

Research Article

The Higher Accuracy Fourth-Order IADE Algorithm

N. Abu Mansor,¹ A. K. Zulkifle,¹ N. Alias,² M. K. Hasan,³ and M. J. N. Boyce¹

¹ College of Engineering, Universiti Tenaga Nasional, Jalan Ikram-UNITEN, 43000 Kajang, Selangor, Malaysia

² Ibnu Sina Institute of Fundamental Science Studies, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

³ Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Correspondence should be addressed to N. Abu Mansor; noreliza@uniten.edu.my

Received 21 May 2013; Accepted 26 July 2013

Academic Editor: Juan Manuel Peña

Copyright © 2013 N. Abu Mansor et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This study develops the novel fourth-order iterative alternating decomposition explicit (IADE) method of Mitchell and Fairweather (IADEMF4) algorithm for the solution of the one-dimensional linear heat equation with Dirichlet boundary conditions. The higher-order finite difference scheme is developed by representing the spatial derivative in the heat equation with the fourth-order finite difference Crank-Nicolson approximation. This leads to the formation of pentadiagonal matrices in the systems of linear equations. The algorithm also employs the higher accuracy of the Mitchell and Fairweather variant. Despite the scheme's higher computational complexity, experimental results show that it is not only capable of enhancing the accuracy of the original corresponding method of second-order (IADEMF2), but its solutions are also in very much agreement with the exact solutions. Besides, it is unconditionally stable and has proven to be convergent. The IADEMF4 is also found to be more accurate, more efficient, and has better rate of convergence than the benchmarked fourth-order classical iterative methods, namely, the Jacobi (JAC4), the Gauss-Seidel (GS4), and the successive over-relaxation (SOR4) methods.

1. Introduction

Numerical methods with accuracy of the order, $O(\Delta h)^n$, $n > 2$, are referred to as higher-order methods (h = mesh size). Recent developments seem to desire methods of higher-order for achieving higher accuracy numerical solutions for problems involving partial differential equations. When higher-order methods are evaluated, factors such as rate of convergence, stability, and boundary conditions have to be considered too. There are mainly two categories of finite difference higher-order schemes.

The first category is the noncompact stencils that utilize any grid point surrounding the grid points of interest where the difference schemes are implemented. The method involves larger matrix bandwidth, but in many cases, the increase is not large [1]. There will be a probable increase in execution time as more grid points are used. However, increasing the number of grid points provides advantages in terms of enhancement in accuracy and improvement in resolution [2]. To enhance accuracy, the approach should consider proper treatment of boundary conditions of comparable accuracy.

The second category is the compact stencils that use Lax-Wendroff's idea [3] proposed by MacKinnon and Carey [4]. The method uses smaller number of stencils, making it computationally efficient and highly accurate. However, it requires inversion of matrix to obtain spatial derivative at each point. Also, the boundary stencil has a large effect on the stability and accuracy of the scheme [5]. Usually, compact schemes of order higher than four require formation of auxiliary equations due to boundary conditions. This results in large bandwidth matrices which are not symmetric [6]. Furthermore, the schemes can get fairly complicated for more complex equations.

Some of the current findings on higher-order methods include the work by Sulaiman et al. [7] who suggested a fourth-order quarter sweep modified successive over-relaxation (QMSOR) iterative method for solving a one-dimensional parabolic equation. It is found to be superior in terms of rate of convergence and execution time as compared to other SOR methods. Jha [8] formulated the six-order accurate quarter sweep alternating group explicit (QSAGE) iterative finite difference method for solving nonlinear singular two-point boundary value problems. The method can

be implemented in parallel and is found to be superior to the corresponding full sweep alternating group explicit (AGE) and SOR methods. Jin et al. [9] proposed an AGE iteration method of the fourth-order accuracy by integrating the grouping explicit method with numerical boundary conditions. The method was used to solve initial-boundary value problem of convection equations. Fu and Tan [10] showed that an unconditionally stable split-step FDTD method with higher-order spatial accuracy is more accurate than the lower-order methods. The dispersion error of the proposed method is comparable with the higher-order ADI-FDTD method.

Higher-order methods are also studied by Mohebbi and Deghan [11] who applied a compact finite difference approximation of fourth-order and the cubic C^1 spline collocation method to some one-dimensional heat and advection-diffusion equations. The scheme has fourth-order accuracy in both space and time and unconditionally stable. Liao [12] proposed an efficient and high-order accuracy of the fourth-order compact finite difference method to solve one-dimensional Burgers' equation. Tian and Ge [13] studied a stable fourth-order compact ADI method for solving two-dimensional unsteady convection diffusion problems. The method is temporally second-order and spatially fourth-order accurate, which requires only a regular five-point 2D stencil similar to that in the standard second-order methods. Chun [14] solved some nonlinear equations by applying the fourth-order iterative methods containing the King's fourth-order family. It was observed that the proposed method has at least equal performance compared to other methods of the same order. Zhu et al. [15] presented a high-order parallel finite difference algorithm based on the domain decomposition strategy. The study used the classical explicit scheme to calculate the interface values between subdomains and the fourth-order compact schemes for the interior values. The method has high accuracy and is convergent and stable. Gao and Xie [16] devised a fourth-order alternating direction implicit compact finite difference schemes for the solution of two-dimensional Schrödinger equations. The method is highly competitive as compared to other existing methods, and it achieves the expected convergence rate.

This study develops a higher-order finite difference algorithm, with noncompact stencils, that is capable of delivering highly accurate solutions for a one-dimensional heat equation with Dirichlet boundary conditions. The study focuses on modifying the unconditionally stable and convergent second-order iterative alternating decomposition explicit (IADE) method of Mitchell and Fairweather (IADEMF2). The IADEMF2, which was originally proposed by Sahimi et al. [17], employs the fractional splitting of the Mitchell and Fairweather (MF) variant that has an accuracy of the order $O((\Delta t)^2 + (\Delta x)^4)$. The scheme executes a two-stage process involving the solution of sets of tridiagonal equations along lines parallel to the first and second time steps, respectively. It is found to be unconditionally stable, convergent, and more accurate than the classical AGE class of methods, namely, the AGE method based on the Peaceman and Rachford variant (AGE-PR), whose spatial accuracy is of second order, and

the AGE method based on the Douglas and Rachford variant (AGE-DR), whose temporal accuracy is only of first order. The detailed derivation of the IADEMF2 can be obtained from [17].

In this paper, a fourth-order Crank-Nicolson (CN) difference approximation is applied to the spatial derivative in the heat equation, and the MF variant is employed, leading to the formation of the fourth-order IADEMF (IADEMF4) numerical algorithm. The convergence of the IADEMF4 is analyzed and proven. Numerical experiments verify the potential of the IADEMF4 in enhancing the accuracy of the IADEMF2. The results of the proposed higher-order scheme are also compared with the benchmarked fourth-order classical iterative methods, such as the fourth-order Gauss-Seidel (GS4), the fourth-order Jacobi (JAC4), and the fourth-order successive over-relaxation (SOR4) methods.

This paper is organized as follows. Section 2 discusses the implementation and stability of the fourth-order CN approximation on the heat equation. In Section 3, the IADEMF4 algorithm is formulated. Section 4 analyses the convergence of the IADEMF4. Section 5 provides the equations for the benchmarked fourth-order classical iterative methods. The computational complexity of the methods considered in this paper is given in Section 6, while the pseudocode for the IADEMF4 sequential algorithm is presented in Section 7. The experiments conducted are given in Section 8. Sections 9 and 10 provide some discussion and conclusion based on the obtained numerical results.

2. Fourth-Order Crank-Nicolson Approximation to the Spatial Derivative in the Heat Equation

Consider the following one-dimensional parabolic heat equation (1) that has been suitably assumed to be nondimensionalised. It models the flow of heat in a homogeneous unchanging medium of finite extent in the absence of heat source

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2} \quad (1)$$

subject to given initial and Dirichlet boundary conditions

$$\begin{aligned} U(x, 0) &= f(x), \quad 0 \leq x \leq 1, \\ U(0, t) &= g(t), \quad 0 < t \leq T, \\ U(1, t) &= h(t), \quad 0 < t \leq T. \end{aligned} \quad (2)$$

For the problem in (1), the finite difference approach discretizes the time-space domain by placing a rectangular grid over the domain, with grid spacing of Δt and Δx in the t - and x -directions, respectively. The grid consists of the set of lines parallel to the t -axis given by $x_i = i\Delta x$, $i = 0, 1, \dots, m, m+1$ and a set of lines parallel to the x -axis given by $t_k = k\Delta t$, $k = 0, 1, \dots, n, n+1$. For simplicity, the grid spacing is taken to be uniform, so that $\Delta x = 1/(m+1)$, and $\Delta t = T/(n+1)$. At a grid-point $P(x_i, t_k)$ in the solution domain, the dependent variable $U(x, t)$ which represents the nondimensional temperature at time t and at position x is approximated by u_i^k .

At the grid-point $P(x_i, t_{k+1/2})$, the IADEFM4 scheme replaces the spatial derivative in the heat equation with a higher-order, particularly the fourth-order Crank-Nicolson (CN) difference approximation [18]. This is shown in the expression given in (3), with the central difference operators defined as $\delta_x^2 u_i^k = u_{i-1}^k - 2u_i^k + u_{i+1}^k$ and $\delta_x^2 u_i^{k+1} = u_{i-1}^{k+1} - 2u_i^{k+1} + u_{i+1}^{k+1}$. The approach gives the fourth-order scheme a spatial truncation error of the order $O(\Delta x)^4$. It enhances the accuracy of its second-order counterpart, which has a larger error of the order $O(\Delta x)^2$.

$$\frac{1}{\Delta t} (u_i^{k+1} - u_i^k) = \frac{1}{2(\Delta x)^2} \left(\delta_x^2 - \frac{1}{12} \delta_x^4 \right) (u_i^{k+1} + u_i^k). \quad (3)$$

To determine the stability of (3), the von Neumann stability analysis can be applied. Let $\lambda = \Delta t/(\Delta x)^2$, and since $\delta_x^4 u_i^k = \delta_x^2 (\delta_x^2 u_i^k)$, the discretization of (3) becomes

$$\begin{aligned} u_i^{k+1} - u_i^k = & \frac{\lambda}{2} \left(-\frac{1}{12} u_{i-2}^{k+1} + \frac{4}{3} u_{i-1}^{k+1} \right. \\ & - \frac{5}{2} u_i^{k+1} + \frac{4}{3} u_{i+1}^{k+1} - \frac{1}{12} u_{i+2}^{k+1} \\ & - \frac{1}{12} u_{i-2}^k + \frac{4}{3} u_{i-1}^k - \frac{5}{2} u_i^k \\ & \left. + \frac{4}{3} u_{i+1}^k - \frac{1}{12} u_{i+2}^k \right). \end{aligned} \quad (4)$$

The discretized equation in (4) is assumed to have a solution in the form of a Fourier harmonic function; that is, $u_i^k = \rho^k e^{s\beta i \Delta x}$, where ρ is referred to as the amplification factor, β is an arbitrary constant, and $s = \sqrt{-1}$. The amplification factor represents the time dependence of the solution. If the Fourier function is substituted into (4) and then solve for ρ , the result will be

$$\rho = \frac{1 - (\lambda/6) \left((\cos \tilde{\beta} - 4)^2 - 9 \right)}{1 + (\lambda/6) \left((\cos \tilde{\beta} - 4)^2 - 9 \right)}, \quad \text{with } \tilde{\beta} = \beta \Delta x. \quad (5)$$

Since $|\rho| \leq 1$, then the fourth-order CN approximation is unconditionally stable for any choice of β , Δt , and Δx .

3. The Formulation of the IADEFM4

The IADEFM4 is firstly developed based on the execution of the unconditionally stable fourth-order CN approximation (3) on the heat equation.

Equation (4) can also be expressed as in (6), using the definitions of the constants given in (7)

$$\begin{aligned} au_{i-2}^{k+1} + bu_{i-1}^{k+1} + cu_i^{k+1} + du_{i+1}^{k+1} + eu_{i+2}^{k+1} \\ = -au_{i-2}^k - bu_{i-1}^k + \hat{c}u_i^k - du_{i+1}^k - eu_{i+2}^k, \end{aligned} \quad (6)$$

$$i = 2, 3, \dots, m-1,$$

$$\begin{aligned} a = \frac{\lambda}{24}, \quad b = -\frac{2\lambda}{3}, \quad c = \frac{4+5\lambda}{4}, \\ d = -\frac{2\lambda}{3}, \quad e = \frac{\lambda}{24}, \quad \hat{c} = \frac{4-5\lambda}{4}. \end{aligned} \quad (7)$$

The approximation in (6) can be displayed in a matrix form such as $A\mathbf{u} = \mathbf{f}$ (8), where A is a sparse pentadiagonal coefficient matrix and $\mathbf{u} = (u_2, u_3, \dots, u_{m-2}, u_{m-1})^T$ is the column vector containing the unknown values of u at the time level $k+1$. The column vector $\mathbf{f} = (f_2, f_3, \dots, f_{m-2}, f_{m-1})^T$ consists of boundary values and known u values at the previous time level k . The definitions for every entry in \mathbf{f} are given in (9)

$$A\mathbf{u} = \mathbf{f} \quad \begin{bmatrix} c & d & e & & & \\ b & c & d & e & & \mathbf{0} \\ a & b & c & d & e & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & a & b & c & d & e \\ \mathbf{0} & & & a & b & c & d \\ & & & & a & b & c \end{bmatrix}_{(m-2) \times (m-2)} \begin{bmatrix} u_2 \\ u_3 \\ \vdots \\ u_{m-2} \\ u_{m-1} \end{bmatrix}_{k+1} = \begin{bmatrix} f_2 \\ f_3 \\ \vdots \\ f_{m-2} \\ f_{m-1} \end{bmatrix}, \quad (8)$$

$$\begin{aligned} f_2 &= -b(u_1^k + u_1^{k+1}) + \hat{c}u_2^k - du_3^k - eu_4^k, \\ f_3 &= -a(u_1^k + u_1^{k+1}) - bu_2^k + \hat{c}u_3^k - du_4^k - eu_5^k, \\ f_i &= -au_{i-2}^k - bu_{i-1}^k + \hat{c}u_i^k - du_{i+1}^k - eu_{i+2}^k \\ &\quad \text{for } i = 4, 5, \dots, m-3, \end{aligned} \quad (9)$$

$$\begin{aligned} f_{m-2} &= -au_{m-4}^k - bu_{m-3}^k + \hat{c}u_{m-2}^k \\ &\quad - du_{m-1}^k - e(u_m^k + u_m^{k+1}), \\ f_{m-1} &= -au_{m-3}^k - bu_{m-2}^k + \hat{c}u_{m-1}^k - d(u_m^k + u_m^{k+1}). \end{aligned}$$

The evaluations of f_2 , f_3 , f_{m-2} , and f_{m-1} require the values of u at the boundaries $i = 1$ and $i = m$. However, these values cannot be obtained numerically because their computations involve nodes at $i = -1$ and $i = m+2$, which are exterior to the considered solution domain. If the exact solutions of u are available at $i = 1$ and $i = m$, then it is appropriate to consider them as the required boundary values. Otherwise, the boundary conditions have to be formulated, bearing in mind that they should be of comparable accuracy [1].

The IADEFM4 scheme secondly employs the higher-order accuracy formula of MF [19]. The variant, whose accuracy is of the order $O((\Delta t)^2 + (\Delta x)^4)$, is as given in (10) and (11)

$$(rI + G_1)\mathbf{u}^{(p+1/2)} = (rI - gG_2)\mathbf{u}^{(p)} + \mathbf{f}, \quad (10)$$

$$(rI + G_2)\mathbf{u}^{(p+1)} = (rI - gG_1)\mathbf{u}^{(p+1/2)} + g\mathbf{f}, \quad (11)$$

where r , p , and I represent an acceleration parameter, the iteration index, and an identity matrix, respectively. G_1 and G_2 are two constituent matrices. The vectors $\mathbf{u}^{(p+1)}$ and $\mathbf{u}^{(p+1/2)}$ represent the required solution at the iteration level $(p+1)$ and at some intermediate level $(p+1/2)$, respectively. The relation of g and r is given by $g = (6+r)/6$, $r > 0$.

Substitute the following expression obtained from (10) into (11)

$$\mathbf{u}^{(p+1/2)} = (rI + G_1)^{-1} \left[(rI - gG_2) \mathbf{u}^{(p)} + \mathbf{f} \right], \quad (12)$$

and then simplify to obtain

$$(rI + G_2) \mathbf{u}^{(p+1)} = (rI - gG_1) (rI + G_1)^{-1} (rI - gG_2) \mathbf{u}^{(p)} + \left[(rI - gG_1) (rI - gG_1)^{-1} + g \right] \mathbf{f}. \quad (13)$$

As p in (13) becomes sufficiently large, the temperature solution reaches a steady state that is, as $p \rightarrow \infty$, then $\mathbf{u}^{(p+1)} \rightarrow \mathbf{u}$ and $\mathbf{u}^{(p)} \rightarrow \mathbf{u}$. Simplify the equation, and then multiply by $(rI + G_1)$. After some algebraic manipulations, the following expression is obtained:

$$\left[r(G_1 + G_2)(1 + g) + (1 - g^2)G_1G_2 \right] \mathbf{u} = r(1 + g)\mathbf{f}. \quad (14)$$

Multiply (14) by $1/r(1 + g)$, and use the definition of g to finally obtain the following form:

$$\left[(G_1 + G_2) - \frac{1}{6}G_1G_2 \right] \mathbf{u} = \mathbf{f}. \quad (15)$$

The comparison between the expression $A\mathbf{u} = \mathbf{f}$ in (8) and the form given in (15) suggests that the coefficient matrix A for the IADEFM4 can be decomposed into

$$A = G_1 + G_2 - \frac{1}{6}G_1G_2. \quad (16)$$

The IADEFM4 requires the constituent matrices G_1 and G_2 in (16) to be in the form of lower and upper tridiagonal matrices, respectively, in order to retain the pentadiagonal structure of A . Thus,

$$G_1 = \begin{bmatrix} 1 & & & & \\ l_1 & 1 & & & O \\ \hat{m}_1 & l_2 & \ddots & & \\ & \hat{m}_2 & \ddots & \ddots & \\ & & \ddots & l_{m-4} & 1 \\ O & & & \hat{m}_{m-4} & l_{m-3} & 1 \end{bmatrix}_{(m-2) \times (m-2)}, \quad (17)$$

$$G_2 = \begin{bmatrix} \hat{e}_1 & \hat{u}_1 & \hat{v}_1 & & & \\ \hat{e}_2 & \hat{u}_2 & \hat{v}_2 & & O & \\ & \ddots & \ddots & \ddots & & \\ O & & \hat{e}_{m-4} & \hat{u}_{m-4} & \hat{v}_{m-4} & \\ & & & \hat{e}_{m-3} & \hat{u}_{m-3} & \\ & & & & \hat{e}_{m-2} & \end{bmatrix}_{(m-2) \times (m-2)}.$$

By substituting G_1 and G_2 in (17) into the formula for the decomposition of A , the entries of the resultant matrix are compared with the entries of A in (8), yielding the following definitions:

$$\begin{aligned} \hat{e}_1 &= \frac{6(c-1)}{5}, & \hat{u}_1 &= \frac{6d}{5}, & l_1 &= \frac{6b}{6-\hat{e}_1}, \\ \hat{e}_2 &= \frac{6(c-1) + l_1\hat{u}_1}{5}, & \hat{v}_i &= \frac{6e}{5} & \text{for } i &= 1, 2, \dots, m-4. \end{aligned} \quad (18)$$

And for $i = 2, 3, \dots, m-3$,

$$\begin{aligned} \hat{u}_i &= \frac{6d + l_{i-1}\hat{v}_{i-1}}{5}, & \hat{m}_{i-1} &= \frac{6a}{6-\hat{e}_{i-1}}, \\ l_i &= \frac{6b + \hat{m}_{i-1}\hat{u}_{i-1}}{6-\hat{e}_i}, & \hat{e}_{i+1} &= \frac{6(c-1) + l_i\hat{u}_i + \hat{m}_{i-1}\hat{v}_{i-1}}{5}. \end{aligned} \quad (19)$$

Since G_1 and G_2 are three banded matrices, then $(rI + G_1)$ and $(rI + G_2)$ can be inverted easily. The equations in (10) and (11) are rearranged as in (20) and (21), respectively,

$$\mathbf{u}^{(p+1/2)} = (rI + G_1)^{-1} (rI - gG_2) \mathbf{u}^{(p)} + (rI + G_1)^{-1} \mathbf{f}, \quad (20)$$

$$\mathbf{u}^{(p+1)} = (rI + G_2)^{-1} (rI - gG_1) \mathbf{u}^{(p+1/2)} + g(rI + G_2)^{-1} \mathbf{f}. \quad (21)$$

The above two equations are computed and simplified, leading to the computational formulae at each of the half iteration levels as given in (22) and (23).

(i) At the $(p + 1/2)$ iteration level,

$$\begin{aligned} u_i^{(p+1/2)} &= \frac{1}{R} \left(E_{i-1} u_i^{(p)} + W_{i-1} u_{i+1}^{(p)} + V_{i-1} u_{i+2}^{(p)} \right. \\ &\quad \left. - \hat{m}_{i-3} u_{i-2}^{(p+1/2)} - l_{i-2} u_{i-1}^{(p+1/2)} + f_i \right), \\ i &= 2, 3, \dots, m-2, m-1 \end{aligned} \quad (22)$$

(ii) At the $(p + 1)$ iteration level,

$$\begin{aligned} u_i^{(p+1)} &= \frac{1}{Z_{i-1}} \left(S_{i-3} u_{i-2}^{(p+1/2)} + Q_{i-2} u_{i-1}^{(p+1/2)} + P u_i^{(p+1/2)} \right. \\ &\quad \left. - \hat{u}_{i-1} u_{i+1}^{(p+1)} - \hat{v}_{i-1} u_{i+2}^{(p+1)} + g f_i \right), \\ i &= m-1, m-2, \dots, 3, 2 \end{aligned} \quad (23)$$

with

$$\begin{aligned} \hat{m}_{-1} &= \hat{m}_0 = l_0 = V_{m-2} = V_{m-3} = W_{m-2} = \hat{u}_{m-2} \\ &= \hat{v}_{m-2} = \hat{v}_{m-3} = Q_0 = S_{-1} = S_0 = 0, \\ R &= 1 + r, & P &= r - g, \\ E_i &= r - g\hat{e}_i, & Z_i &= r + \hat{e}_i, & i &= 1, 2, \dots, m-2, \\ W_i &= -g\hat{u}_i, & Q_i &= -gl_i, & i &= 1, 2, \dots, m-3, \\ V_i &= -g\hat{v}_i, & S_i &= -g\hat{m}_i, & i &= 1, 2, \dots, m-4. \end{aligned} \quad (24)$$

The IADEFM4 algorithm is regarded as a two-stage process involving two iteration levels, $(p + 1/2)$ and $(p + 1)$. It is completed explicitly by using the required equations at the two levels in alternate sweeps along all the grid points in the interval $(0, 1)$ until convergence is reached. In (22), the calculation to determine the unknown $u_i^{(p+1/2)}$ begins at the left boundary and then moves to the right. In a similar

manner, $u_i^{(p+1)}$ in (23) is calculated by proceeding from the right boundary towards the left (Figure 1).

The computational molecules are depicted in Figures 2 and 3.

At each level of iteration, the computational molecules involve two known grid points at the new level and another three known ones at the old level. Clearly, the method is explicit.

4. Convergence Analysis of the IADEMF4

This section proves the convergence of the IADEMF4. Since $\lambda > 1$ will not guarantee an accurate approximation for $\partial U / \partial t$ [18], the values of λ that are considered appropriate for the proof are $0 < \lambda \leq 1$.

From the definitions in ((18) and (19)), the following results are obtained:

$$\begin{aligned}\hat{e}_1 &= \frac{6(c-1)}{5} = \frac{3\lambda}{2}, \quad \text{implying that } 0 < \hat{e}_1 \leq \frac{3}{2}, \\ \hat{u}_1 &= \frac{6d}{5} < 0, \quad l_1 = \frac{6b}{6-\hat{e}_1} < 0, \quad \hat{v}_1 = \frac{6e}{5} > 0 \\ &\quad \text{for } i = 1, 2, \dots, m-4, \\ \hat{u}_2 &= \frac{6d + l_1 \hat{v}_1}{5} = \hat{u}_1 + \frac{l_1 \hat{v}_1}{5} < \hat{u}_1, \quad \text{since } l_1 \hat{v}_1 < 0, \\ \hat{e}_2 &= \frac{6(c-1) + l_1 \hat{u}_1}{5} = \hat{e}_1 + \frac{l_1 \hat{u}_1}{5} > \hat{e}_1, \quad \text{since } l_1 \hat{u}_1 > 0.\end{aligned}\quad (25)$$

Simple computation shows that $l_1 \hat{u}_1 / 5 < 1$. Clearly, $\hat{e}_2 < 6$.

Lemma 1. \hat{e}_i in ((18) and (19)) is such that, $0 < \hat{e}_i < 6$, for all $i = 1, 2, 3, \dots, m-2$.

Proof. The results in (25) show that $6 > \hat{e}_2 > \hat{e}_1 > 0$. Assume it is true that $6 > \hat{e}_k > \hat{e}_{k-1} > 0$ for all $k = 3, 4, 5, \dots, m-3$. Since the assumption implies that $6 > \hat{e}_{k-1} > \hat{e}_{k-2} > 0$, then $\hat{m}_{k-1} = 6a / (6 - \hat{e}_{k-1}) > \hat{m}_{k-2} = 6a / (6 - \hat{e}_{k-2}) > 0$. Therefore, $\hat{m}_{k-1} \hat{v}_{k-1} > \hat{m}_{k-2} \hat{v}_{k-2}$.

It has also been shown that $\hat{u}_2 < \hat{u}_1 < 0$. Assume it is true that $\hat{u}_j < \hat{u}_{j-1} < 0$ for all $j = 3, 4, \dots, m-4$. The assumption implies that $\hat{u}_{j-1} < \hat{u}_{j-2} < 0$. Thus, $\hat{m}_{j-1} \hat{u}_{j-1} < \hat{m}_{j-2} \hat{u}_{j-2} < 0$

$$\begin{aligned}\hat{u}_{j+1} &= \frac{6d + l_j \hat{v}_j}{5} = \hat{u}_1 + \frac{l_j \hat{v}_j}{5} = \hat{u}_1 + \frac{(6b + \hat{m}_{j-1} \hat{u}_{j-1}) \hat{v}_j}{5(6 - \hat{e}_j)} \\ &< \hat{u}_1 + \frac{(6b + \hat{m}_{j-2} \hat{u}_{j-2}) \hat{v}_{j-1}}{5(6 - \hat{e}_{j-1})} \\ &= \hat{u}_1 + \frac{l_{j-1} \hat{v}_{j-1}}{5} = \hat{u}_j < 0.\end{aligned}\quad (26)$$

Since it is true that, for $j+1$, $\hat{u}_{j+1} < \hat{u}_j < 0$, then, by induction, the assumption $\hat{u}_j < \hat{u}_{j-1} < 0$ is true for all j . An

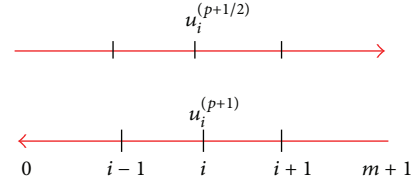


FIGURE 1: The two-stage IADEMF4 algorithm. The directions of the sweeps at the $(p+1/2)$ and $(p+1)$ iteration levels.

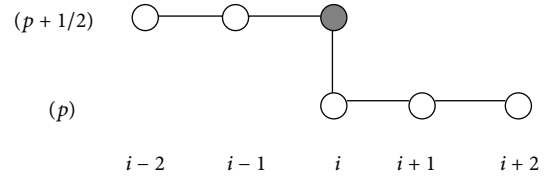


FIGURE 2: Computational molecule of the IADEMF4 at the $(p+1/2)$ iteration level.

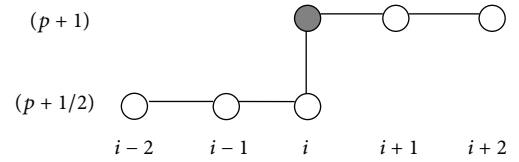


FIGURE 3: Computational molecule of the IADEMF4 at the $(p+1)$ iteration level.

equivalent to the preceding statement would be $\hat{u}_k < \hat{u}_{k-1} < 0$ for all $k = 3, 4, 5, \dots, m-3$. It follows that

$$\begin{aligned}l_k \hat{u}_k &= \frac{(6b + \hat{m}_{k-1} \hat{u}_{k-1}) \hat{u}_k}{6 - \hat{e}_k} \\ &> \frac{(6b + \hat{m}_{k-2} \hat{u}_{k-2}) \hat{u}_{k-1}}{6 - \hat{e}_{k-1}} = l_{k-1} \hat{u}_{k-1}, \\ \hat{e}_{k+1} &= \frac{6(c-1) + l_k \hat{u}_k + \hat{m}_{k-1} \hat{v}_{k-1}}{5} = \hat{e}_1 + \frac{l_k \hat{u}_k + \hat{m}_{k-1} \hat{v}_{k-1}}{5} \\ &> \hat{e}_1 + \frac{l_{k-1} \hat{u}_{k-1} + \hat{m}_{k-2} \hat{v}_{k-2}}{5} = \hat{e}_k > 0.\end{aligned}\quad (27)$$

Since it is true that, for $k+1$, $\hat{e}_{k+1} > \hat{e}_k > 0$, then, by induction, $\hat{e}_k > \hat{e}_{k-1} > 0$ is also true for all k .

Suppose there is a k such that $\hat{e}_k > 6$. Then,

$$\begin{aligned}\hat{e}_{k+1} &= \hat{e}_1 + \frac{l_k \hat{u}_k + \hat{m}_{k-1} \hat{v}_{k-1}}{5} \\ &= \hat{e}_1 + \frac{6b + \hat{u}_k}{5(6 - \hat{e}_k)} + \frac{\hat{m}_{k-1}}{5} \left(\frac{\hat{u}_{k-1} \hat{u}_k}{6 - \hat{e}_k} + \hat{v}_{k-1} \right) < \hat{e}_1.\end{aligned}\quad (28)$$

This is a contradiction since $\hat{e}_{k+1} > \hat{e}_1$. This verifies that $6 > \hat{e}_k > \hat{e}_{k-1} > 0$, $k = 3, 4, 5, \dots, m-3$. Let $i = 2, 3, 4, \dots, m-2$, and then $6 > \hat{e}_i > \hat{e}_{i-1} > 0$. Therefore, $0 < \hat{e}_i < 6$, $i = 1, 2, 3, \dots, m-2$. Lemma 1 is proved. \square

Lemma 2. If $r > 0$ and $g = (6+r)/6$, then $\max |(r - g\hat{e}_i)/(r + \hat{e}_i)| < 1$, $i = 1, 2, 3, \dots, m-2$.

Proof. Let $\max |(r - g\hat{e}_j)/(r + \hat{e}_j)| = |(r - g\hat{e}_j)/(r + \hat{e}_j)|$ for some j .

Assume $|(r - g\hat{e}_j)/(r + \hat{e}_j)| \geq 1$. Since $r > 0$ and $\hat{e}_j > 0$, then $r + \hat{e}_j > 0$.

If $(r - g\hat{e}_j)/(r + \hat{e}_j) \geq 1$, then $-((6+r)/6)\hat{e}_j \geq \hat{e}_j$, which implies that $r \leq -12$. This contradicts the fact that $r > 0$.

If $(r - g\hat{e}_j)/(r + \hat{e}_j) \leq -1$, then $2r - ((6+r)/6)\hat{e}_j \leq -\hat{e}_j$, which implies that $\hat{e}_j \geq 12$. This contradicts Lemma 1.

So, the assumption that $|(r - g\hat{e}_j)/(r + \hat{e}_j)| \geq 1$ is false. Hence, $\max |(r - g\hat{e}_i)/(r + \hat{e}_i)| = |(r - g\hat{e}_j)/(r + \hat{e}_j)| < 1$. \square

Proposition 3. $\|(rI - gG_1)(rI + G_1)^{-1}\|_2 < 1$.

Proof. Let $F = (rI - gG_1)(rI + G_1)^{-1}$; that is,

$$F = \begin{bmatrix} \frac{r-g}{d} & & & & \\ \cdot & \frac{r-g}{d} & & & \\ \cdot & \cdot & \ddots & & \\ \vdots & \vdots & \vdots & \ddots & \\ \dots & \dots & \dots & \dots & \frac{r-g}{d} \end{bmatrix}. \quad (29)$$

F is a lower triangular matrix, with all the diagonal entries (whence the eigenvalues of F) equal to $(r - g)/d$, where $d = r + 1$. Denote all the eigenvalues of F by λ_F . If $\rho[F]$ is defined as the spectral radius of F , then

$$\rho[F] = \max |\lambda_F| = |\lambda_F| = \left| \frac{r-g}{r+1} \right| \leq \|F\|_2. \quad (30)$$

But, by definition of 2-norm,

$$\begin{aligned} \|F\|_2 &= \max_{\|x\|_2 \neq 0} \frac{\|Fx\|_2}{\|x\|_2} = \max_{\|y\|_2=1} \frac{\|Fy\|_2}{\|y\|_2} = \max_{\|y\|_2=1} \|Fy\|_2 \\ &\leq \max_{\|w\|_2=1} \|Fw\|_2 = \max_{\|w\|_2=1} \|\lambda_F w\|_2. \end{aligned} \quad (31)$$

Since all the eigenvalues of F are equal, then

$$\|F\|_2 \leq |\lambda_F| \max_{\|w\|_2=1} \|w\|_2 = |\lambda_F| = \rho[F]. \quad (32)$$

Thus, from (30) and (32), $\|F\|_2 = |\lambda_F| = |(r - g)/(r + 1)|$. By Lemma 2 with e_j replaced by 1, $|(r - g)/(r + 1)| < 1$ is obtained, leading to the result of Proposition 3, which is $\|F\|_2 = \|(rI - gG_1)(rI + G_1)^{-1}\|_2 < 1$. \square

Proposition 4. $\|(rI - gG_2)(rI + G_2)^{-1}\|_2 < 1$.

Proof. Let $K = (rI - gG_2)(rI + G_2)^{-1}$; that is,

$$K = \begin{bmatrix} \frac{r-g\hat{e}_1}{z_1} & \cdot & \cdot & \dots & \dots & \dots \\ & \frac{r-g\hat{e}_2}{z_2} & \cdot & \cdot & \dots & \vdots \\ & & \ddots & \vdots & \vdots & \vdots \\ & & & O & \ddots & \vdots \\ & & & & \ddots & \frac{r-g\hat{e}_m}{z_m} \end{bmatrix}. \quad (33)$$

K is an upper triangular matrix, with all the diagonal entries equal to $(r - g\hat{e}_i)/z_i$, $i = 1, 2, \dots, m-2$, where $z_i = r + \hat{e}_i$. Since all the eigenvalues of K are distinct, then K is similar to a diagonal matrix D .

By the Schur triangularization theorem [20], there is an orthogonal matrix O such that $O^T K O = D$. The diagonal entries of D are the eigenvalues of K

$$\begin{aligned} \|K\|_2 &= \|O^T K O\|_2 = \|D\|_2 \\ &= \sqrt{\text{maximum eigenvalue of } D^T D} \\ &= \sqrt{\text{maximum eigenvalue of } D^2} \\ &= \sqrt{\rho(D^2)} = \sqrt{\rho(K^2)} = \rho(K). \end{aligned} \quad (34)$$

By Lemma 2,

$$\|K\|_2 = \rho(K) = \max \left| \frac{r - g\hat{e}_i}{r + \hat{e}_i} \right| = \left| \frac{r - g\hat{e}_j}{r + \hat{e}_j} \right| < 1, \quad (35)$$

for some j . Hence, this proves Proposition 4. \square

From (20) and (21),

$$\begin{aligned} \mathbf{u}^{(p+1)} &= (rI + G_2)^{-1} (rI - gG_1) \\ &\quad \times [(rI + G_1)^{-1} (rI - gG_2) \mathbf{u}^{(p)} + (rI + G_1)^{-1} \mathbf{f}] \\ &\quad + g (rI + G_2)^{-1} \mathbf{f}. \end{aligned} \quad (36)$$

Let

$$M(r) = (rI + G_2)^{-1} (rI - gG_1) (rI + G_1)^{-1} (rI - gG_2), \quad (37)$$

And let

$$\begin{aligned} q(r) &= [(rI + G_2)^{-1} (rI - gG_1) (rI + G_1)^{-1} \\ &\quad + g(rI + G_2)^{-1}] \mathbf{f}. \end{aligned} \quad (38)$$

Then, $\mathbf{u}^{(p+1)} = M(r)\mathbf{u}^{(p)} + q(r)$.

Theorem 5. The IADEMF4 is convergent if $\rho[M(r)] < 1$, for $r > 0$.

Proof. Define $\widetilde{M}(r) = (rI + G_2)M(r)(rI + G_2)^{-1}$; then

$$\widetilde{M}(r) = (rI - gG_1)(rI + G_1)^{-1}(rI - gG_2)(rI + G_2)^{-1}. \quad (39)$$

Thus, by similarity, $M(r)$ and $\widetilde{M}(r)$ have the same set of eigenvalues. Therefore,

$$\begin{aligned} \rho[M(r)] &= \rho[\widetilde{M}(r)] \leq \|\widetilde{M}(r)\|_2 \\ &= \|(rI - gG_1)(rI + G_1)^{-1}(rI - gG_2)(rI + G_2)^{-1}\|_2 \\ &\leq \|(rI - gG_1)(rI + G_1)^{-1}\|_2 \|(rI - gG_2)(rI + G_2)^{-1}\|_2 \\ &< 1. \end{aligned} \quad (40)$$

The last inequality is due to Propositions 3 and 4. The proven Theorem 5 assures the convergence of the IADEFM4. \square

5. The Fourth-Order Classical Iterative Methods

The system of linear equations in (8) may also be solved by using the classical iterative methods of the fourth order. They include the JAC4, the GS4, and the SOR4 methods. These iterative methods are capable of exploiting the sparse structure of the pentadiagonal matrix.

The JAC4 algorithm can be represented by

$$u_i^{(p+1)} = \frac{(f_i - au_{i-2}^{(p)} - bu_{i-1}^{(p)} - du_{i+1}^{(p)} - eu_{i+2}^{(p)})}{c}, \quad (41)$$

$$i = 2, 3, \dots, m-1.$$

The approximation of $u_i^{(p+1)}$ at the $(p+1)$ th iteration level is computed using the relevant values in the p th iteration.

The GS4 method uses the most recent values of $u_{i-2}^{(p+1)}$ and $u_{i-1}^{(p+1)}$ to update the approximation value of $u_i^{(p+1)}$. The algorithm for the GS4 is as expressed in (42)

$$u_i^{(p+1)} = \frac{(f_i - au_{i-2}^{(p+1)} - bu_{i-1}^{(p+1)} - du_{i+1}^{(p)} - eu_{i+2}^{(p)})}{c}, \quad (42)$$

$$i = 2, 3, \dots, m-1.$$

The SOR4 iterative method accelerates the convergence rate of the GS4. If ω is a relaxation parameter, then for any $\omega \neq 0$, (42) can be rewritten as

$$\begin{aligned} u_i^{(p+1)} &= (1 - \omega) u_i^{(p)} \\ &+ \frac{\omega (f_i - au_{i-2}^{(p+1)} - bu_{i-1}^{(p+1)} - du_{i+1}^{(p)} - eu_{i+2}^{(p)})}{c}, \end{aligned} \quad (43)$$

$$i = 2, 3, \dots, m-1.$$

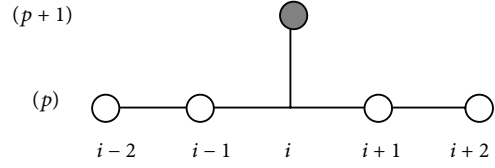


FIGURE 4: Computational molecule of the JAC4.

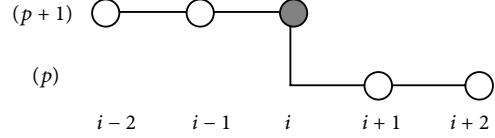


FIGURE 5: Computational molecule of the GS4/SOR4.

TABLE 1: Sequential arithmetic operations per iteration (m : problem size).

Method	Number of additions	Number of multiplications	Total operation count
IADEFM4	$10(m-2)$	$13(m-2)$	$23(m-2)$
IADEFM2	$6m$	$9m$	$15m$
JAC4/GS4	$4(m-2)$	$5(m-2)$	$9(m-2)$
SOR4	$5(m-2)$	$7(m-2)$	$12(m-2)$

The SOR4 algorithm reduces to the GS4 if $\omega = 1$. Except for very special cases, it is difficult to obtain the analytic expression for ω . According to Young [21], the precise determination of the optimal ω in the SOR is only known for a small class of matrices. The value of ω generally lies in the range of $1 < \omega < 2$.

The computational molecules for the JAC4 and the GS4 are as illustrated in Figures 4 and 5, respectively. The SOR4 has the same form as the GS4.

6. Computational Complexity

The cost of implementing an algorithm can be assessed by examining its computational complexity. The number of arithmetic operations such as additions (including subtractions) and multiplications (including divisions) that is needed to perform by the algorithm can be straightforwardly counted. Table 1 gives the number of sequential arithmetic operations per iteration that is required to evaluate the algorithms.

For the higher-order schemes, trade-off between accuracy and speed usually happens. The computational complexity has an effect on the efficiency of a particular scheme. Thus, this factor will be taken into account when discussing the results of the numerical experiments conducted in this study.

7. The Pseudocode of the IADEFM4 Sequential Algorithm

Algorithm 1 illustrates the pseudocode of the IADEFM4 sequential algorithm. This algorithm can also be generalized and applied to other methods discussed in this paper.

```

begin
  determine the parameters  $m, r, \Delta t, \Delta x, \lambda, \varepsilon, T$ 
  determine initial conditions  $u_i^{(p)}$ 
  determine exact solutions  $U_i$  at time level  $T$ 
  while (time level  $< T$ )
    determine boundary conditions
    for  $i = 2$  to  $i = m - 1$ 
      compute  $f_i$  (refer to (9))
    end-for
    set iteration = 0
    while (convergence conditions are not satisfied)
      for  $i = 2$  to  $i = m - 1$ 
        compute  $u_i^{(p+1/2)}$  (refer to (22))
      end-for
      for  $i = 2$  to  $i = m - 1$ 
        compute  $u_i^{(p+1)}$  (refer to (23))
      end-for
      test for convergence:
      compute  $e_i \leftarrow |u_i^{(p+1)} - u_i^{(p)}|$ 
      if  $(\max |e_i| < \varepsilon)$ 
        then  $u_i^{(p)} \leftarrow u_i^{(p+1)}$ 
        add 1 to iteration (if necessary)
      end-while
    end-while
    for  $i = 2$  to  $i = m - 1$ 
      determine average absolute error, root mean
      square error and maximum error
    end-for
  end

```

ALGORITHM 1: The IADEMF4 sequential algorithm.

8. Numerical Experiments

Two experiments were conducted to test the sequential numerical performance of the proposed IADEMF4 method against those of the benchmarked classical iterative methods, namely, the JAC4, the GS4, and the SOR4. Comparison is also made with the corresponding IADE method of the second order.

Experiment 1. This problem was taken from Saulev [22]

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}, \quad 0 \leq x \leq 1 \quad (44)$$

subject to the initial condition

$$U(x, 0) = 4x(1 - x), \quad 0 \leq x \leq 1 \quad (45)$$

and the boundary conditions

$$U(0, t) = U(1, t) = 0, \quad t \geq 0. \quad (46)$$

The exact solution to the given problem is given by

$$U(x, t) = \frac{32}{\pi^3} \sum_{k=1, (2)}^{\infty} \frac{1}{k^3} e^{-\pi^2 k^2 t} \sin(k\pi x). \quad (47)$$

Experiment 2. This problem was taken from Johnson and Reiss [23],

$$\frac{\partial U}{\partial t} = \frac{\partial^2 U}{\partial x^2}, \quad 0 \leq x \leq 1 \quad (48)$$

subject to the initial condition

$$U(x, 0) = \sin(\pi x)(1 + 6 \cos(\pi x)), \quad 0 \leq x \leq 1 \quad (49)$$

and the boundary conditions

$$U(0, t) = U(1, t) = 0, \quad t \geq 0. \quad (50)$$

The exact solution to the given problem is given by

$$U(x, t) = \sin(\pi x) e^{-\pi^2 t} + 3 \sin(2\pi x) e^{-4\pi^2 t}. \quad (51)$$

9. Results and Discussion

In each experiment, the exact solutions at $i = 1$ and $i = m$ were taken as boundary values for the IADEFM4 and the other fourth-order classical iterative methods. The convergence criterion used in the testing of each method was taken as $\|u^{(p+1)} - u^{(p)}\|_{\infty} \leq \varepsilon$, where ε is the convergence tolerance. The selections of the optimum r or ω were determined by experiments. As for the IADEFM2, the CN scheme with $\theta = 1/2$ was employed.

Figures 6, 7, 8, and 9 visualize the behavior of the one-dimensional parabolic heat solutions for both experiments. By using $m = 10$ and a tolerance requirement of $\varepsilon = 10^{-4}$, two different mesh sizes, $\lambda = 0.5$ and $\lambda = 1.0$, were considered in each experiment. The exact solutions are compared with the IADEFM4 and the IADEFM2 numerical solutions. The optimum value of r chosen for each method, as well as their corresponding outcome of the number of iterations, n , is stated in the legend of each figure. Every figure reveals that the IADEFM4 is more accurate than the IADEFM2 and the former converges with fewer numbers of iterations in comparison to the latter. The numerical solutions of the IADEFM4 seem to be in very good agreement with the exact solutions. For example, in Figure 7, at $x = 0.5$, the difference between the exact solution and the numerical solution using the IADEFM2 and using the IADEFM4 is about 3.6% and 0%, respectively. These results imply that the accuracy and convergence rate of the second-order IADEFM method are enhanced by the implementation of the fourth-order CN approximation that leads to the formation of the corresponding fourth-order IADEFM scheme.

Figure 10 displays the graph of $\log(\text{RMSE})$ versus $\log(\Delta x)$ for decreasing values of Δx implemented on Experiment 2. By considering $\varepsilon = 10^{-4}$ and fixing $\Delta t = 0.0001$, the value of Δx was initially taken as $\Delta x = 0.125$. It was then successively halved into $\Delta x/2$, $\Delta x/4$, and $\Delta x/8$.

The figure shows that amongst the tested methods, the root mean square error (RMSE) of the IADEFM4 and the IADEFM2 decreases linearly as the values of Δx decreases. The slope of the IADEFM4 is approximately equal to 4, which corresponds to its fourth-order spatial accuracy. The IADEFM2 with a second-order spatial accuracy has a slope that is approximately equal to 2. It is clear that the IADEFM4 is always more accurate than the IADEFM2, for the different considered mesh sizes. The graphs of the SOR4 ($\omega = 1.05$), the GS4, and the JAC4 show that their accuracies of fourth order tend to lack as Δx decreases, largely due to the effect of increasing round-off errors as the value of m increases.

Tables 2, 3, 4, 5, 6, 7, 8, and 9 provide numerical results in terms of the average absolute error (AAE), root mean square error (RMSE), maximum error (ME), number of iterations (n), and execution time (ET) measured in seconds (s). The results are obtained from both experiments for two different values of m and mesh size, λ . It is generally observed that when $m = 700$, $\lambda = 0.5$, and $\varepsilon = 10^{-6}$, the IADEFM4 has the least average absolute error, root mean square error, and maximum error in comparison with the other methods under consideration (Tables 2–5). When the size of m was

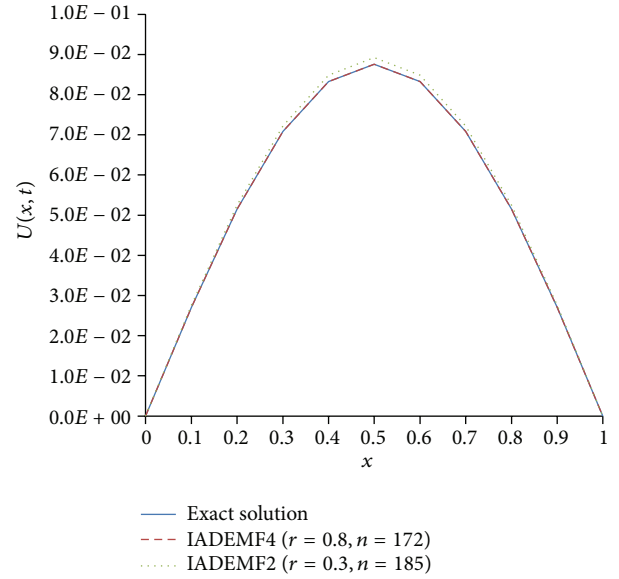


FIGURE 6: Numerical and exact solutions for Experiment 1. $\lambda = 0.5$, $\Delta x = 0.1$, $\Delta t = 0.005$, $t = 0.25$, and $\varepsilon = 10^{-4}$.

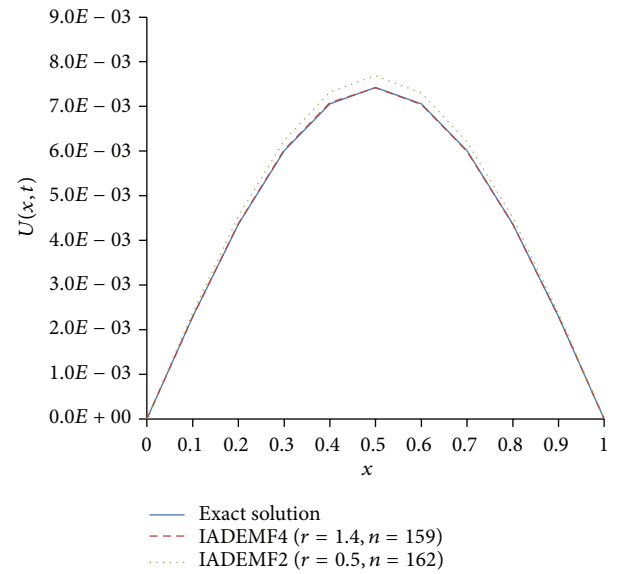


FIGURE 7: Numerical and exact solutions for Experiment 1. $\lambda = 1.0$, $\Delta x = 0.1$, $\Delta t = 0.01$, $t = 0.5$, and $\varepsilon = 10^{-4}$.

ten times bigger ($m = 7000$) and a more stringent tolerance criterion was set ($\varepsilon = 10^{-10}$), the accuracy of the IADEFM4 clearly outperforms the other methods for a mesh size of $\lambda = 0.5$ (Tables 6 and 8). However, the difference in the errors amongst the tested methods is not so obvious for the case of $\lambda = 1.0$ (Tables 7 and 9). The achievement of the fourth-order IADEFM method in Experiments 1 and 2 can be clearly seen from the results in Tables 2 and 8, respectively, where it has caused a huge 94% reduction of RMSE from its corresponding second-order IADEFM method.

The IADEFM4 has the advantage of featuring higher accuracy due to the execution of the fourth-order CN approximation coupled with the fourth-order accurate MF variant.

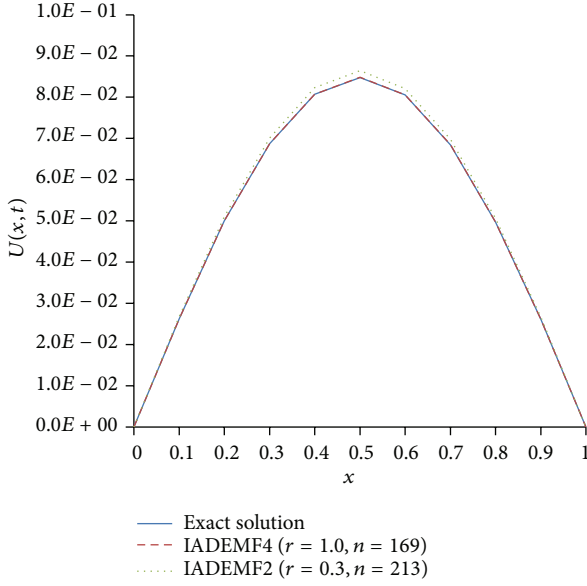


FIGURE 8: Numerical and exact solutions for Experiment 2. $\lambda = 0.5$, $\Delta x = 0.1$, $\Delta t = 0.005$, $t = 0.25$, and $\varepsilon = 10^{-4}$.

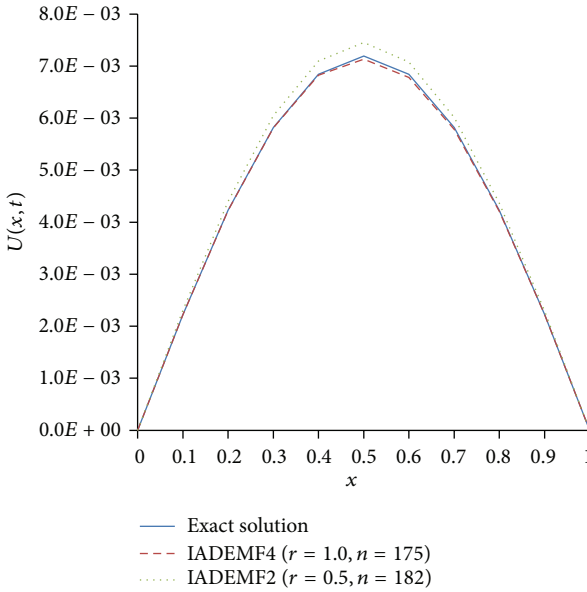


FIGURE 9: Numerical and exact solutions for Experiment 2. $\lambda = 1.0$, $\Delta x = 0.1$, $\Delta t = 0.01$, $t = 0.5$, and $\varepsilon = 10^{-4}$.

The IADEMF2 is only derived from the second-order CN approximation, but its combination with its corresponding fourth-order MF variant managed to produce errors which are better than the GS4, JAC4, and the SOR4. Even though the classical iterative methods are also derived from the fourth-order accurate CN type approximation, they are lacking in accuracy due to the round-off errors that have been accumulated from the time the execution starts till it ends.

With regards to the rate of convergence, the results from each table demonstrate that the number of iterations produced by the IADEMF2 and the fourth-order classical iterative methods is greater than or at least equal to that of

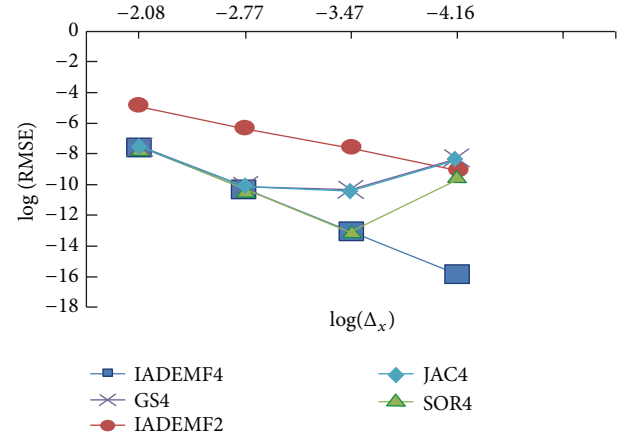


FIGURE 10: Log(RMSE) versus Log(Δx) Experiment 2. $\Delta t = 0.0001$, $t = 0.002$, and $\varepsilon = 10^{-4}$.

TABLE 2: Experiment 1. $m = 700$, $\lambda = 0.5$, $\Delta x = 1.43 \times 10^{-3}$, $\Delta t = 1.02 \times 10^{-6}$, $t = 5.09 \times 10^{-5}$, and $\varepsilon = 10^{-6}$.

Method	AAE	RMSE	ME	n	ET (s)
IADEMF4 ($r = 0.7$)	$2.67e-9$	$3.30e-9$	$2.12e-8$	100	$4.40e-2$
IADEMF2 ($r = 0.8$)	$1.88e-8$	$4.95e-8$	$3.23e-7$	100	$6.35e-3$
SOR4 ($\omega = 1.2$)	$5.40e-7$	$5.44e-7$	$5.52e-7$	108	$4.60e-2$
GS4	$5.40e-6$	$5.43e-6$	$3.32e-6$	150	$4.72e-2$
JAC4	$2.28e-5$	$2.29e-5$	$2.32e-5$	150	$4.73e-2$

TABLE 3: Experiment 1. $m = 700$, $\lambda = 1.0$, $\Delta x = 1.43 \times 10^{-3}$, $\Delta t = 2.04 \times 10^{-6}$, $t = 5.09 \times 10^{-5}$, and $\varepsilon = 10^{-6}$.

Method	AAE	RMSE	ME	n	ET (s)
IADEMF4 ($r = 1.0$)	$2.97e-7$	$3.01e-7$	$3.01e-7$	50	$2.01e-2$
IADEMF2 ($r = 0.9$)	$5.57e-7$	$5.59e-7$	$8.48e-7$	50	$3.17e-3$
SOR4 ($\omega = 1.2$)	$3.29e-6$	$3.31e-6$	$3.35e-6$	76	$2.25e-2$
GS4	$8.95e-6$	$9.00e-6$	$9.00e-6$	100	$2.38e-2$
JAC4	$2.14e-5$	$2.15e-5$	$2.17e-5$	125	$2.50e-2$

the IADEMF4. The convergence rate of the latter surpasses the others in the case of $m = 7000$ and $\varepsilon = 10^{-10}$ (Tables 6–8). Even though the operational count of the IADEMF4 is relatively quite large (Table 1), due to its higher level of accuracy, its increasing number of correct digits at each iteration causes it to converge at a faster rate. The IADEMF2 has the advantage of having less mathematical operations; thus, it is competitive in terms of convergence rate but at the expense of accuracy. The application of the fourth-order CN approximation on the heat equation proves that the three classical iterative methods also converge, with JAC4 appearing to be the slowest and the least accurate amongst all.

TABLE 4: Experiment 2. $m = 700$, $\lambda = 0.5$, $\Delta x = 1.43 \times 10^{-3}$, $\Delta t = 1.02 \times 10^{-6}$, $t = 5.09 \times 10^{-5}$, and $\varepsilon = 10^{-6}$.

Method	AAE	RMSE	ME	n	ET (s)
IADEMF4 ($r = 0.7$)	$2.07e-8$	$2.31e-8$	$3.45e-8$	100	$9.03e-3$
IADEMF2 ($r = 0.8$)	$1.06e-7$	$1.75e-7$	$1.75e-7$	100	$6.47e-3$
SOR4 ($\omega = 1.1$)	$1.54e-6$	$1.71e-6$	$2.55e-6$	200	$1.10e-2$
GS4	$2.94e-6$	$3.27e-6$	$4.87e-6$	250	$1.18e-2$
JAC4	$1.24e-5$	$1.38e-5$	$2.06e-5$	300	$1.27e-2$

TABLE 5: Experiment 2. $m = 700$, $\lambda = 1.0$, $\Delta x = 1.43 \times 10^{-3}$, $\Delta t = 2.04 \times 10^{-6}$, $t = 5.09 \times 10^{-5}$, and $\varepsilon = 10^{-6}$.

Method	AAE	RMSE	ME	n	ET (s)
IADEMF4 ($r = 0.9$)	$1.38e-6$	$1.54e-6$	$2.29e-6$	75	$6.50e-3$
IADEMF2 ($r = 0.5$)	$1.87e-6$	$2.07e-6$	$3.09e-6$	75	$4.72e-3$
SOR4 ($\omega = 1.2$)	$3.12e-6$	$3.47e-6$	$5.17e-6$	125	$9.06e-3$
GS4	$6.67e-6$	$7.42e-6$	$1.11e-5$	175	$9.32e-3$
JAC4	$1.26e-5$	$1.41e-5$	$2.10e-5$	250	$1.05e-2$

TABLE 6: Experiment 1. $m = 7000$, $\lambda = 0.5$, $\Delta x = 1.43 \times 10^{-4}$, $\Delta t = 1.02 \times 10^{-8}$, $t = 5.10 \times 10^{-7}$, and $\varepsilon = 10^{-10}$.

Method	AAE	RMSE	ME	n	ET (s)
IADEMF4 ($r = 0.8$)	$3.67e-10$	$4.33e-10$	$3.66e-9$	173	$1.46e-1$
IADEMF2 ($r = 0.5$)	$3.75e-10$	$4.35e-10$	$3.76e-9$	200	$1.24e-1$
SOR4 ($\omega = 1.12$)	$4.55e-10$	$5.07e-10$	$3.67e-9$	257	$1.47e-1$
GS4	$1.10e-9$	$1.11e-9$	$3.76e-9$	300	$1.60e-1$
JAC4	$2.30e-9$	$2.31e-9$	$3.78e-9$	400	$1.63e-1$

TABLE 7: Experiment 1. $m = 7000$, $\lambda = 1.0$, $\Delta x = 1.43 \times 10^{-4}$, $\Delta t = 2.04 \times 10^{-8}$, $t = 5.1 \times 10^{-7}$, and $\varepsilon = 10^{-10}$.

Method	AAE	RMSE	ME	n	ET (s)
IADEMF4 ($r = 1.0$)	$1.63e-7$	$1.63e-7$	$1.63e-7$	112	$9.60e-2$
IADEMF2 ($r = 1.2$)	$1.63e-7$	$1.63e-7$	$1.63e-7$	120	$7.36e-2$
SOR4 ($\omega = 1.15$)	$1.63e-7$	$1.63e-7$	$1.64e-7$	171	$9.99e-2$
GS4	$1.64e-7$	$1.64e-7$	$1.64e-7$	216	$1.07e-1$
JAC4	$1.65e-7$	$1.65e-7$	$1.65e-7$	312	$1.11e-1$

It is found that the convergence rate of the fourth-order and second-order methods improves with the application of larger mesh size, that is, from $\lambda = 0.5$ to $\lambda = 1.0$. The coarser meshes cause a reduction in the computational operations, thus giving better rate of convergence. However, a

TABLE 8: Experiment 2. $m = 7000$, $\lambda = 0.5$, $\Delta x = 1.43 \times 10^{-4}$, $\Delta t = 1.02 \times 10^{-8}$, $t = 5.10 \times 10^{-7}$, and $\varepsilon = 10^{-10}$.

Method	AAE	RMSE	ME	n	ET (s)
IADEMF4 ($r = 0.7$)	$5.67e-13$	$3.21e-12$	$1.30e-10$	150	$1.31e-1$
IADEMF2 ($r = 0.7$)	$1.99e-11$	$4.60e-11$	$1.82e-9$	200	$1.27e-1$
SOR4 ($\omega = 1.15$)	$2.45e-10$	$2.73e-10$	$3.99e-10$	260	$1.51e-1$
GS4	$3.98e-10$	$4.43e-10$	$6.61e-10$	400	$1.71e-1$
JAC4	$1.05e-9$	$1.17e-9$	$1.74e-9$	550	$1.76e-1$

TABLE 9: Experiment 2. $m = 7000$, $\lambda = 1.0$, $\Delta x = 1.43 \times 10^{-4}$, $\Delta t = 2.04 \times 10^{-8}$, $t = 5.1 \times 10^{-7}$, and $\varepsilon = 10^{-10}$.

Method	AAE	RMSE	ME	n	ET (s)
IADEMF4 ($r = 1.0$)	$1.54e-6$	$1.71e-6$	$2.55e-6$	96	$9.12e-2$
IADEMF2 ($r = 1.0$)	$1.54e-6$	$1.71e-6$	$2.55e-6$	96	$6.07e-2$
SOR4 ($\omega = 1.2$)	$1.54e-6$	$1.71e-6$	$2.55e-6$	192	$9.97e-2$
GS4	$1.54e-6$	$1.71e-6$	$2.55e-6$	288	$1.28e-1$
JAC4	$1.54e-6$	$1.72e-6$	$2.56e-6$	408	$1.46e-1$

more accurate solution is obtained by using finer mesh. For example, Table 6 shows that, for $\lambda = 0.5$, the IADEMF4 has $\text{RMSE} = 4.33e-10$ and $n = 173$, whereas Table 7 shows that, for $\lambda = 1.0$, its $\text{RMSE} = 1.63e-7$ and $n = 112$. In general, amongst all the tested methods, the IADEMF4 still maintains its greater accuracy characteristic, even with coarser meshes.

In terms of execution time, the results from every table display shorter execution time for the IADEMF2 in comparison to the IADEMF4. This is expected, since the IADEMF2 has lower computational complexity (Table 1). Despite the achievement in accuracy, the IADEMF4 has to perform more computational work than the IADEMF2 since the former has to utilize values of u at two grid points on either side of the point $(i\Delta x, k\Delta t)$ along the k th time level. Thus, if accuracy is desired, then the preferred sequential numerical algorithm would be the IADEMF4. On the other hand, if execution time matters, then the choice would be the IADEMF2.

Amongst the fourth-order methods, the IADEMF4 executes in the least amount of time. Even though its computational complexity is relatively large, it operates with the least number of iterations, thus enabling it to be the most efficient.

10. Conclusion

This paper proposes the development of the novel fourth-order IADEMF4 finite difference scheme. The higher-order scheme is developed by representing the spatial derivative in the heat equation with the fourth-order finite difference CN approximation. This leads to the formation of pentadiagonal matrices in the systems of linear equations with larger computational stencils. The algorithm also employs the higher accuracy of the Mitchell and Fairweather variant. Despite

the fourth-order IADE scheme's higher computational complexity, this technique is proved to be valuable because it enhances the accuracy of its second-order counterpart, namely, the IADEMF2. In addition, the higher accuracy of IADEMF4 is verified as a convergent and unconditionally stable scheme and is superior in terms of rate of convergence. It has also been proven to execute more efficiently in comparison to the other benchmarked fourth-order classical iterative methods, such as the GS4, the SOR4, and the JAC4. The increasing number of correct digits at each iteration serves as an advantage for the IADEMF4, thus yielding faster rate of convergence with higher level of accuracy.

In conclusion, the proposed IADEMF4 scheme affords users many advantages with respect to higher accuracy, stability, and rate of convergence, and it serves as an alternative, efficient technique for the solution of a one-dimensional heat equation with Dirichlet boundary conditions.

The proposed fourth-order scheme can be modified and adapted to more general multidimensional linear and nonlinear parabolic, elliptic, and hyperbolic partial differential equations, using different types of boundary conditions. Besides, it can also be considered for applications in problems that require higher-order accuracy with high resolution, such as problems in nanocomputing that require the solution of very large sparse systems of equation.

The explicit and high accuracy feature of the IADEMF4 can be exploited in two- and three-dimensional heat problems, by applying the scheme as pentadiagonal solvers for the system of linear equations arising in the sweeps of a higher-order alternating direction implicit (ADI) scheme. The lower-order ADI scheme was initially proposed by Peaceman and Rachford [24]. The higher space dimensions are expected not to cause unsatisfactory performance for the ADI-IADEMF4, as long as the proposed scheme is stable, convergent, and efficient as the process of its implementation continues to advance from one-time step to another. Pathirana et al. [25] implemented the ADI in developing a two-dimensional model for incorporating flood damage in urban drainage planning. The proposed model is found to be stable, numerically accurate, and computationally efficient. Mirzavand et al. [26] proposed the ADI-FDTD method for the physical modeling of high-frequency semiconductor devices. The approach is able to reduce significantly the full-wave simulation time.

It is possible to parallelize the IADEMF4 algorithm since the calculation of the new iteration only depends on the known values from the last iteration (Figures 2 and 3). Future work is to exploit the explicit computational properties and efficiency of the IADEMF4 for parallelization and execution on distributed parallel computing systems. The idea is to speed up the execution time without compromising its accuracy, especially on problems involving very large linear systems of equations.

References

- [1] J. Noye, *Computational Techniques for Differential Equations*, vol. 83, Elsevier Science, North-Holland Publishing, Amsterdam, The Netherlands, 1984.
- [2] G. W. Recktenwald, "Finite difference approximations to the heat equation," *Mechanical Engineering*, vol. 10, pp. 1–27, 2004.
- [3] P. Lax and B. Wendroff, "Systems of conservation laws," *Communications on Pure and Applied Mathematics*, vol. 13, pp. 217–237, 1960.
- [4] R. J. MacKinnon and G. F. Carey, "Analysis of material interface discontinuities and superconvergent fluxes in finite difference theory," *Journal of Computational Physics*, vol. 75, no. 1, pp. 151–167, 1988.
- [5] R. Hixon and E. Turkel, *High Accuracy Compact MacCormack-Type Schemes For Computational Aerocoustics*, NASA Centre for Aerospace Information, Lewis Research Centre, 1998.
- [6] Y. Lin, *Higher-order finite difference methods for solving heat equations [Ph.D. thesis]*, Southern Illinois University, 2008.
- [7] J. Sulaiman, M. K. Hasan, M. Othman M, and S. A. Abdul Karim, "Fourth-order QSMOR iterative method for the solution of one-dimensional parabolic PDE's," in *Proceedings of the International Conference on Industrial and Applied Mathematics (CIAM '10)*, pp. 34–39, ITB, Bandung, Indonesia, 2010.
- [8] N. Jha, "The application of sixth order accurate parallel quarter sweep alternating group explicit algorithm for nonlinear boundary value problems with singularity," in *Proceedings of the 2nd International Conference on Methods and Models in Computer Science (ICM2CS '10)*, pp. 76–80, December 2010.
- [9] Y. Jin, G. Jin, and J. Li, "A class of high-precision finite difference parallel algorithms for convection equations," *Journal of Convergence Information Technology*, vol. 6, no. 1, pp. 79–82, 2011.
- [10] W. Fu and E. L. Tan, "Development of split-step FDTD method with higher-order spatial accuracy," *Electronics Letters*, vol. 40, no. 20, pp. 1252–1253, 2004.
- [11] A. Mohebbi and M. Dehghan, "High-order compact solution of the one-dimensional heat and advection-diffusion equations," *Applied Mathematical Modelling*, vol. 34, no. 10, pp. 3071–3084, 2010.
- [12] W. Liao, "An implicit fourth-order compact finite difference scheme for one-dimensional Burgers' equation," *Applied Mathematics and Computation*, vol. 206, no. 2, pp. 755–764, 2008.
- [13] Z. F. Tian and Y. B. Ge, "A fourth-order compact ADI method for solving two-dimensional unsteady convection-diffusion problems," *Journal of Computational and Applied Mathematics*, vol. 198, no. 1, pp. 268–286, 2007.
- [14] C. Chun, "Some fourth-order iterative methods for solving nonlinear equations," *Applied Mathematics and Computation*, vol. 195, no. 2, pp. 454–459, 2008.
- [15] S. Zhu, Z. Yu, and J. Zhao, "A high-order parallel finite difference algorithm," *Applied Mathematics and Computation*, vol. 183, no. 1, pp. 365–372, 2006.
- [16] Z. Gao and S. Xie, "Fourth-order alternating direction implicit compact finite difference schemes for two-dimensional Schrödinger equations," *Applied Numerical Mathematics*, vol. 61, no. 4, pp. 593–614, 2011.
- [17] M. S. Sahimi, A. Ahmad, and A. A. Bakar, "The Iterative Alternating Decomposition Explicit (IADE) method to solve the heat conduction equation," *International Journal of Computer Mathematics*, vol. 47, pp. 219–229, 1993.
- [18] G. D. Smith, *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Oxford University Press, 2nd edition, 1978.
- [19] A. R. Mitchell and G. Fairweather, "Improved forms of the alternating direction methods of Douglas, Peaceman, and Rachford for solving parabolic and elliptic equations," *Numerische Mathematik*, vol. 6, no. 1, pp. 285–292, 1964.

- [20] B. N. Datta, *Numerical Linear Algebra and Applications*, Brooks/Cole Publishing, Pacific Grove, Calif, USA, 1995.
- [21] D. M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York, NY, USA, 1971.
- [22] V. K. Saulev, *Integration of Equations of Parabolic Type by the Method of Nets*, Pergamon Press, Oxford, UK, 1964.
- [23] L. W. Johnson and R. D. Riess, *Numerical Analysis*, Addison-Wesley, Reading, Mass, USA, 2nd edition, 1982.
- [24] D. W. Peaceman and H. H. Rachford, Jr., "The numerical solution of parabolic and elliptic differential equations," *Journal of the Society for Industrial and Applied Mathematics*, vol. 3, pp. 28–41, 1955.
- [25] A. Pathirana, S. Tsegaye, B. Gersonius, and K. Vairavamoorthy, "A simple 2-D inundation model for incorporating flood damage in urban drainage planning," *Hydrology and Earth System Sciences*, vol. 15, no. 8, pp. 2747–2761, 2011.
- [26] R. Mirzavand, A. Abdipour, G. Moradi, and M. Movahhedi, "Full-wave semiconductor devices simulation using ADI-FDTD method," *Progress in Electromagnetics Research*, vol. 11, pp. 191–202, 2010.