

An unconditionally stable implicit difference scheme for 2D porous medium equations using four-point NEGMSOR iterative method

Jackel Chew Vui Lung¹ and Jumat Sulaiman^{1,*}

¹Faculty of Science and Natural Resources, Universiti Malaysia Sabah, 88400, Kota Kinabalu, Sabah, Malaysia

Abstract. In this paper, a numerical method has been proposed for solving several two-dimensional porous medium equations (2D PME). The method combines Newton and Explicit Group MSOR (EGMSOR) iterative method namely four-point NEGMSOR. Throughout this paper, an initial-boundary value problem of 2D PME is discretized by using the implicit finite difference scheme in order to form a nonlinear approximation equation. By taking a fixed number of grid points in a solution domain, the formulated nonlinear approximation equation produces a large nonlinear system which is solved using the Newton iterative method. The solution vector of the sparse linearized system is then computed iteratively by the application of the four-point EGMSOR method. For the numerical experiments, three examples of 2D PME are used to illustrate the efficiency of the NEGMSOR. The numerical result reveals that the NEGMSOR has a better efficiency in terms of number of iterations, computation time and maximum absolute error compared to the tested NGS, NEG and NEGMSOR iterative methods. The stability analysis of the implicit finite difference scheme used on 2D PME is also provided.

1 Introduction

This paper considers the numerical solution of the following initial-boundary value problem of the two-dimensional porous medium equation (2D PME):

$$\frac{\partial u}{\partial t} = \alpha \left[\frac{\partial}{\partial x} \left(u^m \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(u^m \frac{\partial u}{\partial y} \right) \right], \tag{1}$$

that is subjected to the initial condition:

$$u(x, y, 0) = u_0(x, y), \quad 0 \leq x, y \leq 1, \tag{2}$$

and the boundary conditions:

* Corresponding author: jumat@ums.edu.my

$$\begin{aligned} u(0, y, t) &= g_1(y, t), \quad u(1, y, t) = g_2(y, t), \\ u(x, 0, t) &= g_3(x, t), \quad u(x, 1, t) = g_4(x, t), \quad 0 < t \leq 1, \end{aligned} \tag{3}$$

with some real parameters: $m \geq 1$ and $\alpha > 0$.

Equation (1) can be known as the generalized form of the heat equation. By referring to equation (1), the heat equation can be obtained by taking $m = 1$. In a particular case like $m = 2$, the Boussinesq equation can be obtained. Equation (1) is a type of partial differential equation in the nonlinear degenerate parabolic class which is widely used to describe models for gases in porous media, heat diffusion, underground infiltration, population dynamics as well as high energy physics [1]. The presence of a nonlinear term u^m in equation (1) makes the exact solution difficult to be obtained.

Generally, the numerical method can be considered as an alternative option in solving the partial differential equation especially for the case where the exact solution is almost impossible to be obtained. The numerical method yields an approximate solution that is differing from the exact solution by less than a specified tolerance. Furthermore, in terms of the implementation of the numerical method, efficiency is one of the important characteristics to be observed. The selected numerical method should be the one that requires less computation time and complexity in obtaining the desired result [2].

The numerical solution of multidimensional PME has received considerable interest from a number of researchers. For instance, Borana et al. [3] studied the numerical solution of the one-dimensional (1D) case of PME that is modelled based on the instability phenomenon in the double phase flow of two immiscible fluids through the inclined homogeneous porous medium. The method used is a Crank-Nicolson finite difference scheme. Meanwhile, Pradhan et al. [4] applied the finite element method for solving the 1D PME model that representing the instability phenomenon in the recovery of oil from a reservoir. For the numerical solution of 2D PMEs, Ngo and Huang [5] studied the application of an adaptive moving mesh finite element method as an improvement to the existing moving mesh method. In addition, Chew and Sulaiman [6] started the investigation on the efficiency of a numerical method that combines Newton and MSOR (NMSOR) iterative method for solving 1D PMEs. Then, Chew and Sulaiman [7] extended their study with the implementation of the Explicit Group (EG) method [8] to improve the efficiency of the NMSOR iterative method in solving the 1D PMEs.

The aim of this paper is to study the efficiency of the Newton and Explicit Group MSOR iterative method or NEGMSOR for solving several examples of 2D PME. The outline of this paper is arranged as follows. Section 2 shows the discretization of 2D PME by using the implicit finite difference scheme into a nonlinear approximation equation and the formation of a sparse and large nonlinear system. Section 3 presents the solution of the nonlinear system by the implementation of the Newton and EGMSOR iterative methods. Section 4 illustrates the numerical experiments and Section 5 concludes the finding of this study.

2 Implicit finite difference approximation equation

To obtain the numerical solutions, we consider a rectangular domain that is marked off in the unit square using the grid lines as follows:

$$x_i = ih, \quad y_j = jh, \quad h = \frac{1}{M}. \tag{4}$$

and for the time step, we define

$$t_n = nk, k = \frac{1}{N} \tag{5}$$

The numerical solutions that are represented by the points of intersection between the two grid lines are denoted by $u(x_i, y_j, t_n)$. For the sake of simplicity, we use $u_{i,j,n} = u(x_i, y_j, t_n)$ with $1 \leq i, j \leq M-1$ and $1 \leq n \leq N$. By using the implicit finite difference scheme, we obtain the implicit finite difference approximation equation of equation (1) that can be written as

$$\begin{aligned} f(u_{i,j,n+1}) = & u_{i,j,n+1} - A_1 \left(u_{i,j,n+1}^m u_{i+1,j,n+1} - 2u_{i,j,n+1}^{m+1} + u_{i,j,n+1}^m u_{i-1,j,n+1} \right) \\ & - A_2 \left(mu_{i,j,n+1}^{m-1} u_{i+1,j,n+1}^2 - 2mu_{i,j,n+1}^{m-1} u_{i+1,j,n+1} u_{i-1,j,n+1} + mu_{i,j,n+1}^{m-1} u_{i-1,j,n+1}^2 \right) \\ & - A_3 \left(u_{i,j,n+1}^m u_{i,j+1,n+1} - 2u_{i,j,n+1}^{m+1} + u_{i,j,n+1}^m u_{i,j-1,n+1} \right) \\ & - A_4 \left(mu_{i,j,n+1}^{m-1} u_{i,j+1,n+1}^2 - 2mu_{i,j,n+1}^{m-1} u_{i,j+1,n+1} u_{i,j-1,n+1} + mu_{i,j,n+1}^{m-1} u_{i,j-1,n+1}^2 \right) - u_{i,j,n} \end{aligned} \tag{6}$$

where $A_1 = A_3 = \alpha k / h^2$, $A_2 = A_4 = \alpha k / 4h^2$.

Equation (6) is the function at each interior grid point over the solution domain that links with neighbourhood grid points to both x- and y-directions. As a result, a sparse and large nonlinear system is formed and generally written as

$$F \begin{pmatrix} U \\ \sim \\ \sim_{n+1} \end{pmatrix} = 0 \tag{7}$$

where

$$\begin{aligned} F \begin{pmatrix} U \\ \sim \\ \sim_{n+1} \end{pmatrix} &= (f_{1,1}, f_{2,1}, \dots, f_{1,2}, f_{2,2}, \dots, f_{M-1,M-1})^T, \\ U_{\sim_{n+1}} &= (u_{1,1,n+1}, u_{2,1,n+1}, \dots, u_{1,2,n+1}, u_{2,2,n+1}, \dots, u_{M-1,M-1,n+1}). \end{aligned}$$

3 Linear stability analysis

Generally, the application of the linear Fourier analysis to the nonlinear partial differential equation such as 2D PME cannot be rigorously justified. Nevertheless, it has been found to effective in practice [9-12]. To show the stability of the implicit finite difference scheme used on 2D PME, we assume that the solution function $u(x, y, t)$ is bounded in the spatiotemporal region of $0 \leq x, y \leq 1$ and $0 \leq t \leq 1$. Now, the nonlinear term

u^m can be frozen at each grid point in the solution domain and gives

$$\frac{\partial u}{\partial t} = \rho \left(\frac{\partial}{\partial x} \left(v \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(v \frac{\partial u}{\partial y} \right) \right) \tag{8}$$

where $v = \max |u^m|$. Equation (8) can be rewritten into

$$\frac{\partial u}{\partial t} = \rho v \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) \tag{9}$$

From here, the stability of the implicit finite difference scheme used in this paper can be shown by implementing the von Neumann method over equation (9) as shown in Theorem 3.

Theorem 3.

The fully implicit finite difference scheme of equation (9) on the finite domain $0 \leq x, y \leq 1$ with zero boundary conditions for all $t \geq 0$ is unconditionally stable.

Proof. To begin, we define the corresponding implicit finite difference operators as follows.

$$D_{-t}U_{p,q,n+1} = \frac{U_{p,q,n+1} - U_{p,q,n}}{k} \tag{10}$$

$$D_{+x}D_{-x}U_{p,q,n+1} = \frac{U_{p+1,q,n+1} - 2U_{p,q,n+1} + U_{p-1,q,n+1}}{h^2} \tag{11}$$

and

$$D_{+y}D_{-y}U_{p,q,n+1} = \frac{U_{p,q+1,n+1} - 2U_{p,q,n+1} + U_{p,q-1,n+1}}{h^2} \tag{12}$$

By substituting equation (10) - (12) into equation (9) and after some rearranging, the corresponding equation obtained is

$$U_{p,q,n+1} - \lambda(U_{p+1,q,n+1} - 2U_{p,q,n+1} + U_{p-1,q,n+1}) - \lambda(U_{p,q+1,n+1} - 2U_{p,q,n+1} + U_{p,q-1,n+1}) = U_{p,q,n}, \tag{13}$$

where $\lambda = k\rho v/h^2$.

To apply the von Neumann method, let

$$U_{p,q,n} = \xi^n e^{i\alpha p} e^{i\beta q} \tag{14}$$

By using equation (14), equation (13) can be transformed into

$$\xi \left(1 - \lambda(e^{i\alpha} - 2 + e^{-i\alpha}) - \lambda(e^{i\beta} - 2 + e^{-i\beta}) \right) = 1. \tag{15}$$

Since $e^{i\alpha} - 2 + e^{-i\alpha} = -4\sin^2 \frac{\alpha}{2}$ and $e^{i\beta} - 2 + e^{-i\beta} = -4\sin^2 \frac{\beta}{2}$, we obtain the amplification factor

$$\xi = \frac{1}{1 + 4\lambda \sin^2 \frac{\alpha}{2} + 4\lambda \sin^2 \frac{\beta}{2}} \tag{16}$$

Based on equation (16), it can be said that the implicit finite difference scheme to 2D PME is unconditionally stable with $0 < \xi \leq 1$ for all positive values of λ and for $\alpha, \beta \in [-\pi, \pi]$.

4 Derivation of the four-point NEGMSOR iterative method

In the context of reducing the computational complexity, the convergence rate of the solution process needs to be increased. There are several iterative methods that have been proposed in order to reduce the computational complexity and this paper considers the application of the Explicit Group (EG) method which was introduced by Evans [8]. This method uses the concept of breaking the original matrix into several small fixed-size groups of points. By this way, the computational complexity can be reduced by solving several small groups of points instead of a large linear system per iteration. The EG method has been used extensively together with some efficient iterative methods such as SOR, MSOR and AOR [13-15].

Before the EG method is applied, equation (17) can be transformed into a linear system by using the Newton method as follows:

$$J_F \begin{pmatrix} U^{(\ell)} \\ \sim_{n+1} \end{pmatrix} \Delta H^{(\ell)} \sim_{n+1} = -F \begin{pmatrix} U^{(\ell)} \\ \sim_{n+1} \end{pmatrix}, \ell = 1, 2, \dots \tag{17}$$

where $J_F \begin{pmatrix} U^{(\ell)} \\ \sim_{n+1} \end{pmatrix}$ is a penta-diagonal Jacobian matrix with a dimension of

$(M-1)^2 \times (M-1)^2$, $\Delta H^{(\ell)} \sim_{n+1}$ and $F \begin{pmatrix} U^{(\ell)} \\ \sim_{n+1} \end{pmatrix}$ are the column matrices with a dimension of

$(M-1)^2 \times 1$, and for the iterative index, we denote it as (ℓ) and $n+1$ is considered to be the next time level. Then, equation (17) is divided into several completed groups of four points and some ungroup that can be treated as a group of two points and/or single point [8]. Based on any completed group of four grid points, the general form of the EG iterative method is given by

$$\begin{bmatrix} \Delta h_{i,j} \\ \Delta h_{i+1,j} \\ \Delta h_{i,j+1} \\ \Delta h_{i+1,j+1} \end{bmatrix}^{(\ell+1)} = \begin{bmatrix} c_{i,j} & d_{i,j} & & e_{i+1,j} \\ b_{i+1,j} & c_{i+1,j} & d_{i-1,j+1} & \\ & b_{i+2,j} & c_{i,j+1} & d_{i,j+1} \\ a_{i,j+1} & & b_{i+1,j+1} & c_{i+1,j+1} \end{bmatrix}^{-1} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \tag{18}$$

where

$$\begin{aligned} S_1 &= f_{i,j} - a_{i,j} \Delta h_{i,j-1}^{(\ell)} - b_{i,j} \Delta h_{i-1,j}^{(\ell)}, \quad S_2 = f_{i+1,j} - a_{i+1,j} \Delta h_{i+1,j-1}^{(\ell)} - e_{i+1,j} \Delta h_{i+1,j+1}^{(\ell)} \\ S_3 &= f_{i,j+1} - a_{i,j+1} \Delta h_{i,j}^{(\ell)} - e_{i,j+1} \Delta h_{i,j+2}^{(\ell)}, \\ S_4 &= f_{i+1,j+1} - d_{i+1,j+1} \Delta h_{i+2,j+1}^{(\ell)} - e_{i+1,j+1} \Delta h_{i+1,j+2}^{(\ell)}. \end{aligned}$$

Besides that, Kincaid and Young [16] introduced Modified Successive Over-Relaxation (MSOR) method which can be considered as a point iterative method that has two relaxation parameters. The basic idea of the MSOR method is to associate the different parameter with each row and column of a sparse and large linear system. Therefore, to derive the NEGMSOR iterative method, two different relaxation parameters, $1 \leq \omega_1 < 2$ and $1 \leq \omega_2 < 2$, are added to equation (18) and forms

$$\begin{bmatrix} \Delta h_{i,j} \\ \Delta h_{i+1,j} \\ \Delta h_{i,j+1} \\ \Delta h_{i+1,j+1} \end{bmatrix}^{(\ell+1)} = (1-\omega_1) \begin{bmatrix} \Delta h_{i,j} \\ \Delta h_{i+1,j} \\ \Delta h_{i,j+1} \\ \Delta h_{i+1,j+1} \end{bmatrix}^{(\ell)} + \omega_1 \begin{bmatrix} c_{i,j} & d_{i,j} & & e_{i+1,j} \\ b_{i+1,j} & c_{i+1,j} & d_{i-1,j+1} & \\ & b_{i+2,j} & c_{i,j+1} & d_{i,j+1} \\ a_{i,j+1} & & b_{i+1,j+1} & c_{i+1,j+1} \end{bmatrix}^{-1} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \tag{19}$$

$$\begin{bmatrix} \Delta h_{i,j} \\ \Delta h_{i+1,j} \\ \Delta h_{i,j+1} \\ \Delta h_{i+1,j+1} \end{bmatrix}^{(\ell+1)} = (1-\omega_2) \begin{bmatrix} \Delta h_{i,j} \\ \Delta h_{i+1,j} \\ \Delta h_{i,j+1} \\ \Delta h_{i+1,j+1} \end{bmatrix}^{(\ell)} + \omega_2 \begin{bmatrix} c_{i,j} & d_{i,j} & & e_{i+1,j} \\ b_{i+1,j} & c_{i+1,j} & d_{i-1,j+1} & \\ & b_{i+2,j} & c_{i,j+1} & d_{i,j+1} \\ a_{i,j+1} & & b_{i+1,j+1} & c_{i+1,j+1} \end{bmatrix}^{-1} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \end{bmatrix} \tag{20}$$

Based on equations (19) and (20), the NEG and NEGSOR iterative methods can be obtained by choosing $\omega_1 = \omega_2 = 1$ and $\omega_1 = \omega_2$ within the range of $[1, 2)$ respectively. The implementation of the NEGMSOR iterative method can be shown in Figure 1.

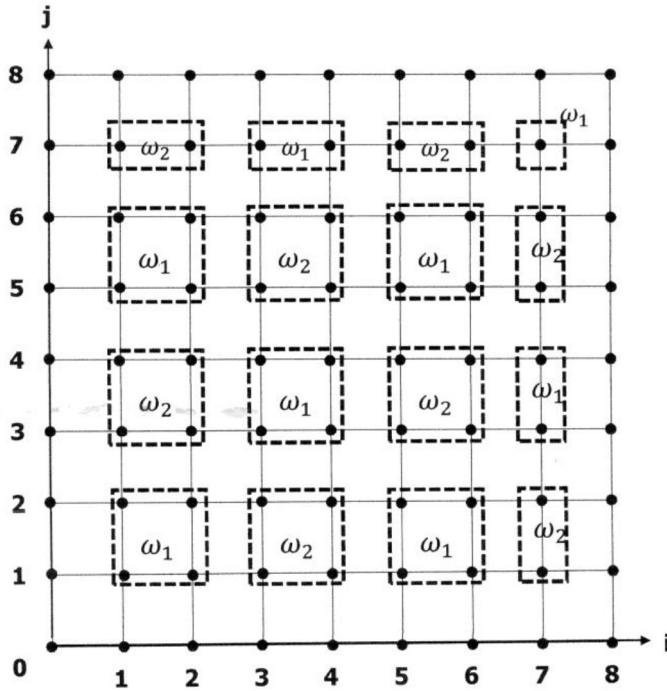


Figure 1. The implementation of NEGMSOR iterative method.

By referring to Figure 1, the numerical solution of 2D PME by using NEGMSOR iterative method can be summarized into Algorithm 1. The optimum values of ω_1 and ω_2 are determined when the least number of iterations is obtained after a few executions. During the executions, we consider a tolerance error of $\varepsilon = 1 \times 10^{-10}$.

Algorithm 1: NEGMSOR iterative method

- i. Set up the values for A_1, A_2, A_3 , and A_4 ,
- ii. Set the initial guess $U_{\sim n+1}^{(0)} = 1.0$,
- iii. Formulation of the square matrix $J_F \left(U_{\sim n+1}^{(\ell)} \right)$ and the column matrix $-F \left(U_{\sim n+1}^{(\ell)} \right)$,
- iv. Iteration of equations (19) and (20),
- v. Convergence test 1: if $\left| \Delta H_{\sim n+1}^{(\ell+1)} - \Delta H_{\sim n+1}^{(\ell)} \right| < \varepsilon$, then $U_{\sim n+1}^{(\ell+1)} = \Delta H_{\sim n+1}^{(\ell+1)} + U_{\sim n+1}^{(\ell)}$,
- vi. Convergence test 2: if $\left| U_{\sim n+1}^{(\ell+1)} - U_{\sim n+1}^{(\ell)} \right| < \varepsilon$, then go to the next time step,
- vii. Display the approximate solutions.

5 Numerical experiments

As a comparative analysis, three examples of 2D PME are used. The efficiency of the NEGMSOR iterative method is investigated and is compared to the NGS, NEG and NEGSOR iterative methods. Number of iterations, execution time in seconds and maximum absolute error (error) become the subjects of comparison.

Example 1 [17]:

$$\frac{\partial u}{\partial t} = 0.2 \left[\frac{\partial}{\partial x} \left(u \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(u \frac{\partial u}{\partial y} \right) \right] \tag{21}$$

with exact solution $u(x, y, t) = x + y + (0.4)t$.

Example 2 [17]:

$$\frac{\partial u}{\partial t} = 0.2 \left[\frac{\partial}{\partial x} \left(u^2 \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(u^2 \frac{\partial u}{\partial y} \right) \right] \tag{22}$$

with exact solution $u(x, y, t) = (5(x + y + t))^{1/2}$.

Example 3 [18]:

$$\frac{\partial u}{\partial t} = \frac{\partial}{\partial x} \left(u^5 \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left(u^5 \frac{\partial u}{\partial y} \right) \tag{23}$$

with exact solution $u(x, y, t) = (4(x + y + 2t)/5)^{1/4}$.

In the numerical experiments, we consider five different orders of the linear system, $(M - 1)$ in which the different values of mesh size, M are 16, 32, 64, 128 and 256. The numerical results are indicated in Table 1 to 3 and the comparative analysis between the NEGMSOR iteration family against the NGS methods is done by calculating the reduction percentage in the number of iterations and the computation time as follows:

Against other iterative methods:

$$(\Delta \ell)\% = \left| \frac{\ell_{NGS} - \ell_{NEGMSOR}}{\ell_{NGS}} \right| \times 100\%, \quad (\Delta s)\% = \left| \frac{s_{NGS} - s_{NEGMSOR}}{s_{NGS}} \right| \times 100\% \tag{24}$$

where $(\Delta \ell)$ and (Δs) are reduction percentage of number of iterations and execution time for NEGMSOR method respectively.

Table 1. Comparison in terms of number of iterations based on all iterative methods for three examples.

Example	Method	Mesh Size				
		16	32	64	128	256
1	NGS	136	436	1525	5462	19404
	NEG	84	244	818	2903	10338
	NEGSOR	64	131	262	523	1020
	NEGMSOR	54	102	203	399	787
2	NGS	130	440	1380	4901	17458
	NEG	80	223	738	2596	9279
	NEGSOR	62	124	255	505	992
	NEGMSOR	58	104	202	394	779
3	NGS	739	2630	9478	34098	121649
	NEG	406	1393	4996	18034	64894
	NEGSOR	199	395	780	1525	2971
	NEGMSOR	181	349	680	1333	2616

Table 2. Comparison in terms of execution time (seconds) based on all iterative methods for three examples.

Example	Method	Mesh Size				
		16	32	64	128	256
1	NGS	0.76	2.93	19.59	360.89	4586.69
	NEG	0.44	2.64	8.55	214.01	3717.50
	NEGSOR	0.33	2.05	6.37	62.79	390.80
	NEGMSOR	0.29	1.73	5.60	40.01	263.99
2	NGS	0.97	2.70	18.26	248.82	4243.79
	NEG	0.62	1.93	8.56	187.24	2295.15
	NEGSOR	0.43	1.53	5.35	50.98	369.60
	NEGMSOR	0.39	1.28	4.90	38.32	225.95
3	NGS	0.98	9.51	113.63	1653.85	29234.80
	NEG	0.86	7.75	61.96	763.70	14890.79
	NEGSOR	0.59	2.05	12.89	118.29	921.40
	NEGMSOR	0.57	1.96	10.21	96.27	739.37

Table 3. Comparison in terms of maximum absolute error based on all iterative methods for three examples.

Example	Method	Mesh Size				
		16	32	64	128	256
1	NGS	8.86e-11	3.25e-10	1.90e-09	9.02e-09	3.86e-08
	NEG	1.05e-11	1.36e-10	6.59e-10	3.96e-09	1.80e-08
	NEGSOR	1.27e-12	7.44e-12	1.62e-11	2.60e-11	3.82e-11
	NEGMSOR	1.26e-12	7.98e-12	1.32e-11	2.90e-11	4.12e-11
2	NGS	7.57e-11	2.31e-09	1.31e-08	4.95e-08	1.75e-07
	NEG	2.09e-11	8.45e-10	6.39e-09	2.79e-08	9.92e-08
	NEGSOR	8.52e-13	2.11e-11	5.45e-11	3.69e-10	1.28e-09
	NEGMSOR	1.07e-12	9.03e-12	1.93e-11	3.91e-11	5.67e-11
3	NGS	1.10e-09	6.69e-09	3.56e-08	1.72e-07	7.78e-07
	NEG	1.55e-10	2.50e-09	1.51e-08	7.45e-08	3.56e-07
	NEGSOR	1.43e-11	2.89e-11	3.00e-11	4.06e-11	4.91e-11
	NEGMSOR	5.58e-12	3.34e-11	4.36e-11	3.96e-11	4.86e-11

From the numerical experiment results in Table 1-3, the reduction percentage of the NEGMSOR iteration family as compared the GS iterative method for three examples are summarized in Table 4.

Table 4. Reduction percentage of the number of iterations and the execution time for NMEGSOR iteration family compared to NGS method

Example	Methods	Number of Iterations (%)	Execution Time (%)
1	NEG	38.23-46.85%	9.89-56.36%
	NEGSOR	52.94-94.74%	30.03-91.48%
	NEGMSOR	60.29-95.94%	40.96-94.24%
2	NEG	38.46-49.32%	24.75-53.12%
	NEGSOR	52.31-94.32%	43.33-91.29%
	NEGMSOR	55.38-95.54%	52.59-94.68%
3	NEG	45.06-47.03%	12.24-53.82%
	NEGSOR	73.07-91.77%	39.80-96.84%
	NEGMSOR	75.51-96.09%	41.84-97.47%

Referring to the results of numerical experiments for three tested examples in Table 4, it can be pointed out that the percentage reduction of number of iterations for the NEGMSOR iterative method have declined approximately by 60.29-95.94%, 55.38-95.54%, and 75.51-96.09% respectively as compared with the NGS method. In fact, implementations of execution time for NEGMSOR iterative method are much faster about 40.96-94.24%, 52.59-94.68%, and 41.84-97.47% respectively than the NGS method.

6 Conclusion

This paper proposed the unconditionally stable implicit difference scheme together with the NEGMSOR iterative method for solving several examples of 2D PME. The implicit finite difference scheme is successfully applied to discretize the general form of 2D PME. In addition to that, the sparse and large linear system is solved using the NEGMSOR iterative method and the numerical results reveal that the proposed method is more efficient in terms of number of iterations, and execution time as compared to the tested NGS, NEG and NEGSOR iterative methods. Overall, in term of and the magnitude of maximum absolute error, the accuracy of the proposed NEGMSOR iterative method and the three tested numerical methods are comparable.

The authors gratefully acknowledge that this paper was partially supported by the research grant scheme (GUG0022-SG-M-1/2016) from Universiti Malaysia Sabah, Malaysia..

References

1. J.L. Vazquez. *The Mathematical Theories of Diffusion: Nonlinear and Fractional Diffusion*. Lecture Notes in Mathematics, 2186 (2016).
2. P.B. Patil, U.P. Verma. *Numerical Computational Methods*. Revised Ed. Oxford: Alpha Science International Ltd. (2009)
3. R.N. Borana, V.H. Pradhan, M.N. Mehta. Intern. J. of Innovative Research in Science, Engineering and Technology. **3**, 2 (2014)
4. V.H. Pradhan, M.N. Mehta, T. Patel. Intern. J. of Advanced Engineering Technology **2**, 1 (2011).
5. C. Ngo, W.Z. Huang. J. Comp. Physics. **331** (2017)
6. J.V.L. Chew. J. Sulaiman. *The 2nd International Conference on Mathematical Sciences and Statistics (ICMSS2016)* AIP Conf. Proc. **1739** (2016).
7. J.V.L. Chew. J. Sulaiman. Journal of Applied Mathematics and Computational Mechanics **15**, 2 (2016)
8. D.J. Evans. 1985. Intern. J. Comp Math. **17** (1985)
9. R.D. Richtmyer, K.W. Morton. *Difference methods for initial-value problems*. 2nd ed. New York: John Wiley & Sons, Inc. (1967)
10. B.F. Feng, T. Mitsui. J. Computational and Applied Mathematics. **90**, (1998)
11. D. Kim, W. Proskurowski. 2004. Intern. J. Math. & Mathematical Sciences. **17**, (2004)
12. J. Strikwerda. 2004. Finite difference schemes and partial differential equations. Philadelphia: SIAM. (2004)
13. J. Sulaiman, M.K. Hasan, M. Othman, S.A.A. Karim. Journal of Mathematics and System Science. **2**, (2012)
14. A. Saudi, J. Sulaiman. International Journal of Computer Applications. **53**, 16 (2012)

15. M.K.M. Akhir, J. Sulaiman. *Applied Mathematical Sciences*. **9**, 112 (2015)
16. D.R. Kincaid, D.M. Young. 1972. *Mathematics of Computation*. **26**, 119 (1972)
17. A.D. Polyanin, V.F. Zaitsev. *Handbook of Nonlinear Partial Differential Equations*. Boca Raton: Chapman and Hall/CRC Press. (2004)
18. B.P. Sommeijer, P.J. van der Houwen. *ACM Transactions on Mathematical Software*. **10**, 4 (1984).