \frac{\partial u(x, y, t)}{\partial x} - \frac{\partial u(x, y, t)}{\partial y} \\ &+ (\pi t^2)(\cos(\pi x) + \pi\sin(\pi x) + \cos(\pi y) + \pi\sin(\pi y)) \\ &+ \frac{2t^{2-\gamma}(\sin(\pi x) + \sin(\pi y))}{\Gamma(3-\gamma)}, \end{aligned}$$

subject to the initial and boundary conditions,

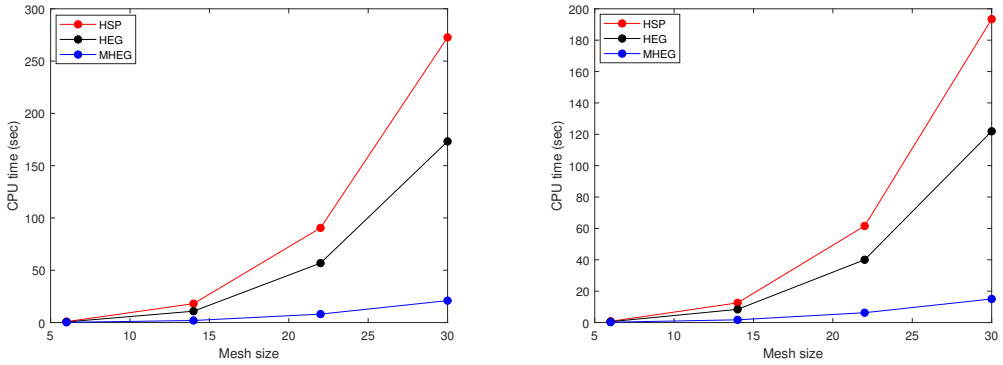
$$\begin{aligned} &u(x, y, 0) = 0, \quad 0 \le x, y, \le 1, \\ &u(0, y, t) = t^2\sin(\pi y), \quad u(1, y, t) = t^2\sin(\pi y), \quad 0 < t \le 1, \\ &u(x, 0, t) = t^2\sin(\pi x), \quad u(x, 1, t) = t^2\sin(\pi x), \quad 0 < t \le 1, \end{aligned}$$

and the exact analytical solution is

$$u(x, y, t) = t^2(\sin(\pi x) + \sin(\pi y)).$$

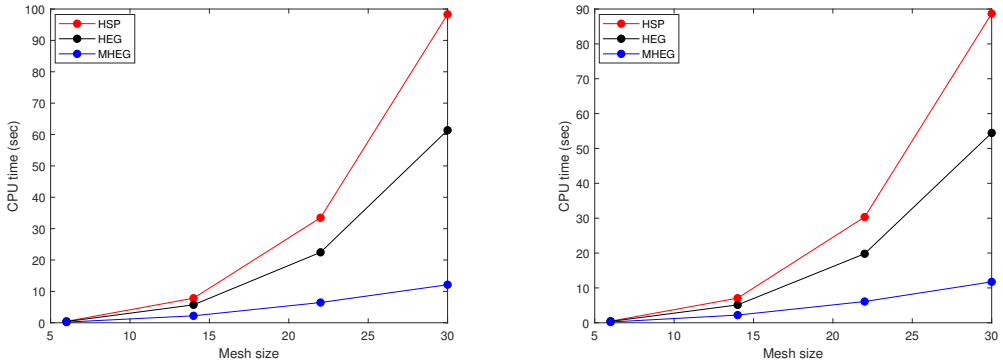
The initial and Dirichlet-type boundary conditions of the given numerical test problems can be drawn from the exact analytical solutions. We present the numerical results of Examples 5.1 and 5.2, which are solved by the HSP, HEG and MHEG iterative schemes at fixed time step $\tau = 0.01$ with respect to successively refined mesh sizes and different values of γ in Tables 2–5. From the tables along with Figures 3 and 4, it is evident that the proposed HEG and MHEG methods cost lesser CPU times than the HSP method. For instance, the CPU time in HEG method is reduced by (30.96–39.99)%, (28.62–36.97)%, (13.90–37.57)% and (13.90–38.60)% as compared to the HSP method in Tables 2 to 5, respectively. Similarly, the CPU time in the MHEG method is reduced by (74.62–92.32)%, (71.50–92.22)%, (51.87–87.66)% and (51.87–86.76)% in comparison to the HSP method in Tables 2 to 5, respectively. It follows based on all the tabulated results that the improvement percentages in terms of CPU time compared to the HSP method are about (13.90–39.99)% and (51.87–92.32)% for the HEG and MHEG methods, respectively. Likewise, it can be observed that the numbers of iterations and arithmetic operations of the HEG method are declined approximately by (0.25–41.17)% and (15.40–30.92)%, respectively, in comparison to the HSP method. Similarly, the counts of iterations and executed arithmetic operations of the MHEG method are decreased significantly by (75.00–87.50)% and (84.70–94.01)%, respectively, compared to the HSP method. These reductions in iteration number as well as computational cost can be seen in

Figures 5–8, which are consistent with the improvement in CPU timings portrayed in Figures 3 and 4. These results illustrate the success of the proposed methods to simulate the considered model problem with lower computing effort. By comparing the CPU times, iterations numbers and total arithmetic operations, we indicate that the MHEG method is the most efficient among the three tested methods. Figures 9 and 10 introduce the 3D plots for the maximum absolute errors of Examples 5.1 and 5.2, respectively. Again,these testify the effectiveness and reliability of the proposed methods. It is worth noting here that the accuracy of numerical solutions is dependent on the Laplace transform technique discussed in [41]. The effect of the aforementioned technique on the accuracy of numerical solutions is an intriguing line of future research. The tabulated and sketched results show that the proposed methods can achieve acceptable accuracy for various values of fractional order γ. Considering the proposed methods' simplicity and computational efficiency, they can be a reliable approach that achieves an acceptable accuracy for solving the problem under consideration.



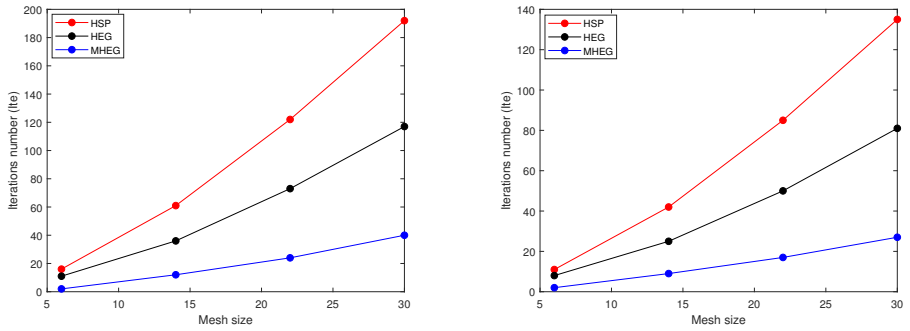
(a) $\gamma = 0.1$ (b) $\gamma = 0.3$

Figure 3. 2D plots CPU time for Example 5.1.



(a) $\gamma = 0.1$ (b) $\gamma = 0.3$

Figure 4. 2D plots CPU time for Example 5.2.



(a) $\gamma = 0.1$ (b) $\gamma = 0.3$

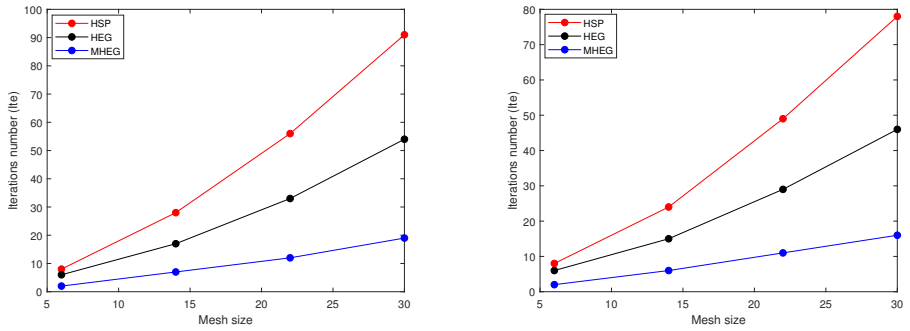
Figure 5. 2D plots iterations number for Example 5.1.

Table 2. The CPU time, iterations number, maximum errors and total operations of Example 5.1 at $\gamma = 0.1$ and $\tau = 0.01$.

h^{-1}	Method	Sec	η	$Max\ Err$	TAO
6	HSP	0.859	16	3.1180E-03	6,000
	HEG	0.593	11	3.1233E-03	4,653
	MHEG	0.218	2	4.7176E-03	459
14	HSP	18.046	61	2.6121E-03	154,635
	HEG	10.828	36	2.6943E-03	106,812
	MHEG	1.953	12	3.1139E-03	9,771
22	HSP	90.531	122	2.2954E-03	807,030
	HEG	56.828	73	2.5172E-03	570,495
	MHEG	8.046	24	2.8194E-03	48,315
30	HSP	272.672	192	1.9199E-03	2,422,080
	HEG	173.188	117	2.2979E-03	1,751,139
	MHEG	20.953	40	2.6958E-03	150,795

Table 3. The CPU time, iterations number, maximum errors and total operations of Example 5.1 at $\gamma = 0.3$ and $\tau = 0.01$.

h^{-1}	Method	Sec	η	$Max\ Err$	TAO
6	HSP	0.765	11	7.0109E-03	4,125
	HEG	0.546	8	7.0164E-03	3,384
	HMEG	0.218	2	8.4447E-03	459
14	HSP	12.546	42	6.6935E-03	106,470
	HEG	8.406	25	6.7768E-03	74,175
	MHEG	1.734	9	7.1581E-03	7,827
22	HSP	61.562	85	6.4074E-03	562,275
	HEG	40.015	50	6.6223E-03	390,750
	MHEG	6.265	17	6.9166E-03	35,715
30	HSP	193.422	135	6.0044E-03	1,703,025
	HEG	121.906	81	6.4234E-03	1,212,327
	MHEG	15.031	27	6.7921E-03	104,931



(a) $\gamma = 0.7$ (b) $\gamma = 0.9$

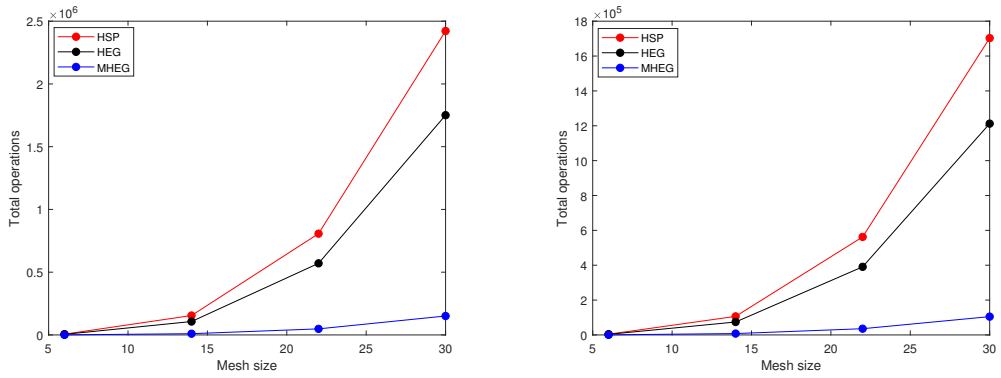
Figure 6. 2D plots iterations number for Example 5.2

Table 4. The CPU time, iterations number, maximum errors and total operations of Example 5.2 at $\gamma = 0.7$ and $\tau = 0.01$.

h^{-1}	Method	Sec	η	$Max\ Err$	TAO
6	HSP	0.453	8	2.6906E-02	3,000
	HEG	0.390	6	2.6908E-02	2,538
	MHEG	0.218	2	8.6102E-02	459
14	HSP	7.828	28	6.1030E-03	70,980
	HEG	5.718	17	6.1995E-03	50,439
	MHEG	2.218	7	2.0151E-02	6,531
22	HSP	33.484	56	3.0232E-03	370,440
	HEG	22.453	33	3.2549E-03	257,895
	MHEG	6.4531	12	9.1659E-03	26,715
30	HSP	98.312	91	1.7293E-03	1,147,965
	HEG	61.375	54	2.1788E-03	808,218
	MHEG	12.125	19	5.6006E-03	76,707

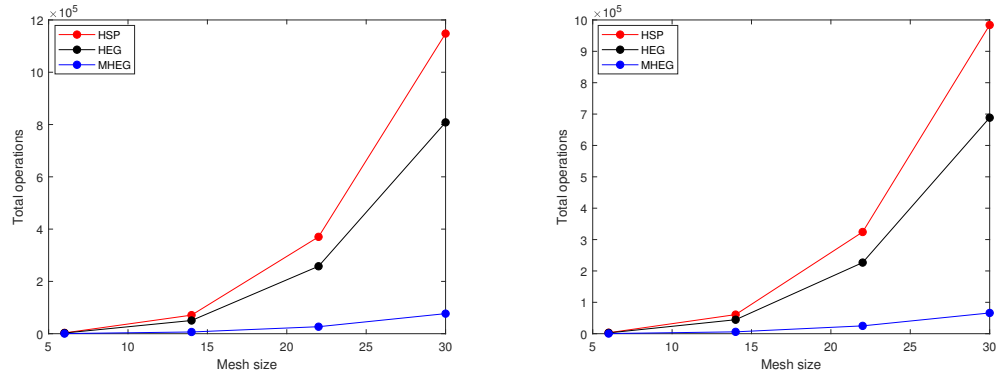
Table 5. The CPU time, iterations number, maximum errors and total operations of Example 5.2 at $\gamma = 0.9$ and $\tau = 0.01$.

h^{-1}	Method	Sec	η	$Max\ Err$	TAO
6	HSP	0.453	8	2.6920E-02	3,000
	HEG	0.390	6	2.6925E-02	2,538
	MHEG	0.218	2	8.5452E-02	459
14	HSP	7.062	24	6.3499E-03	60,840
	HEG	5.125	15	6.4284E-03	44,505
	MHEG	2.203	6	2.0218E-02	5,883
22	HSP	30.328	49	3.2926E-03	324,135
	HEG	19.796	29	3.5445E-03	226,635
	MHEG	6.093	11	9.3556E-03	24,915
30	HSP	88.687	78	1.9995E-03	983,970
	HEG	54.453	46	2.4626E-03	688,482
	MHEG	11.734	16	5.8358E-03	66,123



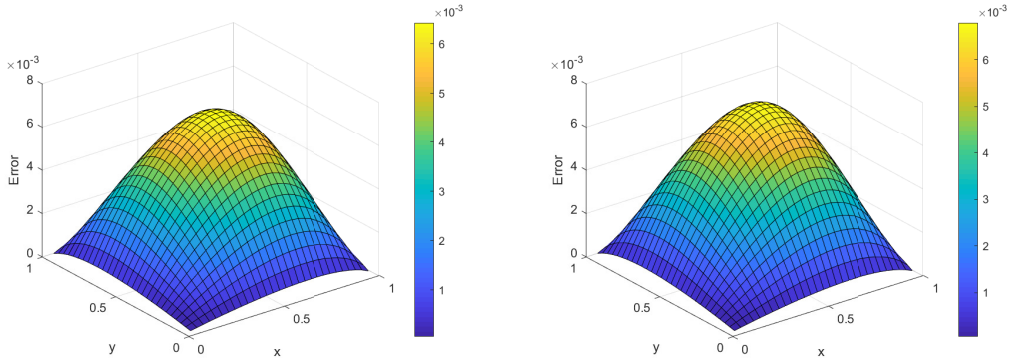
(a) $\gamma = 0.1$ (b) $\gamma = 0.3$

Figure 7. 2D plots total arithmetic operations for Example 5.1.



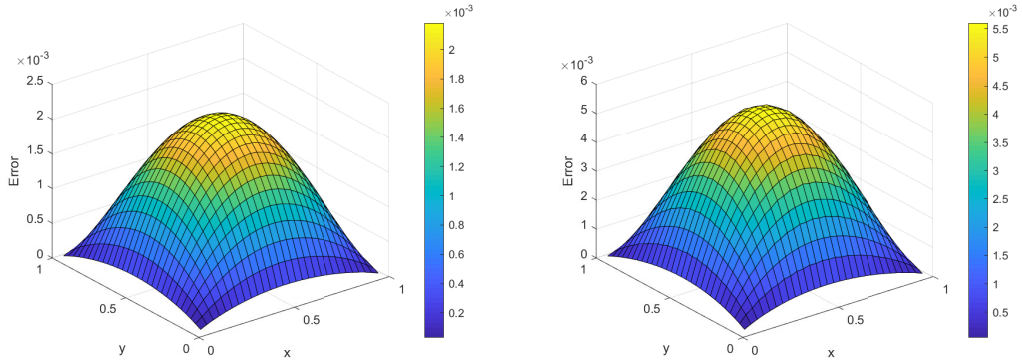
(a) $\gamma = 0.7$ (b) $\gamma = 0.9$

Figure 8. 2D plots total arithmetic operations for Example 5.2.



(a) HEG (b) MHEG

Figure 9. 3D plots maximum absolute errors for Example 5.1 when $\gamma = 0.3$ and $h = 1/30$.



(a) HEG (b) MHEG

Figure 10. 3D plots maximum absolute errors for Example 5.2 when $\gamma = 0.7$ and $h = 1/30$.

6. Conclusions

In this paper, the HEG and MHEG methods are developed for fast and accurate numerical solutions of the 2D TFADE. In our methods, we borrowed the idea of the Laplace transform technique [52] to convert the original fractional advection-diffusion problem (1.1) into its corresponding PDE. Afterwards, two implicit difference schemes are used to discretize the resulting PDE and construct the HEG and MHEG methods. The stability and convergence analyses are investigated rigorously by the means of the matrix norm analysis, which shows that the proposed methods are stable and convergent without any restricting conditions. Numerical test problems with tabulated and sketched results are provided to verify the applicability, accuracy and efficiency of the proposed methods. The obtained numerical results revealed that both HEG and MHEG methods compare well with the exact solutions and require less iterations numbers, computational costs and hence CPU times in comparison to the HSP method introduced in [41]. On average, the HEG method decreased the CPU time and iterations number compared to the HSP method by 31.51% and 36.91%, respectively. In addition, the MHEG method decreased the mentioned outcomes by 79.13% and 78.90%, respectively, compared to the HSP method. Moreover, the computational efficiency of the MHEG method is shown to be optimal among all the tested methods. Hence, The power of the HEG and MHEG methods to reduce the amount of computational complexity in solving the advection-diffusion problem of fractional order is shown. As a future work, we recommend parallel implementation of the proposed methods in this work.

Acknowledgments

The authors extend their appreciation to Universiti Sains Malaysia for supporting this work by Research University Grant 1001/PMATHS/8011101. The authors are also grateful to the anonymous reviewers for their careful reading and useful suggestions, which have greatly improved the presentation of the paper.

Conflict of interest

The authors declare no conflicts of interest.

References

1. M. Javaid, M. Tahir, M. Imran, D. Baleanu, A. Akgül, M. A. Imran, Unsteady flow of fractional Burgers' fluid in a rotating annulus region with power law kernel, *Alex. Eng. J.*, **61** (2022), 17–27. https://doi.org/10.1016/j.aej.2021.04.106
2. T. Anwar, P. Kumam, P. Thounthong, Asifa, S. Muhammad, F. Z. Duraihem, Generalized thermal investigation of unsteady MHD flow of Oldroyd-B fluid with slip effects and Newtonian heating; a Caputo-Fabrizio fractional model, *Alex. Eng. J.*, **61** (2022), 2188–2202. https://doi.org/10.1016/j.aej.2021.06.090
3. C. Xu, C. Aouiti, Z. Liu, Q. Qin, L. Yao, Bifurcation control strategy for a fractional-order delayed financial crises contagions model, *AIMS Math.*, **7** (2022), 2102–2122. https://doi.org/10.3934/math.2022120
4. W. A. E. M. Ahmed, H. M. A. Mageed, S. A. Mohamed, A. A. Saleh, Fractional order Darwinian particle swarm optimization for parameters identification of solar PV cells and modules, *Alex. Eng. J.*, **61** (2022), 1249–1263. https://doi.org/10.1016/j.aej.2021.06.019
5. M. Farman, A. Akgül, K. S. Nisar, D. Ahmad, A. Ahmad, S. Kamangar, et al., Epidemiological analysis of fractional order COVID-19 model with Mittag-Leffler kernel, *AIMS Math.*, **7** (2022), 1249–1263. https://doi.org/10.3934/math.2022046
6. M. S. Ullah, M. Higazy, K. A. Kabir, Modeling the epidemic control measures in overcoming COVID-19 outbreaks: A fractional-order derivative approach, *Chaos Soliton. Fract.*, **155** (2022), 111636. https://doi.org/10.1016/j.chaos.2021.111636
7. M. Farman, A. Akgül, T. Abdeljawad, P. A. Naik, N. Bukhari, A. Ahmad, Modeling and analysis of fractional order Ebola virus model with Mittag-Leffler kernel, *Alex. Eng. J.*, **61** (2022), 2062–2073. https://doi.org/10.1016/j.aej.2021.07.040
8. E. Bonyah, M. L. Juga, C. W. Chukwu, Fatmawati, A fractional order dengue fever model in the context of protected travelers, *Alex. Eng. J.*, **61** (2022), 927–936. https://doi.org/10.1016/j.aej.2021.04.070
9. S. Rashid, F. Jarad, F. S. Bayones, On new computations of the fractional epidemic childhood disease model pertaining to the generalized fractional derivative with nonsingular kernel, *AIMS Math.*, **7** (2022), 4552–4573. http://dx.doi.org/10.3934/math.2022254
10. A. S. V. Ravi Kanth, N. Garg, An unconditionally stable algorithm for multiterm time fractional advection-diffusion equation with variable coefficients and convergence analysis, *Numer. Methods Partial Differ. Equations*, **37** (2021), 1928–1945. https://doi.org/10.1002/num.22629
11. S. Savović, A. Djordjevich, Finite difference solution of the one-dimensional advection-diffusion equation with variable coefficients in semi-infinite media, *Int. J. Heat Mass Tran.*, **55** (2012), 4291–4294. https://doi.org/10.1016/j.ijheatmasstransfer.2012.03.073

12. H. Tajadodi, A Numerical approach of fractional advection-diffusion equation with Atangana-Baleanu derivative, *Chaos Soliton. Fract.*, **130** (2020), 109527. https://doi.org/10.1016/j.chaos.2019.109527

13. S. U. S. Choi, J. A. Eastman, *Enhancing thermal conductivity of fluids with nanoparticles*, New York: Argonne National Lab., 1995.

14. T. Hayat, M. Tamoor, M. I. Khan, A. Alsaedi, Numerical simulation for nonlinear radiative flow by convective cylinder, *Results Phys.*, **6** (2016), 1031–1035. https://doi.org/10.1016/j.rinp.2016.11.026

15. S. Qayyum, M. I. Khan, T. Hayat, A. Alsaedi, Comparative investigation of five nanoparticles in flow of viscous fluid with Joule heating and slip due to rotating disk, *Phys. B*, **534** (2018), 173–183. https://doi.org/10.1016/j.physb.2018.01.044

16. M. Waqas, M. I. Khan, T. Hayat, M. M. Gulzar, A. Alsaedi, Transportation of radiative energy in viscoelastic nanofluid considering buoyancy forces and convective conditions, *Chaos Soliton. Fract.*, **130** (2020), 109415. https://doi.org/10.1016/j.chaos.2019.109415

17. M. I. Khan, A. Alsaedi, T. Hayat, N. B. Khan, Modeling and computational analysis of hybrid class nanomaterials subject to entropy generation, *Comp. Methods Prog. Biom.*, **179** (2019), 104973. https://doi.org/10.1016/j.cmpb.2019.07.001

18. B. Gireesha, G. Sowmya, M. I. Khan, H. F. Öztop, Flow of hybrid nanofluid across a permeable longitudinal moving fin along with thermal radiation and natural convection, *Comp. Methods Prog. Biom.*, **185** (2020), 105166. https://doi.org/10.1016/j.cmpb.2019.105166

19. T. Hayat, S. Ahmad, M. I. Khan, A. Alsaedi, Simulation of ferromagnetic nanomaterial flow of Maxwell fluid, *Results Phys.*, **8** (2018), 34–40. https://doi.org/10.1016/j.rinp.2017.11.021

20. H. Sun, Y. Zhang, D. Baleanu, W. Chen, Y. Chen, A new collection of real world applications of fractional calculus in science and engineering, *Commun. Nonlinear Sci. Numer. Simul.*, **64** (2018), 213–231. https://doi.org/10.1016/j.cnsns.2018.04.019

21. 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. Funct. Anal. Optim.*, **41** (2020), 257–293. https://doi.org/10.1080/01630563.2019.1627369

22. G. H. Gao, H. W. Sun, Three-point combined compact difference schemes for time-fractional advection-diffusion equations with smooth solutions, *J. Comput. Phys.*, **298** (2015), 520–538. https://doi.org/10.1016/j.jcp.2015.05.052

23. A. Mardani, M. R. Hooshmandasl, M. H. Heydari, C. Cattani, A meshless method for solving the time fractional advection-diffusion equation with variable coefficients, *Comput. Math. Appl.*, **75** (2018), 122–133. https://doi.org/10.1016/j.camwa.2017.08.038

24. A. Tayebi, Y. Shekari, M. H. Heydari, A meshless method for solving two-dimensional variable-order time fractional advection–diffusion equation, *J. Comput. Phys.*, **340** (2017), 655–669. https://doi.org/10.1016/j.jcp.2017.03.061

25. C. E. Mejía, A. Piedrahita, A numerical method for a time-fractional advection-dispersion equation with a nonlinear source term, *J. Appl. Math. Comput.*, **61** (2019), 593–609. https://doi.org/10.1007/s12190-019-01266-x

26. S. T. Mohyud-Din, T. Akram, M. Abbas, A. I. Ismail, N. H. Ali, A fully implicit finite difference scheme based on extended cubic B-splines for time fractional advection-diffusion equation, *Adv. Differ. Equ.*, **2018** (2018), 109. https://doi.org/10.1186/s13662-018-1537-7

27. M. Abbaszadeh, H. Amjadian, Second-order finite difference/spectral element formulation for solving the fractional advection-diffusion equation, *Commun. Appl. Math. Comput.*, **2** (2020), 1–17. https://doi.org/10.1007/s42967-020-00060-y

28. M. Badr, A. Yazdani, H. Jafari, Stability of a finite volume element method for the time-fractional advection-diffusion equation, *Numer. Methods Partial Differ. Equations*, **34** (2018), 1459–1471. https://doi.org/10.1002/num.22243

29. H. Azin, F. Mohammadi, M. Heydari, A hybrid method for solving time fractional advection-diffusion equation on unbounded space domain, *Adv. Differ. Equ.*, **2020** (2020), 1–10. https://doi.org/10.1186/s13662-020-03053-6

30. Y. E. Aghdam, H. Mesgrani, M. Javidi, O. Nikan, A computational approach for the space-time fractional advection-diffusion equation arising in contaminant transport through porous media, *Eng. Comput.*, **37** (2021), 3615–3627. https://doi.org/10.1007/s00366-020-01021-y

31. M. Shafiq, M. Abbas, K. M. Abualnaja, M. Huntul, A. Majeed, T. Nazir, An efficient technique based on cubic B-spline functions for solving time-fractional advection diffusion equation involving Atangana-Baleanu derivative, *Eng. Comput.*, **38** (2022), 901–917. https://doi.org/10.1007/s00366-021-01490-9

32. Z. Liu, Q. Wang, A non-standard finite difference method for space fractional advection-diffusion equation, *Numer. Methods Partial Differ. Equations*, **37** (2021), 2527–2539. https://doi.org/10.1002/num.22734

33. A. Atangana, Fractional discretization: The African's tortoise walk, *Chaos Soliton. Fract.*, **130** (2020), 109399. https://doi.org/10.1016/j.chaos.2019.109399

34. A. Sunarto, P. Agarwal, J. Sulaiman, J. V. L. Chew, S. Momani, Quarter-sweep preconditioned relaxation method, algorithm and efficiency analysis for fractional mathematical equation, *Fractal Fract.*, **5** (2021), 98. https://doi.org/10.3390/fractalfract5030098

35. A. M. Saeed, N. M. AL-harbi, Group splitting with SOR/AOR methods for solving boundary value problems: A computational comparison, *Eur. J. Pure Appl. Math.*, **14** (2021), 905–914. https://doi.org/10.29020/nybg.ejpam.v14i3.4031

36. 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.

37. 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. http://dx.doi.org/10.22436/jmcs.023.02.04

38. C. Gong, W. Bao, G. Tang, B. Yang, J. Liu, An efficient parallel solution for Caputo fractional reaction-diffusion equation, *J. Supercomput.*, **68** (2014), 1521–1537. https://doi.org/10.1007/s11227-014-1123-z

39. F. R. Lin, S. W. Yang, X. Q. Jin, Preconditioned iterative methods for fractional diffusion equation, *J. Comput. Phys.,* **256** (2014), 109–117. https://doi.org/10.1016/j.jcp.2013.07.040

40. Y. Xu, Z. He, The short memory principle for solving Abel differential equation of fractional order, *Comput. Math. Appl.,* **62** (2011), 4796–4805. https://doi.org/10.1016/j.camwa.2011.10.071

41. 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), 4796–4805. https://doi.org/10.33889/IJMEMS.2020.5.3.036

42. N. H. M. Ali, K. Foo, Modified explicit group AOR methods in the solution of elliptic equations, *Appl. Math. Sci.,* **6** (2012), 2465–2480.

43. K. B. Tan, N. H. M. Ali, C. H. Lai, Parallel block interface domain decomposition methods for the 2D convection–diffusion equation, *Int. J. Comput. Math.,* **89** (2012), 1704–1723. https://doi.org/10.1080/00207160.2012.693606

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

45. 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. https://doi.org/10.1016/j.jcp.2012.06.025

46. 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. https://doi.org/10.1016/j.jcp.2015.03.052

47. M. A. Khan, N. H. M. Ali, N. N. Abd Hamid, A new fourth-order explicit group method in the solution of two-dimensional fractional Rayleigh-Stokes problem for a heated generalized second-grade fluid, *Adv. Differ. Equ.*, **2020** (2020), 598. https://doi.org/10.1186/s13662-020-03061-6

48. 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. https://doi.org/10.1186/s13662-020-02717-7

49. A. Ali, T. Abdeljawad, A. Iqbal, T. Akram, M. Abbas, On unconditionally stable new modified fractional group iterative scheme for the solution of 2D time-fractional telegraph model, *Symmetry*, **13** (2021), 2078. https://doi.org/10.3390/sym13112078

50. N. Abdi, H. Aminikhah, A. 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. https://doi.org/10.1155/2021/6669287

51. F. M. Salama, N. N. Abd Hamid, N. H. M. Ali, U. Ali, An efficient modified hybrid explicit group iterative method for the time-fractional diffusion equation in two space dimensions, *AIMS Math.*, **7** (2022), 2370–2392. https://doi.org/10.3934/math.2022134

52. 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. https://doi.org/10.1016/j.apm.2015.10.011

53. S. Jiang, J. Zhang, Q. Zhang, Z. Zhang, Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations, *Commun. Comput. Phys.*, **21** (2017), 650–678. https://doi.org/10.4208/cicp.OA-2016-0136

54. Y. Yan, Z. Z. Sun, J. Zhang, Fast evaluation of the Caputo fractional derivative and its applications to fractional diffusion equations: A second-order scheme, *Commun. Comput. Phys.*, **22** (2017), 1028–1048. https://doi.org/10.4208/cicp.OA-2017-0019

55. M. Zhao, H. Wang, Fast finite difference methods for space-time fractional partial differential equations in three space dimensions with nonlocal boundary conditions, *Appl. Numer. Math.*, **145** (2019), 411–428. https://doi.org/10.1016/j.apnum.2019.05.007

56. P. Lyu, Y. Liang, Z. Wang, A fast linearized finite difference method for the nonlinear multi-term time-fractional wave equation, *Appl. Numer. Math.*, **151** (2020), 448–471. https://doi.org/10.1016/j.apnum.2019.11.012

57. C. Gong, W. Bao, G. Tang, Y. Jiang, J. Liu, Computational challenge of fractional differential equations and the potential solutions: A survey, *Math. Probl. Eng.*, **2015** (2015), 258265. https://doi.org/10.1155/2015/258265

58. Z. Liu, A. Cheng, X. Li, A fast-high order compact difference method for the fractional cable equation, *Numer. Methods Partial Differ. Equations,* **34** (2018), 2237–2266. https://doi.org/10.1002/num.22286

59. J. L. Zhang, Z. W. Fang, H. W. Sun, Exponential-sum-approximation technique for variable-order time-fractional diffusion equations, *J. Appl. Math. Comput.*, **68** (2022), 323–347. https://doi.org/10.1007/s12190-021-01528-7

60. A. Balasim, N. H. M. Ali, A rotated Crank-Nicolson iterative method for the solution of two-dimensional time-fractional diffusion equation, *Indian J. Sci. Technol.*, **8** (2015), 1–8. https://doi.org/10.17485/ijst/2015/v8i32/92045

61. R. A. Horn, C. R. Johnson, *Topics in matrix analysis*, Cambridge: Cambridge university press, 1994.

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

63. A. Mohebbi, M. Abbaszadeh, Compact finite difference scheme for the solution of time fractional advection-dispersion equation, *Numer. Algor.*, **63** (2013), 431–452. https://doi.org/10.1007/s11075-012-9631-5

AIMS Press