PENERBIT UMT
Menjana Khazanah Ilmuwan

Journal of Mathematical Sciences and Informatics
Journal Homepage: https://journal.umt.edu.my/index.php/jmsi
eISSN: 2948-3697
DOI: http://doi.org/10.46754/jmsi.2021.12.002

JMSI
Journal of Mathematical Sciences and Informatics

HYBRID FDM-STABILIZED LANCZOS-TYPE IN SOLVING PDE PROBLEMS

REHANA THALIB[1], MAHARANI A. BAKAR[2]*, SEE G. BIN[1], LEE Y. YEE[1], YONG S. XIA[1] AND NORIZAN MOHAMED[1]

[1]Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Malaysia. p3720@pps.umt.edu.my, gbsee0511@gmail.com, yeetwin97@gmail.com, aprilyong97@gmail.com, norizan@umt.edu.my. [2]Special Interest Group Modelling and Data Analytics, Faculty of Ocean Engineering Technology and Informatics, Universiti Malaysia Terengganu, Malaysia. maharani@umt.edu.my.

Corresponding author: maharani@umt.edu.my

ARTICLE INFO

Article History:
Received June 2021
Accepted July2021
Available online December 2021

Keywords:
Finite difference;
Lanczos solvers;
PDEs;
SLEs

ABSTRACT

This study investigates the combination of finite difference method (FDM) and the stabilized Lanczos method to solve various partial differential equation (PDE) problems. This combination is wrapped in the algorithms called hybrid FDM-RMEIEMLA and hybrid FDM-RLMinRes. FDM is the discretization method which converts the PDEs into algebraic formula, whereas both RMEIEMLA and RLMinRes are known as the stabilized Lanczos methods in solving large-scale problems of SLEs. Their hybrids enable us to find the solutions of PDE problems accurately. There are at least three types of PDEs solved in this study, namely Helmholtz, wave, and heat equations. The convergence rate of our methods computed using the residual norms $\| \mathbf{b} - A\mathbf{x}_k \|$. Numerical results showed that our proposed methods performed well in solving the various PDEs with small residual norms.

2020 Mathematics Subject Classification:
2020 ACM Computing Classification Codes:

INTRODUCTION

Partial Differential Equation (PDE) is a mathematical equation that consists of multivariable function and partial derivatives. The application PDEs can be found in several subjects, such as biological sciences, engineering, and physical problems [1]. PDEs are split into three general categories, they are hyperbolic, parabolic, and elliptic. There are several equations based on those categories, for instance, the Laplace and Poisson equations follow the elliptic PDE, heat equation and wave equation represent the parabolic and hyperbolic PDEs, respectively [2].

There are several numerical methods to solve PDEs, one of them is known as finite difference method (FDM), which typically discretizes the PDEs into the algebraic system of linear equations (SLEs), which is then solved numerically using an iterative solver [2]. Traditionally, the FDM can be combined with stationary methods, such as Gauss-Seidel and SOR to handle the SLEs part [2]. As we know that the stationary methods are only effective if the matrix of the SLEs is symmetric, otherwise alternative methods are needed. In this study, we employ the stabilized Lanczos methods to be combined with FDM to solve the PDE problems. Lanczos Method, discovered by Cornelius Lanczos in 1950, was earlier employed to solve the eigen problems [3], before it was then developed to solve SLEs [4]. Particularly, Lanczos method can be efficiently used to solve high dimensional non-symmetric SLEs. In the 1990s, Lanczos method was developed, which led to several formulas, called Lanczos-based formula [5], by applying the theory of Formal Orthogonal Polynomial (FOPs) into the recurrence formula of orthogonal polynomials. There are at least two new classes based on those formulas, namely Baheux-types [6] and Farooq-types [7]. They were all implemented in the form of Lanczos-type algorithms.

All Lanczos-based formulas experience breakdown which causes the algorithms to halt before converging. It is not our focus here to discuss the breakdown, however, the readers can refer [8, 9] to find out more about the issue of breakdown. In relation to combating the breakdown, several authors have investigated some strategies. Amongst them, we are interested in investigating more deeply two algorithms called RLMinRes and RMEIEMLA, which applied the restarting strategy in Lanczos-based formula. The early algorithm is the Lanczos-types combined with a restarting strategy with focus on using quality points to start with, particularly using the iterate with the minimum residual norm [10]. The latter algorithm is a Lanczos-based model combined with restarting strategy. The model is based on interpolation and extrapolation to predict the new solution when the breakdown occurs [11, 12]. These two methods are called stabilized Lanczos-type [13], which are one of the latest strategies to combat breakdown and improve the convergence rates of Lanczos-type algorithms. These two methods are combined with the FDM to make the method applicable to solve PDE problems.

PRELIMINARY

A partial differential equation (PDE) is defined as a differential equation that involves partial derivatives of the dependent variables, called unknown function, in more than one variable *x,y,z* and *t*. Linearity of a PDE is defined as the unknown function and its derivative entering the equation linearly [14]. A PDE solution is a function that depends on time and space or only space.

Definition 2.1: The general formula of the second order PDE for two-dimensional space is as follows:

$$a\frac{\partial^2 u}{\partial x^2} + 2b\frac{\partial^2 u}{\partial x \partial y} + c\frac{\partial^2 u}{\partial y^2} + d_1\frac{\partial u}{\partial x} + d_2\frac{\partial u}{\partial y} + e = u(x,y) \qquad (1)$$

where a,b,c,d_1,d_2, and e are some constants. If $b^2 - ac > 0$, the PDE (1) is called hyperbolic. If $b^2 - ac = 0$, it is called parabolic, and is called elliptic if $b^2 - ac < 0$[15]. There are several phenomena in the real problems that can be described as those types of PDEs. In this study, we discuss some equations which can be included in one of the PDE types.

Helmholtz Equation

The Helmholtz Equation is widely used in mathematics and physics. This equation is named after German physician Hermann von Helmholtz, who was an expert in psychology, physiology, and physics. He made a lot of contributions in several scientific areas [16]. Helmholtz Equation is a second order linear PDE and can be expressed as the general form as follows:

Definition 2.2: The general form of second order Helmholtz Equation:

$$\nabla^2 u(x,y) + s^2 u(x,y) = 0 \qquad (2)$$

where $\Delta=\nabla^2$ is the Laplacian, s is a scalar wave number and $u(x,y)$ is the corresponding solution for PDE (2). The Helmholtz Equation can also be derived through the general wave equation as Definition 2.3 below.

Definition 2.3: General wave equation:

$$\frac{\partial^2 w}{\partial t^2} = \alpha^2 \Delta w \qquad (3)$$

which shows the propagation of the wave travelling through a medium with a constant speed, α, while the Laplacian, Δ gives out a two-dimensional space where $\Delta = (\partial^2/\partial x^2) + (\partial^2/\partial y^2)$. Suppose that the solution is separable, we know that $w(x,y,t)=u(x,y)v(t)$ as u and v are independent, and the function of (x,y) and (t) only, respectively. By substituting and rewriting them into (3), we have $(1/\alpha^2 v)(\partial^2 v/\partial t^2)=\Delta u/u$, then suppose that both sides are equal to a constant $-s^2$, then we have $\partial^2 v/\partial t^2+\alpha^2 s^2 v=0$. For function u, this gives us the Helmholtz Equation (4) below.

Definition 2.4: Helmholtz equation:

$$\Delta^2 u + s^2 u = 0 \quad (4)$$

Wave Equation

Wave equation is an example of a hyperbolic partial differential equation. There are several fields of study such as dynamics of the population, tides and waves and nuclear reactors, which arose by hyperbolic PDE [17]. Hyperbolic equations can be used for the simulation of sound waves, electromagnetic waves, seismic waves, shock waves, and many other waveforms [15]. Wave equation is the simplest form of the one-dimensional hyperbolic equation. The wave equation, as written in (5), is important in mechanics. This goes on to describe the strings and wires motion and the flow of waves of water [18].

Definition 2.5: One dimensional wave equation:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \quad (5)$$

Heat Equation

The heat equation is one of the general examples of a parabolic equation. For parabolic PDEs, numerical solution methods are important in areas such as heat transfer, nuclear reactor analysis, molecular diffusion, and fluid flow [19]. Heat equation (6) is the simplest form of the one-dimensional parabolic PDE [20]. The heat equation was first implemented by Jean-Baptiste Joseph Fourier (1768-1830) as the mathematical interpretation of heat flow in solids [21]. The heat equation could be generated from Fourier's principle as well as energy conservation. Across numerous fields of science, the heat equation seems to be essential. In mathematics, the heat equation has been related to Brownian motion via the Fokker-Planck equation [22].

Definition 2.6: One dimensional heat equation:

$$\frac{\partial u}{\partial t} = k \frac{\partial^2 u}{\partial x^2} \quad (6)$$

where k is a constant.

MATERIALS AND METHODS

Methods to treat the instability of Lanczos-type algorithms have been studied and found in [7]. One of them is the restarting strategy. As its name suggests, restarting enables us to restart the algorithm before it faces breakdown. In this paper, we used two recent proven restarting strategies, namely restarting from the iterate with the minimum residual norm (RLMinRes) and restarting modified embedding interpolation and extrapolation model in Lanczos-types Algorithms (RMEIEMLA).

Restarting Lanczos-type Algorithms from the Iterate with the Minimum Residual Norm (RLMinRes)

This algorithm was proposed in [10] as one of the stabilized Lanczos-types for solving large dimensions of SLEs. It has been numerically proven more robust than the previous restarting Lanczos-type investigated in [23]. The idea of RLMinRes is that there are some quality points that can be considered as a restarting point, one of them is the iterate with the minimum residual norm. Thus, the algorithm is called RLMinRes, [10]. Basically, after running a Lanczos-type for iterations (or after the breakdown), we stop the algorithm and collect all the sequences of iterates generated by the Lanczos algorithm. We then pick one iterate in the sequence which corresponds to the lowest residual norm and initialize the Lanczos-type algorithm by using this iterate. These procedures are formalized in the following algorithm.

Algorithm 1: RLMinRes [10]

1. Fix the number of iterations to say, k and the tolerance, ε, to 1E-13 and run a Lanczos-type algorithm [10]
2. Initialization. Choose $\boldsymbol{x}_0$ and $\mathbf{y}_0$. Set $\mathbf{x}=\mathbf{x}_0$, $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$,$\mathbf{y}_0= \mathbf{y}$, and $\mathbf{z}_0= \mathbf{r}_0$
3. Collect all k vector solutions and save them in $S = \{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_m,\ldots,\mathbf{x}_k\}$
4. Collect all of the residual norms and save them in $R=\{\|\boldsymbol{r}_1\|.\|\boldsymbol{r}_2\|\ldots,\|\boldsymbol{r}_m\|,\ldots,\|\boldsymbol{r}_k\|\}$
5. Compute the minimum values and specify the index of the minimum value as
6. Obtain the approximate solution as well as the residual norm as follows

   $sol_{min}= \mathbf{x}_\mathbf{m}$

   $norm_{min} = \|\mathbf{r}_\mathbf{m}\|$
7. **While** $norm_{min} \geq \varepsilon$ **do**

   $\mathbf{x} = sol_{min}$

   $\mathbf{y} = \mathbf{b} - A\mathbf{x}$
8. Run LMinRes [10] for iterations
9. **end while**
10. Take sol_{min} as the approximate solution
11. Stop

The value of k in Algorithm 1 above is a fixed maximum value to control our iteration or as reasonable stopping criterion if the method does not converge. In this study, we use $k = 100$ iterations that make up one cycle. While the is an index that stores minimum value of our solution ($\mathbf{x}_\mathbf{m}$) and minimum value of residual norm ($\|\mathbf{r}_\mathbf{m}\|$), which is the goal of this algorithm to use it as a restarting point.

Restarting Modified Embedding Interpolation and Extrapolation Model in Lanczos-types Algorithms (RMEIEMLA)

RMEIEMLA was studied in [11] as the modification of [12]. Different from RLMinRes, RMEIEMLA not only used a restarting approach, but also predicting the solution generated by Lanczos-type algorithms. On RMEIEMLA, PCHIP (piecewise cubic Hermite interpolation polynomial) is used to interpolate the entries of the solutions generated by the Lanczos-type algorithms. Then, we extrapolate it to get a new approximate solution which gives us a better solution as our restarting point. The algorithm of RMEIEMLA is presented in Algorithm 2 below.

Algorithm 2: RMEIEMLA [12]

1. Fix the number of iterations to say, k and the tolerance, ε, to 1E-13 and run a Lanczos-type algorithm [10]
2. Initialization. Choose $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$. Set $\mathbf{x}=\mathbf{x}_0$, $\mathbf{r}_0 = \mathbf{b} - \mathrm{A}\mathbf{x}_0$,$\mathbf{y}_0= \mathbf{y}$, and $\mathbf{z}_0= \mathbf{r}_0$
3. **if** ,$\|\mathbf{r}_\mathbf{k}\| \leq \varepsilon$ **then**
4. The solution obtained
5. Stop
6. **else**
7. Collect all k vector solutions and save them in $S = \{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_m, ...,\mathbf{x}_k\}$
8. Choose some j such that $m - j \leq k$

9. Set $W = \{(t_{m-j}, \mathbf{x}_{m-j}),(t_{m-j+1}, \mathbf{x}_{m-j+1}),...,(t_k, \mathbf{x}_k)\}$
10. Interpolate W using PCHIP to get f
11. Choose $t^* \in [m,s] \subset R$, where $s \geq m \geq k$ is an integer, and calculate $\mathbf{x}^* = f(t^*)$,
12. Calculate the residual norms as follows $\|\mathbf{r}^*\| = \|\mathbf{b}\text{-}A\mathbf{x}^*\|$
13. **end if**
14. Obtain a sequence of iterates $\{\mathbf{x}_{k+1}, \mathbf{x}_{k+2},...,\mathbf{x}_m, ...,\mathbf{x}_s\}$, where $s \geq k+1$, and calculate the residual norms of these iterates
15. $sol_{model} = \mathbf{x}_{model}$
16. $norm_{model} = \|r_{model}\|$
17. **While** $norm_{model} \geq \varepsilon$ **do**

    $\mathbf{x} = sol_{model}$,

    $\mathbf{y} = \mathbf{b}\text{-}A\mathbf{x}$
18. end while
19. Take sol_{model} as the approximate solution
20. Stop

The sol_{model} is a variable that stores our predicted approximate solution and then uses it as a restarting point.

RESULTS AND DISCUSSION

Hybrid Model FDM-RLMinRes and FDM-RMEIEMLA for Solving PDE Problems

We combine the procedure of finite difference method (FDM) and stabilized Lanczos-types to solve PDE problems. FDM is a classical method to discretize PDE form into the algebraic system by using the finite difference formula. In this study, we use explicit FDM, the reason is as an example to show how the FDM-stabilized Lanczos-type algorithm can work for PDEs problems. Furthermore, we choose explicit FDM since it is simpler and easier than implicit FDM. In fact, our proposed method can be done for PDE problems by using implicit FDM, too, and this can be analysed more for future study. To achieve our goal, this study has two general stages, which discretizes the PDEs problem to the system of linear equations (SLEs), then solves the SLEs by using the stabilized Lanczos methods.

Discretization Form of Helmholtz Equation

Recall Helmholtz equation as in Definition 2.4 and rewrite it into $\Delta^2 u(x,y) + s^2 u(x,y) = 0$, with boundary conditions $u(x,0) = f(x)$, $u(0,y) = 0$, and $u_x(0,y) = g(x)$, where $0 < x < l$, $0 < y < Y$ while l and Y are the endpoints of x and y respectively. The discretization process of the Helmholtz equation is explained in Definition 4.1 below.

Definition 4.1: Finite difference method of Helmholtz equation:

$$\frac{\partial^2 u(x,y)}{\partial x^2} + \frac{\partial^2 u(x,y)}{\partial y^2} + s^2 u(x,y) = 0 \tag{7}$$

By applying central difference formula of $\frac{\partial^2 u(x,y)}{\partial x^2}$ and $\frac{\partial^2 u(x,y)}{\partial y^2}$ to equation (7) we get

$$\frac{u(x+h,y)-2u(x,y)+u(x-h,y)}{h^2} + \frac{u(x,y+k)-2u(x,y)+u(x,y-k)}{k^2} + s^2 u(x,y) = 0 \tag{8}$$

where h is the step size for x and k is the step size for y. Besides, this equation can be also be written in terms of i and j (9), where i and j are the number of steps for x and j respectively.

$$\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{h^2} + \frac{u_{i,j+1}-2u_{i,j}+u_{i,j-1}}{k^2} + s^2 u_{i,j} = 0. \tag{9}$$

Then set $r = k^2/h^2$, rewrite equation (9) in the below form.

$$r(u_{i+1,j} + u_{i-1,j}) + u_{i,j}(k^2s^2 - 2r - 2) + u_{i,j-1} = -u_{i,j+1}. \tag{10}$$

Let $z=k^2s^2-2r-2$, and by dividing equation (10) with z we get general form of discretized Helmholtz equation as equation (11) as follows

$$-\frac{1}{z}u_{i,j+1} - \frac{r}{z}(u_{i+1,j} + u_{i-1,j}) = u_{i,j} + \frac{1}{z}u_{i,j-1} \tag{11}$$

Discretization Form of Wave Equation

The one-dimensional wave equation is $u_{tt}(x,t) = \alpha^2[u_{tt}(x,t)]$ with boundary conditions $u(x,0) = g(x)$ and $u_t(x,0) = h(x)$, where $0 \le x \le L, 0 \le t \le T$. The discretization process of wave equation is explained in Definition 4.2 below.

Definition 4.2: Finite difference method of wave equation:

Choose an integer $n > 0$ to define the points on the x-axis using $h = L/n$ and we choose a time-step size $l > 0$. The grid points (x_j, t_k) are defined by $x_j = jh$ and $t_k = kl$, for each $j = 0, 1,..., n$, and $k = 0, 1,..., n$. At any interior grid point (x_j, t_k), the wave equation becomes

$$u_{tt}(x_j, t_k) = \alpha^2[u_{xx}(x_j, t_k)] \tag{12}$$

For a short writing, we represent the approximation solution $u(x_j, t_k)$ by w_{jk}. To discretize the above wave equation (12) at the grid point (x_j, t_k), we approximate the second partial derivatives by central differences formula, to get

$$w_{j,k+1} - 2w_{j,k} + w_{j,k-1} = \frac{l^2\alpha^2}{h^2}(w_{j+1,k} - 2w_{j,k} + w_{j-1,k}) \tag{13}$$

Let $\mu = l\alpha/h$, then the general discretized form of wave equation is as follows

$$w_{j,k+1} - 2(1-\mu^2)w_{j,k} - \mu^2(w_{j+1,k} + w_{j-1,k}) + w_{j,k-1} = 0 \tag{14}$$

Discretization Form of Heat Equation

The one-dimensional heat equation is $\frac{\partial u}{\partial t}(x,t) = \alpha^2 \frac{\partial^2 u}{\partial x^2}(x,t)$, with $0 \le x \le l;\ t > 0$. Moreover, the boundary conditions are $u(0,t) = u(l,t) = 0$ and $u(x, 0) = f(x)$, where $t > 0$ and $0 \le x \le l$. The discretization process of the heat equation is explained in Definition 4.3 below.

Definition 4.3: Finite difference method of heat equation:

Choose an integer $m > 0$ to be the total number of steps in the x direction and define $h = l/m$, then choose a step size in t direction, $k = T/N$. The grid points for , where for (v_a, t_b) where $v_a = ah$ *for* $a= 0,1,2, \dots, m$ and $t_b=bk$ for $b =0,1,2, \dots$. By using Taylor series in t to form the difference quotient

$$\frac{\partial u}{\partial t}(v_a, t_b) = \frac{u(v_a,t_b+k)-u(v_a,t_b)}{k} \tag{15}$$

and in x to form the difference quotient

$$\frac{\partial^2 u}{\partial v^2}(v_a, t_b) = \frac{u(v_a+h,t_b)-2u(v_a,t_b)+u(v_a-h,t_b)}{h^2} \tag{16}$$

The parabolic PDE implies at the interior grid points (v_a, t_b), for each $a = 1,2,3,...,m\text{-}1$ and $b = 1,2,...$,we have

$$\frac{\partial u}{\partial t}(v,t) = \alpha^2 \frac{\partial^2 u}{\partial v^2}(v,t) = 0 \tag{17}$$

Then, substituting the equation above and rearrange for $w_{a,b+1}$ we get

$$w_{a,b+1} = \left(1 - \frac{2a^2 k}{h^2}\right) w_{a,b} + \alpha^2 \frac{k}{h^2}\left(w_{a+1,b} + w_{a-1,b}\right) \tag{18}$$

for each a =1,2,3,...,m-1 and b = 1,2,...

The Algorithms of Hybrid Model FDM-RLMinRes and FDM-RMEIEMLA for Solving PDE Problems

At this stage, all of the final equations in Definition 4.1, 4.2, and 4.3 are solved by using RMEIEMLA and RLMinRes algorithms as written in Algorithm 1 and Algorithm 2. The combination of FDM and the stabilized Lanczos algorithms is wrapped in an algorithm called hybrid FDM-RLMinRes and FDM-RMEIEMLA. They are presented in Algorithm 3 and Algorithm 4 respectively.

Algorithm 3: Hybrid FDM-RLMinRes

1. Set step sizes h and l with a suitable size as to decide the dimension for the SLEs
2. Discretize the PDEs into the algebraic system of linear equations (SLEs) by using FDM
3. Initialization. Choose $\boldsymbol{x}_0$ and $\mathbf{y}_0$. Set $\mathbf{x} = \mathbf{x}_0$, $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, $\mathbf{y}_0 = \mathbf{y}$, and $\mathbf{z}_0 = \mathbf{r}_0$
4. Fix the number of iterations to say, k and the tolerance, ε, to 1E-13
5. Run RLMinRes for k iterations
6. **While** $norm_{min} \geq \varepsilon$ **do**
7. Initialize the algorithm with
   $\mathbf{x} = sol_{min}$
   $\mathbf{y} = \mathbf{b} - A\mathbf{x}$
8. **end while**
9. Take sol_{min} as the approximate solution
10. Stop

Algorithm 4: Hybrid FDM-RMEIEMLA

1. Set step sizes h and l with a suitable size as to decide the dimension for the SLEs
2. Discretizes the PDEs into the algebraic system of linear equations (SLEs) by using FDM
3. Form a SLE for the equation obtained at step 2
4. Initialization. Choose $\mathbf{x}_0$ and $\mathbf{y}_0$. Set $\mathbf{x} = \mathbf{x}_0$, $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$, $\mathbf{y}_0 = \mathbf{y}$, and $\mathbf{z}_0 = \mathbf{r}_0$
5. Fix the number of iterations to say, k, and the tolerance, ε, to 1E-13
6. Run RMEIEMLA for k iterations
7. While $norm_{min} \geq \varepsilon$ **do**
8. Initialize the algorithm with
9. $\mathbf{x} = sol_{min}$
10. $\mathbf{y} = \mathbf{b} - A\mathbf{x}$
11. **end while**
12. Take sol_{min} as the approximate solution
13. Stop.

Numerical Results

We solved different sized problems of SLEs $A\mathbf{x} = \mathbf{b}$, with the matrix A of the system is obtained as a result in discretizing the three types of PDEs equations by using FDM with different step sizes. Then the SLEs is solved by Lanczos-type algorithms, in this case, we employed Orthodir algorithm which is one of the variants of Lanczos algorithms. The implementation of all algorithms of this study was done using Matlab under Windows. All of results of Hybrid FDM-RLMinRes and RMEIEMLA are presented in Table 1 for the Helmholtz equation, Table 2 for wave equation, and Table 3 for heat equation. The dimension is based on the step sizes and the end points of x- and y- directions for PDE Helmholtz, also step sizes, time steps and end points of x- directions and time step.

Table 1: Hybrid FDM-RLMinRes and Hybrid FDM-RMEIEMLA for solving PDE Helmholtz

Step Sizes		**Dimension**	**Residual Norms**	
h	***k***		**FDM-RMEIEMLA**	**FDM-RLMinRes**
0.001000	0.005	999	7.5992e-16	4.7434e-19
0.000100	0.001	1999	9.1676e-18	7.3598e-20
0.000400	0.001	2499	9.3529e-18	3.8711e-18
0.000250	0.001	3999	8.7072e-18	8.4504e-19
0.000200	0.001	4999	8.4967e-18	6.0423e-19
0.000125	0.001	7999	6.7339e-18	1.6175e-19
0.000010	0.001	9999	5.2278e-18	6.8628e-19
0.000020	0.001	19999	9.0806e-18	1.2857e-19

Table 2: Hybrid FDM-RLMinRes and Hybrid FDM-RMEIEMLA for solving PDE waves problems

Step Sizes		**Dimension**	**Residual Norms**	
h	***k***		**FDM-RMEIEMLA**	**FDM-RLMinRes**
0.0200	0.0200	2401	8.2950e-14	1.9796e-13
0.0143	0.0143	4761	8.4580e-14	1.1569e-13
0.0125	0.0125	6241	9.3549e-14	1.2750e-13
0.0111	0.0111	7921	9.6476e-14	1.0342e-13
0.0100	0.0100	9801	9.6648e-14	1.0718e-13
0.0077	0.0077	16641	1.8329e-13	7.3391e-13

Table 3: Hybrid FDM-RLMinRes and Hybrid FDM-RMEIEMLA for solving PDE heat problem

Step Sizes		**Dimension**	**Residual Norms**	
h	***k***		**FDM-RMEIEMLA**	**FDM-RLMinRes**
0.6061	0.6250	1024	5.0605e-16	3.9505e-17
0.3279	0.3333	3600	5.8988e-16	9.5831e-17
0.2817	0.2857	4900	6.6275e-16	5.5648e-16
0.2198	0.2222	8100	7.0633e-16	8.7181e-16
0.1980	0.2000	10000	9.7795e-16	9.3454e-16
0.1626	0.1639	14884	8.5897e-16	9.9966e-16

Moreover, we record and visualize both methods in Figure 1- Figure 3. Figures 1, 2, and 3 represent Helmholtz equation, wave equation, and heat equation, respectively, using both Hybrid FDM-RLMinRes and FDM-RMEIEMLA.

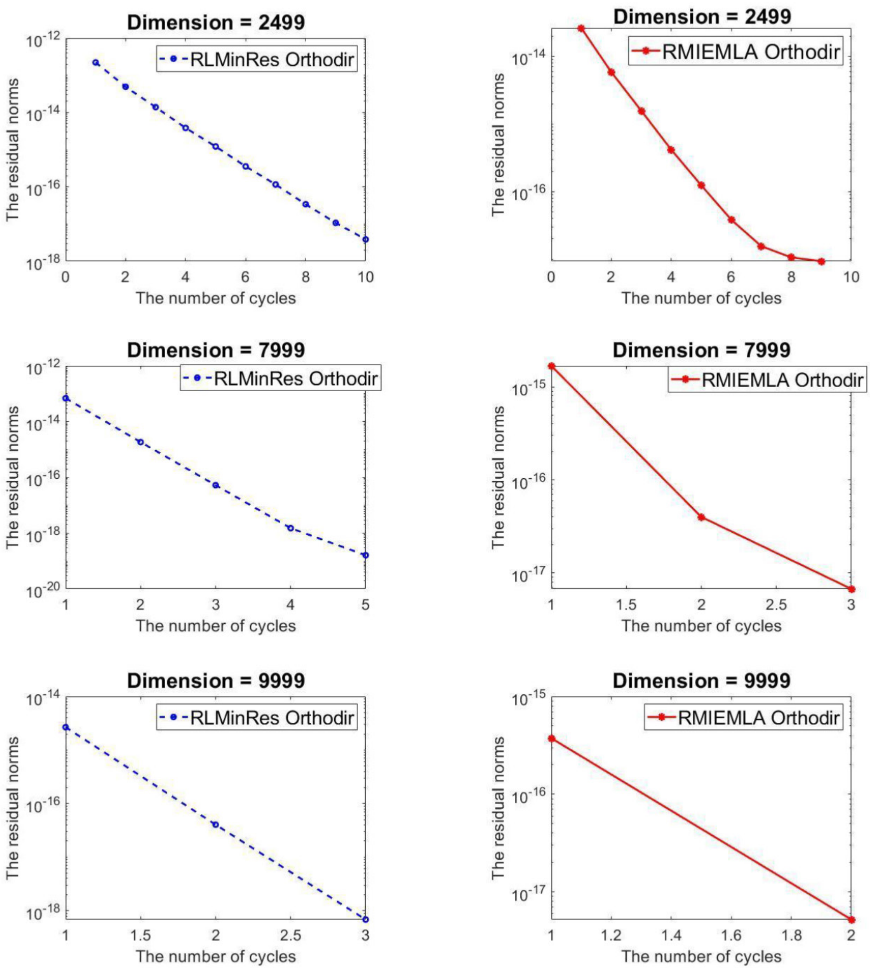

Figure 1: The performance of Hybrid FDM-RLMinRes (left side) and Hybrid FDM-RMEIEMLA (right side) on solving PDE Helmholtz

Both hybrid FDM-RLMinRes and hybrid FDM-RMEIEMLA involve cycles. One cycle, which consists of a maximum 100 iterations, represents one run of Algorithm 1 and Algorithm 2 respectively, and thus the residual norm is computed each time restarting. We use cycles to point out how many times the algorithm restarts until we reach our desired error. As can be seen in Figure 1 for instance, the hybrid FDM-RLMinRes for solving PDE Helmholtz in dimensions of SLE is 7999, needs five cycles, whereas hybrid FDM-RMEIEMLA needs three cycles. Also, as is clearly seen in Table 1, the hybrid FDM-RLMinRes gives a smaller residual norm, which means a better approximate solution for the Helmholtz problem.

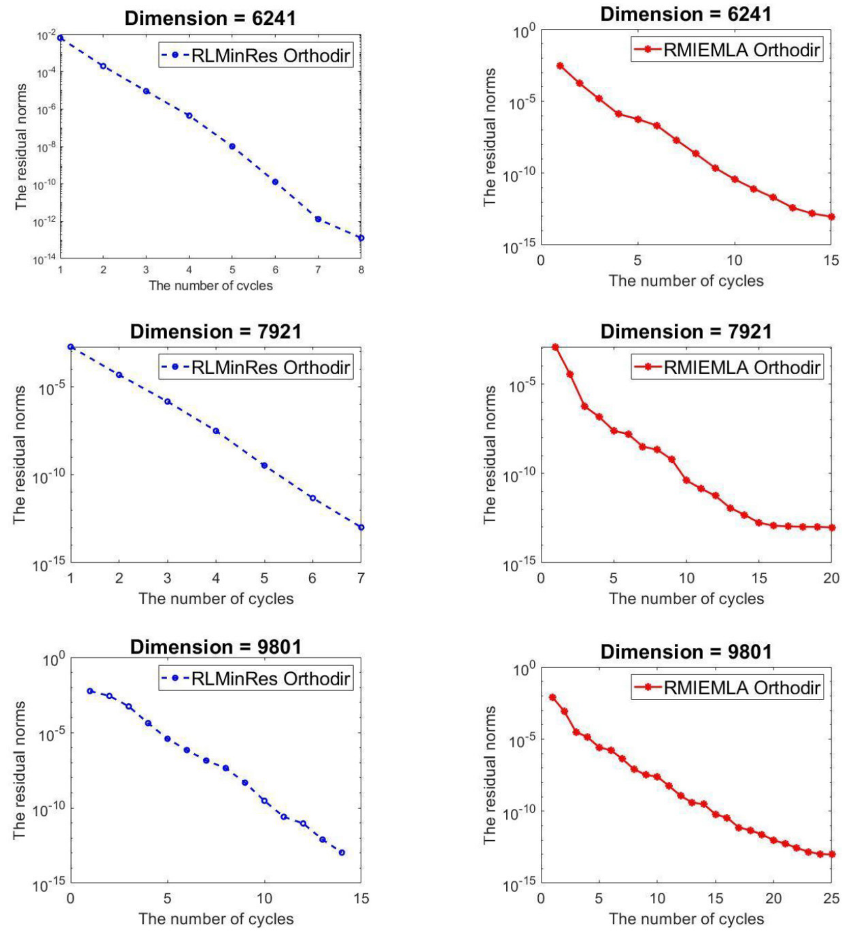

Figure 2: The performance of Hybrid FDM-RLMinRes (left side) and Hybrid FDM-RMEIEMLA (right side) on solving PDE wave

However, in a different case of number of cycles with the Helmholtz problem, in wave problem, based on Figure 2, the hybrid FDM-RMEIEMLA needs more cycles than the hybrid FDM-RLMinRes to solve the PDE wave. The FDM-RMEIEMLA also gives smaller residual norms than FDM-RLMinRes in wave problems, this can be seen in Table 2.

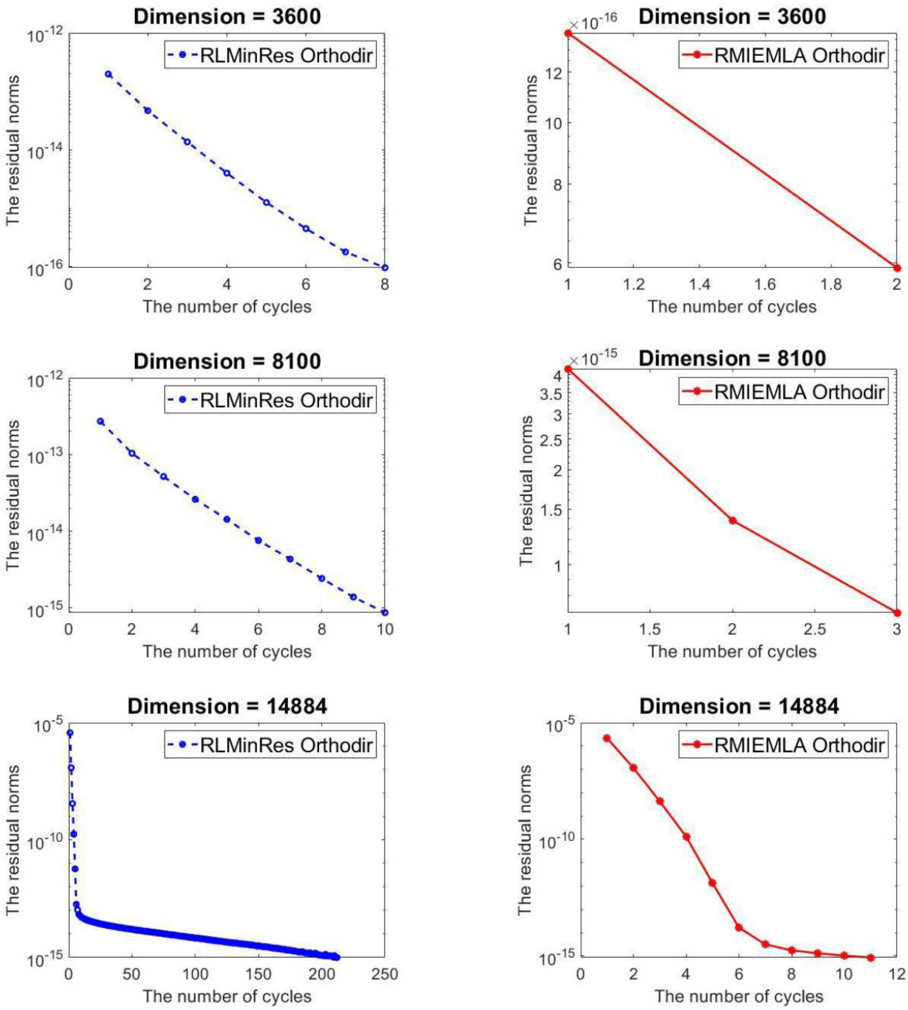

Figure 3: The performance of Hybrid FDM-RLMinRes (left side) and Hybrid FDM-RMEIEMLA (right side) on solving PDE heat problem

As for the heat problem, based on Figure 2 and Table 3, we find a similar explanation of cycles as the Helmholtz problem, that the hybrid FDM-RLMinRes needs more cycles and gives smaller residual norms than the hybrid FDM-RMEIEMLA.

DISCUSSION

Overall, the residual norms of approximate solutions generated by both hybrid FDM-RMEIEMLA and hybrid FDM-RLMinRes were small. Both methods work well in overcoming breakdown issues and in its application on solving various PDEs problems, namely, Helmholtz, wave, and heat equations. However, it can be seen in Table 1 and 3 that FDM-RLMinRes performed slightly better in Helmholtz and heat equations. While for the wave equation problem, FDM-RMEIEMLA performed better. This shows that the greater number of iterations used in the algorithm, the smaller the residual norms, which gives us better approximate solutions. For future study, we suggest applying these methods on implicit scheme FDM and for other PDEs boundary problems besides rectangular domain, i.e., circular domain. Moreover, we suggest solving PDEs problems in higher dimensions, that will give better accuracy if the value of time step and step size is very small.

CONCLUSION

We have implemented the hybrid FDM-RLMinRes and hybrid FDM-RMEIEMLA for solving various PDE problems which include Helmholtz, wave, and heat equations. Experimentally, the proposed hybrid methods resulted in good performance with small residual norms on the solutions of the PDE problems. This can be shown in numerical results, overall, the hybrid FDM-RMEIEMLA and hybrid FDM-RLMinRes computed the approximate solutions of the three PDE problems accurately.

CONFLICTS OF INTEREST

The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

ACKNOWLEDGEMENTS

This research is supported financially by Universiti Malaysia Terengganu.

REFERENCES

[1] T. Hillen. (2014). *Partial Differential Equations: Theory and completely solved problems*. Canada: John Wiley & Sons.

[2] R. L. Burden, & J. D. Faires. (2010). *Numerical Analysis* (9th ed.). Canada: Brooks/Cole.

[3] C. Lanczos. (1950). An iteration method for the solution of the eigenvalue problem of the linear differential and integral operators. *J. Res. Nat. Bur. Stand.*, *45*(4), 255-282. https://doi.org/10.6028/jres.045.026.

[4] C. Lanczos. (1952). Solution of system of linear equations by minimized iterations. *J. Res. Nat. Bur. Stand.*, *49*(1), 33-53. https://doi.org/10.6028/jres.049.006.

[5] C. Brezinski, & H. Sadok. (1993). Lanczos-type algorithms for solving systems of linear equations. *Journal of Applied Numerical Mathematics*, *11*(6), 443-473. https://doi.org/10.1016/0168-9274(93)90087-8.

[6] C. Baheux. (1995). New implementation of Lanczos method. *Journal of Computational and Applied Mathematics*, *57*(1-2), 3-15. https://doi.org/10.1016/0377-0427(93)E0230-J.

[7] M. Farooq, & A. Salhi. (2012). New recurrence relationships between orthogonal polynomials which lead to new Lanczos-type algorithms. *Journal of Prime Research in Mathematics*, *8*, 61-75.

[8] C. Brezinski, & M. R. Zaglia. (1994). Breakdowns in the computation of orthogonal polynomials. *Nonlinear Numerical Methods and Rational Approximation II*, *296*, 49–59. https://doi.org/10.1007/978-94-011-0970-3_5.

[9] C. Brezinski, & M. R. Zaglia. (1995). Look-ahead in Bi-CGSTAB and other methods for linear systems. *BIT*, *35*(2), 169–201. http://dx.doi.org/10.1007/BF01737161.

[10] A. B. Maharani, & A. Salhi. (2015). Restarting from specific points to cure breakdown in Lanczos-type algorithms. *Journal of Mathematical and & Fundamental Science*, *47*(2), 167-184. http://dx.doi.org/10.5614/j.math.fund.sci.2015.47.2.5.

[11] A. B. Maharani, A. Salhi, & R. A. Suharto. (2018). Introduction of interpolation and extrapolation model in Lanczos-type algorithms A13/B6 and A13/B3 to enhance their stability. *Journal of Mathematical and Fundamental Science*, *50*(2), 148-165. https://doi.org/10.5614/j.math.fund.sci.2018.50.2.4.

[12] A. B. Maharani, & A. Salhi. (2019). RMEIEMLA: The recent advance in improving the robustness of Lanczos-type algorithms. *AIP Conference Proceedings*, *2138*(1). http://dx.doi.org/10.1063/1.5121046.

[13] A. B. Maharani, A. Juliansyah, R. Thalib, & D. A. Pratama. (2021). High performances of stabilized Lanczos-Types for solving high dimension problems: A survey. *Int. J. of Math. and Comp. Sci.*, *16*(2), 837-854. https://doi.org/10.5281/zenodo.4431070.

[14] V. Grigoryan. (2010). Partial Differential Equations. Lecture note, University of California, Santa Barbara, version 0.1. Accessed April 8, 2021, from https://web.math.ucsb.edu/~grigoryan/124A.pdf

[15] S. Thirumugam. (2012). Numerical methods for hyperbolic Partial Differential Equation, Department of Physics School of Physical, Chemical and Applied Sciences Pondicherry University. Accessed April 5, 2021 from https://www.academia.edu/1827576/Numerical_Methods_for_Hyperbolic_PDE

[16] L. Zhang. (2002). The finite difference method for Helmholtz equation with application of cloaking. Accessed April 5, 2021, from https://math.missouristate.edu/assets/Math/li.pdf

[17] M. Witten. (2014). *Hyperbolic Partial Differential Equations: Populations, reactors, tides and waves: Theory and applications*. Chicago: Elsevier.

[18] B. E. Rapp. (2017). *Microfluidics: Modelling, mechanics and mathematics*. Elsevier.

[19] A. Z. Azmi. (2007). Numerical solution for heat equation, University Technology Malaysia. Accessed April 8, 2021, from https://www.nepjol.info/index.php/JNPhysSoc/article/view/34858/27355

[20] A. Schikorra. (2017). Parabolic Partial Differential Equations; lecture notes [PDF]. http://www.pitt.edu/~armin/PDEIIWS16/pde2.p

[21] L. Niragire. (2013). 2D numerical heat inverse model for diffusivity coefficients in cartesian coordinates. (master thesis, Indiana University of Pennsylvania).

Accessed April 5, 2021, from https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.881.3549&rep=rep1&type=pdf

[22] S. K. Njogu. (2009). Heat equations and their applications (one and two dimension heat equations). (master thesis, University of Nairobi). Accessed April 5, 2021, from http://erepository.uonbi.ac.ke/bitstream/handle/11295/95896/Njogu_Heat%20Equations%20And%20Their%20Applications%20%28One%20And%20Two%20Dimension%20Heat%20Equations%29.pdf?sequence=1&isAllowed=y

[23] M. Farooq, & A. Salhi. (2013). A preemptive restarting approach to beating inherence instability. *Iranian Journal of Science and Technology Transaction a Science*, *37*(3.1), 349-358. https://doi.org/10.22099/IJSTS.2013.1634.