Thai Journal of Mathematics
Special Issue: The 17$^{\text{th}}$ IMT-GT ICMSA 2021
Pages 212–228 (2022)
http://thaijmath.in.cmu.ac.th

ISSN 1686-0209

Hybrid Finite Integration Method for Solving Partial Differential Equations

Nifatamah Makaje[1], **Areeyuth Sama-Ae**[1],**Aniruth Phon-On**[1],**Areena Hazanee**[1,*]

[1] *Department of Mathematics and Computer Science, Prince of Songkla University, Pattani campus, Pattani 94000*
e-mail : nifatamah.m@psu.ac.th (N. Makaje); areeyuth.s@psu.ac.th (A. Sama-Ae); aniruth.p@psu.ac.th (A. Phon-On); areena.h@psu.ac.th (A. Hazanee, corresponding author)*

Abstract In this paper, a Hybrid Finite Integration Method (HFIM), modified based on the research of P.H. Wen et al. (Wen et al. 2015), is presented to solve partial differential equations. The method requires the integral matrix, induced by the combination of the trapezoidal rule and Simpson's rule to produce the integral matrix. This paper also confirms the advantage of using HFIM that only one integral matrix is required to solve the n-th order differential equation. The integral matrix is straightforward to implement and not complicated. Moreover, in order to demonstrate the HFIM's accuracy and efficiency, we illustrate numerical examples and compared the results with the finite difference method and the traditional FIM.

MSC: 35G15; 37J35; 65N06; 65N22

Keywords: numerical method; differential equation; finite integration method; trapezoidal rule; Simpsons rule

Submission date: 15.03.2022 / Acceptance date: 31.03.2022

1. Introduction

Many problems in science, engineering and economics can be formulated as mathematical models involving partial differential equations (PDEs). Since the exact solutions of these equations in many cases are too complicated to determine in closed form, it is common using numerical methods to approximate solutions. There are many techniques to deal with the numerical solution of PDEs such as the finite difference method (FDM), finite element method (FEM), finite volume method (FVM), boundary element method (BEM) and spectral method.

In 2013 Wen et al. [1] proposed the finite integration method (FIM) to solve numerical solution of differential equations. The finite integration matrices of the first order can be constructed by implementing ordinary linear approach (OLA) or radial basis function (RBF) interpolation. These matrices of first order can be directly used to obtain finite integration matrices of higher order. Therefore, the advantage of the FIM is using only

*Corresponding author.

Published by The Mathematical Association of Thailand.

an (one layer) integral matrix to solve any n-th order differential equation. Moreover, the first order integral matrix of FIM has a form of lower triangular matrix so the FIM procedure is simple and saves the computational core memory. There are many researchers developed and modified the FIM to improve the accuracy of the approximated solutions. The common idea to improve the FIM is to construct the first order integral matrix using various numerical methods such as the trapezoidal rule [1–3], radial basis interpolating function [1, 2], Simpson's rule, Cotes integral formula, Lagrange formula [2], Chebyshev polynomial interpolation [4, 5], or Legendre polynomial interpolation [6]. The traditional- and modified-FIM have been successfully applied to solve differential equations in 1D as well as in multi-dimensional PDE. Moreover, the FIM has been implemented to deal with direct and indirect problem solving [7–11].

In 2020, Makaje et al. [12] modified FIM by combining the trapezoidal rule and the Simpsons rule in order to construct the first order integral triangular matrix to solve the initial value problem and boundary value problem. This hybrid method provided higher accuracy than the traditional FIM and FDM. However, this work has been only focus on one dimensional but not yet considering the PDE problems. This became the motivation for us to extend this hybrid FIM method to solve PDEs.

This paper is organized as follows. In Section 2, the hybrid FIM is introduced by expressing the integral matrix, A, induced by the combination of the trapezoidal rule and the Simpson's rule. The use of HFIM for solving the heat equation is presented in Section 3, whereas Section 4 constructs the case of the Poisson equation solved by HFIM. Several examples are studied in Section 5 to verify the accuracy and efficiancy of using HFIM to solved PDEs. Section 6 presents the conclusion of this study.

2. A Hybrid Finite Integration Method

In this section, we present the hybrid finite integration method (HFIM) which is the numerical method for approximating any n-layer integrations. This method is modified based on the traditional FIM, presented by P.H. Wen et al. [1], together with the trapezoidal rule and the Simpsons rule. We firstly mention the use of the trapezoidal rule and Simsons rule that the trapezoidal rule employ the first-order interpolating polynomial to connect two points equally spaced whereas the Simpsons rule use the second-order interpolating polynomial to connect three points equally spaced, then the number of interval of using the trapezoidal rule can be any positive integer whereas the Simsons rule has to be even number. The HFIM was first introduced in [12] by considering the finite integral from 0 to the spatial domain $x \in [0, L]$ defined as $F^{(1)}(x) := \int_0^x f(y)dy$. Let $\{0 = x_1 < x_2 < \ldots < x_N = L\}$ be a partition of the interval $[0, L]$, where $h = \dfrac{L}{N-1}$ and $x_i = (i-1)h$, for $i = 1, 2, 3, \ldots, N$. Then the first order approximated definite integral becomes,

$$F^{(1)}(x_k) := \int_{x_1}^{x_k} f(y)dy \quad \text{for} \quad k \in \{1, 2, \ldots, N\}. \tag{2.1}$$

When $k = 1$, , the definite integral from x_1 to x_1 equal to zero,

$$\int_{x_1}^{x_1} f(y)dy = 0.$$

When $k = 2$, the definite integral is approximated via the trapezoidal rule,

$$\int_{x_1}^{x_2} f(y)dy = \frac{h}{2}[f(x_1) + f(x_2)].$$

When $k > 2$, we first define

$$N_{odd} = \begin{cases} N, & \text{if } N \text{ is odd} \\ N-1, & \text{if } N \text{ is even} \end{cases} \quad \text{and} \quad N_{even} = \begin{cases} N, & \text{if } N \text{ is even} \\ N-1, & \text{if } N \text{ is odd} \end{cases}$$

and

i) For $k \in \{3, 5, 7, \ldots, N_{odd}\}$, the definite integral is approximated via the Simpsons rule as,

$$\int_{x_1}^{x_k} f(y)dy = \frac{h}{3}\left[f(x_1) + 4\sum_{i=1}^{(k-1)/2} f(x_{2i}) + 2\sum_{i=1}^{(k-1)/2-1} f(x_{2i-1}) + f(x_k)\right].$$

ii) For $k \in \{4, 6, 8, \ldots, N_{even}\}$, the definite integral is approximated via the combination of the trapezoidal rule and the Simpsons rule as,

$$\begin{aligned} \int_{x_1}^{x_k} f(y)dy &= \int_{x_1}^{x_{k-1}} f(y)dy + \int_{x_{k-1}}^{x_k} f(y)dy, \qquad k-1 \text{ is odd}, \\ &= \frac{h}{3}\left[f(x_1) + 4\sum_{i=1}^{(k-2)/2} f(x_{2i}) + 2\sum_{i=1}^{(k-4)/2} f(x_{2i-1}) + f(x_{k-1})\right] \\ &\quad + \frac{h}{2}\left[f(x_{k-1}) + f(x_k)\right] \\ &= \frac{h}{6}\left[2f(x_1) + 8\sum_{i=1}^{(k-2)/2} f(x_{2i}) + 4\sum_{i=1}^{(k-4)/2} f(x_{2i-1}) + 5f(x_{k-1}) + 3f(x_k)\right]. \end{aligned}$$

Therefore the first order definite integral $F^{(1)}(x_k) := \int_{x_1}^{x_k} f(y)dy$ can be rewritten in a matrix form as

$$\mathbf{F}^{(1)} = \mathbf{A}^{(1)}\underline{\mathbf{f}}, \tag{2.2}$$

where $\mathbf{F}^{(1)} = \left[\int_{x_1}^{x_1} f(y)dy, \int_{x_1}^{x_2} f(y)dy, \ldots, \int_{x_1}^{x_N} f(y)dy\right]^T$, $\underline{\mathbf{f}} = [f(x_1), f(x_2), \ldots, f(x_N)]^T$ and $\mathbf{A}^{(1)}$ is a coefficient matrix of the first order definite integral, and we denote $\mathbf{A}^{(1)} = \mathbf{A}$ which can be expressed in matrix form as

$$\mathbf{A} = \frac{h}{6}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 3 & 3 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 2 & 8 & 2 & 0 & 0 & \ldots & 0 & 0 \\ 2 & 8 & 5 & 3 & 0 & \ldots & 0 & 0 \\ 2 & 8 & 4 & 8 & 2 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 2 & 8 & 4 & 8 & 4 & 8 & 5 & 3 \end{bmatrix}_{N\times N} \qquad \text{for } N \text{ is even}$$

$$\mathbf{A} = \frac{h}{6}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 3 & 3 & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\ 2 & 8 & 2 & 0 & 0 & \dots & 0 & 0 & 0 \\ 2 & 8 & 5 & 3 & 0 & \dots & 0 & 0 & 0 \\ 2 & 8 & 4 & 8 & 2 & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 2 & 8 & 4 & 8 & 4 & \dots & 5 & 3 & 0 \\ 2 & 8 & 4 & 8 & 4 & \dots & 4 & 8 & 2 \end{bmatrix}_{N\times N} \qquad \text{for } N \text{ is odd.}$$

Furthermore by above consideration, for each $k \in \{1, 2, 3, \dots, N\}$ the integral $\int_{x_1}^{x_k} f(y)dy$ can be written as

$$\int_{x_1}^{x_k} f(y)dy = \sum_{j=1}^{k} a_{kj} f(x_j), \tag{2.3}$$

where a_{kj} is the element of matrix $\mathbf{A}^{(0)}$ corresponding to (2.2) for $j \in \{1, 2, 3, \dots, N\}$. For the second order integration, we define the definite integral as

$$F^{(2)}(x_k) = \int_{x_1}^{x_k} \int_{x_1}^{y^*} f(y)dydy^*.$$

Applying the first order definite integral in (2.3), we have

$$\int_{x_1}^{x_k} \int_{x_1}^{y^*} f(y)dydy^* = \sum_{i=1}^{k} \sum_{j=1}^{i} a_{ki} a_{ij} f(x_j).$$

Explicitly,
for $k = 2$, $\int_{x_1}^{x_2} \int_{x_1}^{y^*} f(y)dydy^* = (a_{21}a_{11} + a_{22}a_{21}) f(x_1) + a_{22}a_{22} f(x_2)$ and
for $k = 3$, $\int_{x_1}^{x_3} \int_{x_1}^{y^*} f(y)dydy^* = \sum_{k=1}^{3} (a_{3k}a_{k1}) f(x_1) + \sum_{k=2}^{3} (a_{3k}a_{k2}) f(x_2) + (a_{33}a_{33}) f(x_3)$.
In general, for $k = t$, where $t \in \{2, \dots, N\}$, we have

$$\int_{x_1}^{x_t} \int_{x_1}^{y^*} f(y)dydy^* = \sum_{k=1}^{t} a_{tk}a_{k1} f(x_1) + \sum_{k=2}^{t} a_{tk}a_{k2} f(x_2) + \dots + \sum_{k=t}^{t} a_{tk}a_{kt} f(x_t).$$

This second order definite integral $F^{(2)}(x_k) = \int_{x_1}^{x_k} \int_{x_1}^{y^*} f(y)dydy^*$ can also be rewritten in a matrix form as

$$\mathbf{F}^{(2)} = \mathbf{A}^{(2)}\underline{\mathbf{f}}, \tag{2.4}$$

where $\mathbf{A}^{(2)}$ is a coefficient matrix of the second order definite integral, and can be calculated as a multiplication matrix of $\mathbf{A}$ as

$$\mathbf{A}^{(2)} = \left[a_{ij}^{(2)}\right] = \left[\sum_{k=1}^{N} a_{ik}a_{kj} f(x_j)\right] = \mathbf{A} \cdot \mathbf{A} = \mathbf{A}^2.$$

Similarly, the n-order definite integral

$$F^{(n)}(x_k) = \underbrace{\int_{x_1}^{x_k} \cdots \int_{x_1}^{y^{(2)}} \int_{x_1}^{y^{(1)}} f(y)dydy^{(1)} \dots dy^{n-1}}_{n \text{ layers}}$$

can be expressed in the matrix form as

$$\mathbf{A}^{(n)} = \mathbf{A}^{(n)}\underline{\mathbf{f}} = \mathbf{A}^n\underline{\mathbf{f}}.$$

One thing to note that the coefficient matrix $\mathbf{A}^n$ is a lower triangular matrix for all n. This is an advantage of using HFIM since it can save the computational core memory.

3. The HFIM for Solving Heat Equation

This section describes the use of the HFIM to solve the heat equation. We propose to solve the time-dependent PDE by applying the FDM for the time discretization. For illustration, we here consider the heat equation together with the initial and boundary conditions:

$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t) + p(x,t)u(x,t) + f(x,t), \quad x \in (0,L), t \in (0,T) \tag{3.1}$$

$$\text{I.C.} \quad u(x,0) = \varphi(x), \quad x \in (0,L) \tag{3.2}$$

$$\text{B.C.} \quad \frac{\partial u}{\partial x}(0,t) = \mu_1(t), \quad \frac{\partial u}{\partial x}(L,t) = \mu_2(t), \quad t \in (0,T], \tag{3.3}$$

where $u(x,t)$ is the temperature at point $x \in [0,L]$ and time $t \in [0,T]$. For the discretization, we divide equally the time domain $[0,T]$ as $t_j \in [t_1, t_2, t_3, \dots, t_{N_t}]$ for $N_t - 1$ small subintervals $[t_{j-1}, t_j]; j \in \{2,3,\dots,N_t\}, t_j = (j-1)h_0$, and $h_0 = \dfrac{T}{N_t - 1}$. We use the backward FDM to approximate the time domain as

$$\frac{\partial u}{\partial t}(x,t_j) = \frac{u(x,t_j) - u(x,t_{j-1})}{h_0}.$$

Therefore the heat equation (3.1) can be discretized as

$$\frac{1}{h_0}\left[u(x,t_j) - u(x,t_{j-1})\right] = \frac{\partial^2 u}{\partial x^2}(x,t_j) + p(x,t_j)u(x,t_j) + f(x,t_j). \tag{3.4}$$

Here, the discretization of temperature $u(x,t)$ and given functions $p(x,t), f(x,t)$ in heat equation (3.4) are defined as

$$\begin{aligned} u(x,t) &\approx u(x_i,t_j) =: u_i^j \\ p(x,t) &\approx p(x_i,t_j) =: p_i^j \\ f(x,t) &\approx f(x_i,t_j) =: f_i^j, \end{aligned}$$

where $i \in \{1,2,\dots,N\}, j \in \{1,2,\dots,N_t\}$.

In order to apply HFIM for solving the discrete heat equation (3.4), we apply the integral over the equation (3.4) twice and then transform into the matrix form defined in (2.4). Thus we have

$$\frac{1}{h_0}\mathbf{A}^2\underline{u}^j - \frac{1}{h_0}\mathbf{A}^2\underline{u}^{j-1} = \underline{u}^j + \mathbf{A}^2\mathbf{P}^j\underline{u}^j + \mathbf{A}^2\underline{f}^j + c_0\underline{\mathbf{x}} + c_1\underline{\mathbf{i}}, \tag{3.5}$$

where $\underline{u}^j = \left[u_1^j, u_2^j, \ldots, u_N^j\right]^T$ is an N column vector of temperature solution at time t_j ,$\mathbf{P}^j = diag\left(p_1^j, p_2^j, \ldots, p_N^j\right)$ is an N diagonal matrix, $\underline{f}^j = \left[f_1^j, f_2^j, \ldots, f_N^j\right]^T$, $\underline{\mathbf{x}} = \left[x_1, x_2, \ldots, x_N\right]^T$, $\underline{\mathbf{i}} = [1, 1, \ldots, 1]^T$ are N column vectors of source function, space nodes and constant of 1, respectively, and c_0, c_1 are integral constants corresponding to the above equation. Rearranging equation (3.5) gives

$$\left[\frac{1}{h_0}\mathbf{A}^2 - \mathbf{I} - \mathbf{A}^2\mathbf{P}^j\right]\underline{u}^j = \frac{1}{h_0}\mathbf{A}^2\underline{u}^{j-1} + \mathbf{A}^2\underline{f}^j + c_0\underline{\mathbf{x}} + c_1\underline{\mathbf{i}}. \tag{3.6}$$

Next, we discretize the Neumann boundary condition (3.3) as

$$\frac{\partial u}{\partial x}(0, t_j) = \mu_1(t_j) =: \mu_1^j \quad \text{and} \quad \frac{\partial u}{\partial x}(L, t_j) = \mu_2(t_j) =: \mu_2^j$$

In order to apply the boundary condition, we take the integral once over the heat equation and approximate the integral via HFIM in (2.2). This yields

$$\frac{1}{h_0}\mathbf{A}\underline{u}^j - \frac{1}{h_0}\mathbf{A}\underline{u}^{j-1} = \frac{\partial u^j}{\partial x} + \mathbf{A}\mathbf{P}^j\underline{u}^j + \mathbf{A}\underline{f}^j + c_0\underline{\mathbf{i}}. \tag{3.7}$$

Considering the matrix equation (3.7) at $x = 0$, we have

$$\frac{1}{h_0}\sum_{i=1}^{N} a_{1i}^{(1)} u_i^j - \frac{1}{h_0}\sum_{i=1}^{N} a_{1i}^{(1)} u_i^{j-1} = \mu_1^j + \sum_{i=1}^{N} a_{1i}^{(1)} p_i^j u_i^j + \sum_{i=1}^{N} a_{1i}^{(1)} f_i^j + c_0.$$

This yields

$$c_0 = -\mu_1^j. \tag{3.8}$$

Plugging in $x = L$ and $c_0 = -\mu_1^j$ into equation (3.7), we get

$$\sum_{i=1}^{N}\left[\frac{1}{h_0}a_{Ni}^{(1)} - a_{Ni}^{(1)}p_i^j\right]u_i^j = \sum_{i=1}^{N} a_{Ni}^{(1)}\left(\frac{1}{h_0}u_i^{j-1} + f_i^j\right) + \mu_2^j - \mu_1^j. \tag{3.9}$$

From equations (3.6), (3.8) and (3.9), we obtain the following block matrix equation,

$$\left[\begin{array}{c|cc} \frac{1}{h_0}\mathbf{A}^2 - \mathbf{I} - \mathbf{A}^2\mathbf{P}^j & -\underline{\mathbf{x}} & -\underline{\mathbf{i}} \\ \hline \underline{0} & 1 & 0 \\ \underline{a}_N^{(1)}\left(\frac{1}{h_0}\mathbf{I} - \mathbf{P}^j\right) & 0 & 0 \end{array}\right]\left[\begin{array}{c} \underline{\mathbf{u}}^j \\ \hline c_0 \\ c_1 \end{array}\right] = \left[\begin{array}{c} \frac{1}{h_0}\mathbf{A}^2\underline{\mathbf{u}}^{j-1} + \mathbf{A}^2\underline{f}^j \\ \hline -\mu_1^j \\ \sum_{i=1}^{N} a_{Ni}^{(1)}\left(\frac{1}{h_0}u_i^{j-1} + f_i^j\right) + \mu_2^j - \mu_1^j \end{array}\right], \tag{3.10}$$

where $\underline{a}_N^{(1)} = \left[a_{N1}^{(1)}, a_{N2}^{(1)}, \ldots, a_{NN}^{(1)}\right]$ is an N row vector of integral coefficients $a_{Ni}^{(1)}$. In order to solve the solution of the problem (3.1)-(3.3), we use $\underline{\mathbf{u}}_0 := [u(x_i, 0)] = [\varphi(x_i)], i \in \{1, 2, \ldots, N\}$, for the initial temperature at time step $t_1 = 0$ or $j = 1$, as prescribed in the initial condition. At each time step $j = 2, \ldots, N_t$, for computing $\underline{\mathbf{u}}^j$, we use $\underline{\mathbf{u}}^{j-1}$ for previous time step, and then substitute it into the block matrix equation. Hence, at final time step $t_{N_t} = T$ or $j = N_t$, $\underline{\mathbf{u}}^{N_t}$ is the temperature solution as desired.

4. The HFIM for Solving Poisson Equation

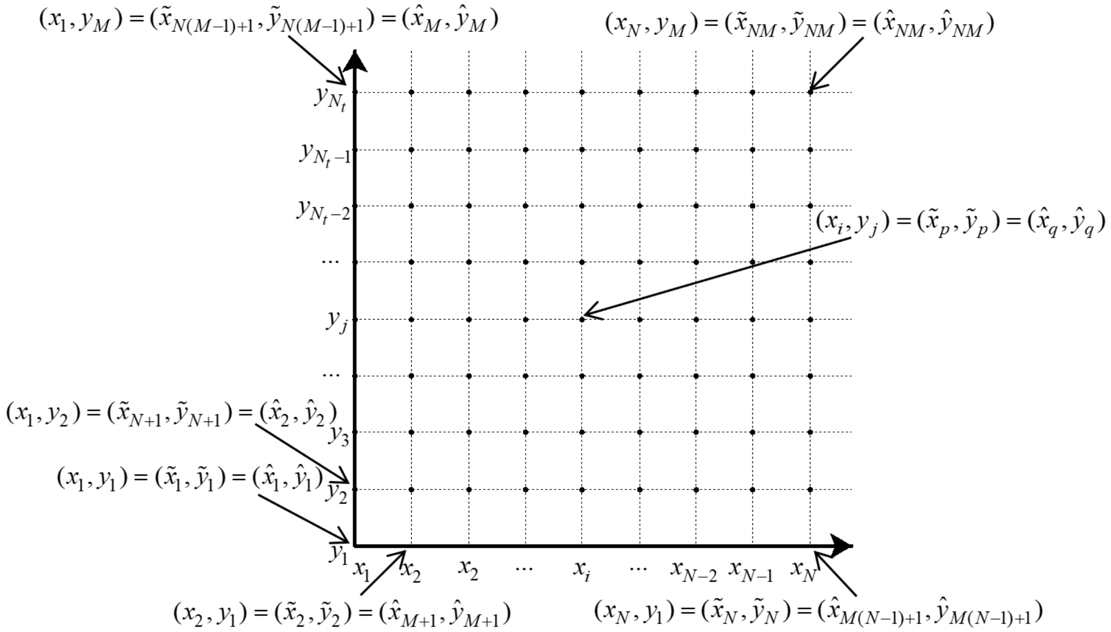
The previous section has presented the use of HFIM to solve the heat equation in 1D which is a parabolic equation. In this section we will extend the study in 2D elliptic equation by considering the Poisson equation together with the Dirichlet boundary conditions:

$$\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y) = f(x,y), \quad x \in (a,b), y \in (c,d) \tag{4.1}$$

$$\text{B.C.} \quad u(a,y) = \mu_1(y), \quad u(b,y) = \mu_2(y), \quad y \in [c,d] \tag{4.2}$$

$$u(x,c) = \varphi_1(x), \quad u(x,d) = \varphi_1(x), \quad x \in [a,b] \tag{4.3}$$

where $u(x,y)$ is a real function in rectangular domain $[a,b] \times [c,d]$. For the discretization procedure, we divide the domain $[a,b] \times [c,d]$ to the grid with mesh points $x \in \{x_1, x_2, ..., x_N\}$, $y \in \{y_1, y_2, ..., y_M\}$ and step sizes $h_x = \dfrac{b-a}{N-1}$ and $h_y = \dfrac{d-c}{M-1}$.



$(x_1, y_M) = (\tilde{x}_{N(M-1)+1}, \tilde{y}_{N(M-1)+1}) = (\hat{x}_M, \hat{y}_M)$
$(x_N, y_M) = (\tilde{x}_{NM}, \tilde{y}_{NM}) = (\hat{x}_{NM}, \hat{y}_{NM})$
y_{N_t}
y_{N_t-1}
y_{N_t-2}
$(x_i, y_j) = (\tilde{x}_p, \tilde{y}_p) = (\hat{x}_q, \hat{y}_q)$
...
y_j
...
$(x_1, y_2) = (\tilde{x}_{N+1}, \tilde{y}_{N+1}) = (\hat{x}_2, \hat{y}_2)$
y_3
$(x_1, y_1) = (\tilde{x}_1, \tilde{y}_1) = (\hat{x}_1, \hat{y}_1)$
y_2
y_1
x_1 x_2 x_2 ... x_i ... x_{N-2} x_{N-1} x_N
$(x_2, y_1) = (\tilde{x}_2, \tilde{y}_2) = (\hat{x}_{M+1}, \hat{y}_{M+1})$
$(x_N, y_1) = (\tilde{x}_N, \tilde{y}_N) = (\hat{x}_{M(N-1)+1}, \hat{y}_{M(N-1)+1})$

Figure 1. The grid points of global and local numbering systems

Here we use the global numbering system with the total point number $N \times M$ to index the local node (x_i, y_j) for $i \in \{1,2,...,N\}$ and $j \in \{1,2,...,M\}$. We define the global numbering system along X-axis as $(\tilde{x}_p, \tilde{y}_p)$ and the global numbering system along Y-axis as $(\hat{x}_q, \hat{y}_q)$ illustrating in Figure 1 for $p = N(j-1)+i$, $q = M(i-1)+j$ and $p, q \in \{1,2,...,NM\}$. Along the finite integration introduced in Section 2, the first order integral approximations with respect to x and y can be defined via using HFIM as $U^{(1,x)}(x,y) := \int_a^x u(\xi, y)\, d\xi$ and $U^{(1,y)}(x,y) := \int_c^y u(x,\zeta)\, d\zeta$. Therefore the integral approximations at the global points $\tilde{U}^{(1,x)}(\tilde{x}_p,\tilde{y}_p)$ can be expressed as

$$\tilde{U}^{(1,x)}(\tilde{x}_p,\tilde{y}_p)=\sum_{k=1}^{p}\tilde{a}_k^{(1,x)}\tilde{u}(\tilde{x}_k,\tilde{y}_k)$$

where $\tilde{a}_k^{(1,x)}$ defined as $\tilde{a}_k^{(1,x)}=a_{ir}$, when $i=p-N\left\lfloor\frac{p}{N}\right\rfloor, r=k-N\left\lfloor\frac{k}{N}\right\rfloor$ and $1\le r\le i$, otherwise $\tilde{a}_k^{(1,x)}=0$. Then the first order integral with respect to x can be rewritten in a matrix form as

$$\tilde{\mathbf{U}}^{(1,x)}=\tilde{\mathbf{A}}^{(1,x)}\underline{\tilde{u}},\tag{4.4}$$

where $\tilde{\mathbf{U}}^{(1,x)}=\left[\int_a^{\tilde{x}_1^*}u(\xi,\tilde{y}_1)\,d\xi,\int_a^{\tilde{x}_2^*}u(\xi,\tilde{y}_2)\,d\xi,...,\int_a^{\tilde{x}_{NM}^*}u(\xi,\tilde{y}_{NM})\,d\xi\right]^T$,

$$\underline{\tilde{u}}=[\tilde{u}(\tilde{x}_1,\tilde{y}_1),\tilde{u}(\tilde{x}_2,\tilde{y}_2),...,\tilde{u}(\tilde{x}_{NM},\tilde{y}_{NM})]^{\,T},$$

and $\tilde{\mathbf{A}}^{(1,x)}=\left[\underbrace{\begin{matrix}\mathbf{A}^{(1)}&0&\cdots&0\\0&\mathbf{A}^{(1)}&\cdots&0\\\vdots&\vdots&\ddots&0\\0&0&\cdots&\mathbf{A}^{(1)}\end{matrix}}_{M\text{ Blocks}}\right]_{NM\times NM}$ for $\mathbf{A}^{(1)}$ is a first-order integral matrix introduced in Section 2. Whereas the first order integral with respect to y at the global points $\hat{U}^{(1,y)}(\hat{x}_q,\hat{y}_q)$ can be expressed as

$$\hat{U}^{(1,y)}(\hat{x}_q,\hat{y}_q)=\sum_{k=1}^{q}\hat{a}_k^{(1,y)}\hat{u}(\hat{x}_k,\hat{y}_k),$$

where $\hat{a}_k^{(1,y)}$ defined as $\hat{a}_k^{(1,y)}=a_{js}$, when $j=q-M\left\lfloor\frac{q}{M}\right\rfloor, s=k-M\left\lfloor\frac{k}{M}\right\rfloor$ and $1\le s\le j$, otherwise $\hat{a}_k^{(1,y)}=0$. Then the first order integral with respect to y can be rewritten in a matrix form as

$$\hat{\mathbf{U}}^{(1,y)}=\hat{\mathbf{A}}^{(1,y)}\underline{\hat{u}},\tag{4.5}$$

where $\hat{\mathbf{U}}^{(1,y)}=\left[\int_c^{\hat{y}_1^*}u(\hat{x}_1,\zeta)\,d\zeta,\int_c^{\hat{y}_2^*}u(\hat{x}_2,\zeta)\,d\zeta,...,\int_c^{\hat{y}_{NM}^*}u(\hat{x}_{NM},\zeta)\,d\zeta\right]^T$,

$$\underline{\hat{u}}=[\hat{u}(\hat{x}_1,\hat{y}_1),\hat{u}(\hat{x}_2,\hat{y}_2),...,\hat{u}(\hat{x}_{NM},\hat{y}_{NM})]^{\,T},$$

and $\hat{\mathbf{A}}^{(1,y)}=\left[\underbrace{\begin{matrix}\mathbf{A}^{(1)}&0&\cdots&0\\0&\mathbf{A}^{(1)}&\cdots&0\\\vdots&\vdots&\ddots&0\\0&0&\cdots&\mathbf{A}^{(1)}\end{matrix}}_{N\text{ Blocks}}\right]_{NM\times NM}$.

Similarly as the double layer integration in Section 2, we have the second order integral with respect to x as follows.

$$\tilde{U}^{(2,x)}(\tilde{x}_p, \tilde{y}_p) = \int_a^{\tilde{x}_p^*} \int_a^{\xi_1} u(\xi, \tilde{y}_p)\, d\xi d\xi_1,$$

or the matrix form as

$$\tilde{\mathbf{U}}^{(2,x)} = \hat{\mathbf{A}}^{(2,x)} \underline{\tilde{u}}, \tag{4.6}$$

where $\tilde{\mathbf{U}}^{(2,x)} = \left[\int_a^{\tilde{x}_1^*} \int_a^{\xi_1} u(\xi, \tilde{y}_1)\, d\xi d\xi_1, \int_a^{\tilde{x}_2^*} \int_a^{\xi_1} u(\xi, \tilde{y}_2)\, d\xi d\xi_1, ..., \int_a^{\tilde{x}_{NM}^*} \int_a^{\xi_1} u(\xi, \tilde{y}_{NM})\, d\xi d\xi_1, \right]^T$

and $\tilde{\mathbf{A}}^{(2,x)} = \begin{bmatrix} \underbrace{\begin{matrix} \mathbf{A}^{(2)} & 0 & \cdots & 0 \\ 0 & \mathbf{A}^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \mathbf{A}^{(2)} \end{matrix}}_{M \text{ Blocks}} \end{bmatrix}_{NM \times NM}$ for $\mathbf{A}^{(2)}$ is a second-order integral matrix introduced in Section 2. And the second order integral with respect to y can be defined as follows

$$\hat{U}^{(2,y)}(\hat{x}_q, \hat{y}_q) = \int_c^{\tilde{y}_q^*} \int_c^{\zeta_1} u(\hat{x}_q, \zeta)\, d\zeta d\zeta_1,$$

or the matrix form as

$$\hat{\mathbf{U}}^{(2,y)} = \hat{\mathbf{A}}^{(2,y)} \underline{\hat{u}}, \tag{4.7}$$

where $\hat{\mathbf{U}}^{(2,y)} = \left[\int_a^{\hat{y}_1^*} \int_a^{\zeta_1} u(\hat{x}_1, \zeta)\, d\zeta d\zeta_1, \int_a^{\hat{y}_2^*} \int_a^{\zeta_1} u(\hat{x}_2, \zeta)\, d\zeta d\zeta_1, ..., \int_a^{\hat{y}_{NM}^*} \int_a^{\zeta_1} u(\hat{x}_{NM}, \zeta)\, d\zeta d\zeta_1, \right]^T$

and $\hat{\mathbf{A}}^{(2,y)} = \begin{bmatrix} \underbrace{\begin{matrix} \mathbf{A}^{(2)} & 0 & \cdots & 0 \\ 0 & \mathbf{A}^{(2)} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \mathbf{A}^{(2)} \end{matrix}}_{N \text{ Blocks}} \end{bmatrix}_{NM \times NM}$.

Likewise for the n-layer integration, we have

$$\tilde{U}^{(n,x)}(\tilde{x}_p, \tilde{y}_p) = \int_a^{\tilde{x}_p^*} ... \int_a^{\xi_2} \int_a^{\xi_1} u(\xi, \tilde{y}_p)\, d\xi d\xi_1 ... d\xi_{n-1},$$

and $\quad \hat{U}^{(n,y)}(\hat{x}_q, \hat{y}_q) = \int_c^{\tilde{y}_q^*} ... \int_c^{\zeta_2} \int_c^{\zeta_1} u(\hat{x}_q, \zeta)\, d\zeta d\zeta_1 ... d\zeta_{n-1}$

which can be written in matrix forms as $\tilde{\mathbf{U}}^{(n,x)} = \tilde{\mathbf{A}}^{(n,x)}\underline{\tilde{u}}$ and $\hat{\mathbf{U}}^{(n,y)} = \hat{\mathbf{A}}^{(n,y)}\underline{\hat{u}}$

$$\text{where} \quad \tilde{\mathbf{A}}^{(n,x)} = \left[\underbrace{\begin{matrix} \mathbf{A}^{(n)} & 0 & \cdots & 0 \\ 0 & \mathbf{A}^{(n)} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \mathbf{A}^{(n)} \end{matrix}}_{M \text{ Blocks}}\right]_{NM\times NM}$$

$$\text{and} \quad \hat{\mathbf{A}}^{(n,y)} = \left[\underbrace{\begin{matrix} \mathbf{A}^{(n)} & 0 & \cdots & 0 \\ 0 & \mathbf{A}^{(n)} & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \mathbf{A}^{(n)} \end{matrix}}_{N \text{ Blocks}}\right]_{NM\times NM} .$$

Turning to the point of solving the Poisson equation (4.1) by using HFIM, we first apply the four layer integral over the equation as

$$\begin{aligned} &\int\int\int\int \left[\frac{\partial^2 u}{\partial x^2}(x,y) + \frac{\partial^2 u}{\partial y^2}(x,y)\right] dxdxdydy \\ &\quad = \int\int\int\int f(x,y)\, dxdxdydy + C_{0,x}(y)x + C_{1,x} + C_{0,y}(x)y + C_{1,y} \end{aligned} \tag{4.8}$$

where $C_{0,x}(y)$, $C_{1,x}(y)$ are unknown functions of y obtaining from the above integration with respect to x and $C_{0,y}(x)$, $C_{1,y}(x)$ are unknown functions of x obtaining from the above integration with respect to y. These unknown functions assumed to be approximated by Taylor interpolation polynomial as follows,

$$\begin{aligned} C_{0,x}(y) &= c_{0,x}^{(1)} + yc_{0,x}^{(2)} + y^2c_{0,x}^{(3)} + ... + y^{M-1}c_{0,x}^{(M)}, \\ C_{1,x}(y) &= c_{1,x}^{(1)} + yc_{1,x}^{(2)} + y^2c_{1,x}^{(3)} + ... + y^{M-1}c_{1,x}^{(M)}, \\ C_{0,y}(x) &= c_{0,y}^{(1)} + xc_{0,y}^{(2)} + x^2c_{0,y}^{(3)} + ... + x^{N-1}c_{0,y}^{(N)}, \\ C_{1,y}(x) &= c_{1,y}^{(1)} + xc_{1,y}^{(2)} + x^2c_{1,y}^{(3)} + ... + x^{N-1}c_{1,y}^{(N)}. \end{aligned}$$

For convenience to rewrite the integral (4.8) as the matrix form with unknowns $\tilde{U}(\tilde{x}_p, \tilde{y}_p), \hat{U}(\hat{x}_q, \hat{y}_q), C_{0,x}(y), C_{1,x}(y), C_{0,y}(x)$ and $C_{1,y}(x)$, we are fixing on considering the global numbering system along X-axis. Therefore we transform all nodal points of integration in the global numbering system along Y-axis to the global numbering system along X-axis defined by $\tilde{\mathbf{A}}^{(2,y)} = T \cdot \hat{\mathbf{A}}^{(2,y)} \cdot T^{-1}$ where $T = [t_{k_1,k_2}]_{NM\times NM}$ is the transformation matrix defined by

$$t_{k_1,k_2} = \begin{cases} 1 & ; k_1 = N(j-1)+i,\ k_2 = M(i-1)+j, \\ 0 & ; \text{otherwise}, \end{cases}$$

for $i \in \{1,2,...,N\}$, $j \in \{1,2,...,M\}$.

Hence the matrix form of (4.8) via the global numbering system along X-axis can be written as

$$\left[\tilde{\mathbf{A}}^{(2,x)}+\tilde{\mathbf{A}}^{(2,y)}\right]\underline{\tilde{u}} = \tilde{\mathbf{A}}^{(2,x)}\tilde{\mathbf{A}}^{(2,y)}\underline{\tilde{f}}+X\Psi_y\underline{C}_{0,x}+\Psi_y\underline{C}_{1,x}+Y\Psi_x\underline{C}_{0,y}+\Psi_x\underline{C}_{1,x}, \tag{4.9}$$

where $\underline{\tilde{f}} = \left[\tilde{f}(\tilde{x}_1,\tilde{y}_1),\ \tilde{f}(\tilde{x}_2,\tilde{y}_2),\ ...,\ \tilde{f}(\tilde{x}_{NM},\tilde{y}_{NM})\right]^T$ is an NM column vector of function $f(x,y)$, $\underline{C}_{0,x}$, $\underline{C}_{1,x}$ are M column vectors of unknown constant value of $c_{0,x}^{(j)}$, $c_{1,x}^{(j)}$ for $j\in\{1,2,...,M\}$, $\underline{C}_{0,y}$, $\underline{C}_{1,y}$ are N column vectors of unknown constant value of $c_{0,y}^{(i)}$, $c_{1,y}^{(i)}$ for $i\in\{1,2,...,N\}$ and the coefficient matrices of unknown functions defined as

$$\Psi_y = \begin{bmatrix} \begin{bmatrix} 1 & y_1^1 & \cdots & y_1^{M-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_1^1 & \cdots & y_1^{M-1} \end{bmatrix} \\ \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \end{bmatrix} \\ \begin{bmatrix} 1 & y_M^1 & \cdots & y_M^{M-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & y_M^1 & \cdots & y_M^{M-1} \end{bmatrix} \end{bmatrix}_{NM\times M}, X\Psi_y = \begin{bmatrix} \begin{bmatrix} x_1 & x_1y_1^1 & \cdots & x_1y_1^{M-1} \\ \vdots & \vdots & \vdots & \vdots \\ x_N & x_Ny_1^1 & \cdots & x_Ny_1^{M-1} \end{bmatrix} \\ \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \end{bmatrix} \\ \begin{bmatrix} x_1 & x_1y_M^1 & \cdots & x_1y_M^{M-1} \\ \vdots & \vdots & \vdots & \vdots \\ x_N & x_Ny_M^1 & \cdots & x_Ny_M^{M-1} \end{bmatrix} \end{bmatrix}_{NM\times M}$$

$$\Psi_x = \begin{bmatrix} \begin{bmatrix} 1 & x_1^1 & \cdots & x_1^{N-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_1^1 & \cdots & x_1^{N-1} \end{bmatrix} \\ \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \end{bmatrix} \\ \begin{bmatrix} 1 & x_N^1 & \cdots & x_N^{N-1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_N^1 & \cdots & x_N^{N-1} \end{bmatrix} \end{bmatrix}_{NM\times N}, Y\Psi_y = \begin{bmatrix} \begin{bmatrix} y_1 & y_1x_1^1 & \cdots & y_1x_1^{N-1} \\ \vdots & \vdots & \vdots & \vdots \\ y_M & y_Mx_1^1 & \cdots & y_Mx_1^{N-1} \end{bmatrix} \\ \begin{bmatrix} \vdots & \vdots & \vdots & \vdots \end{bmatrix} \\ \begin{bmatrix} y_1 & y_1x_N^1 & \cdots & y_1x_N^{N-1} \\ \vdots & \vdots & \vdots & \vdots \\ y_M & y_Mx_N^1 & \cdots & y_Mx_N^{N-1} \end{bmatrix} \end{bmatrix}_{NM\times N}.$$

Next we discretize the Dirichlet boundary conditions (4.2) - (4.3) and apply to the matrix equation (4.9), this yields the following block matrix equation,

$$\left[\begin{array}{c|cccc} \tilde{\mathbf{A}}^{(2,x)}+\tilde{\mathbf{A}}^{(2,y)} & X\Psi_y & \Psi_y & Y\Psi_x & \Psi_x \\ \hline \mathbf{I}_a & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_b & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_c & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{I}_d & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{array}\right] \left[\begin{array}{c} \underline{\tilde{u}} \\ \hline \underline{C}_{0,x} \\ \underline{C}_{1,x} \\ \underline{C}_{0,y} \\ \underline{C}_{1,y} \end{array}\right] = \left[\begin{array}{c} \tilde{\mathbf{A}}^{(2,x)}\tilde{\mathbf{A}}^{(2,y)}\underline{\tilde{f}} \\ \hline \underline{\mu}_1 \\ \underline{\mu}_2 \\ \underline{\varphi}_1 \\ \underline{\varphi}_2 \end{array}\right], \tag{4.10}$$

where $\underline{\mu}_1$, $\underline{\mu}_2$ are M column vectors of boundary functions $\mu_1(y)$, $\mu_2(y)$,
$\underline{\varphi}_1$, $\underline{\varphi}_2$ are M column vectors of boundary functions $\varphi_1(x)$, $\varphi_2(x)$,
$\mathbf{I}_a$, $\mathbf{I}_b$ are $M\times NM$ matrices corresponding to the boundary on the left-hand

and right-hand sides of rectangular domain.

and $\mathbf{I}_c, \mathbf{I}_d$ are $N \times NM$ matrices corresponding to the boundary on the bottom and top sides of rectangular domain.

5. Numerical Examples

This section presents three benchmark test examples in order to demonstrate the HFIM's accuracy and efficiency. The first two examples are problems of heat equation with Neumann boundary conditions and initial condition. Whereas the last example is a problem of Poisson equation together with Dirichlet boundary conditions. Furthermore, we will compare results with the FDM and the traditional FIM.

5.1. Example 1

We consider the first example with the following heat problem,

$$\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t) + (x+t+1)u(x,t) - (2+x^2t+x^3)e^t, (x,t) \in (0,1)\times(0,1)$$

$$\text{I.C.} \quad u(x,0) = \varphi(x) = x^2, \quad x \in (0,1)$$

$$\text{B.C.} \quad \frac{\partial u}{\partial x}(0,t) = \mu_1(t) = 0, \quad \frac{\partial u}{\partial x}(1,t) = \mu_2(t) = 2e^t, \quad t \in (0,1].$$

The analytical solution of this example is $u(x,t) = x^2e^t$ which is used for testing the accuracy of the approximation with RMSE. In order to test the accuracy of the approximation, let us introduce the root mean square error (RMSE) defined as

$$\text{RMSE}(u(x,t_j)) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(v_i^j - u_i^j\right)}, \tag{5.1}$$

where v_i^j and u_i^j represent the analytical and numerical solutions of $u(x_i,t_j)$, respectively. In order to apply HFIM to solve this example, we substitute known functions and parameters, i.e. $\mathbf{P}^j, \underline{f}^j, \mu_1^j, \mu_2^j, h_0$, and others into equation (3.10). Then the approximate solution can be obtained by solving the system of linear equation (3.10). Table 1 shows the RMSEs of the temperature $u(x,1)$ obtained by using HFIM with $N, N_t \in \{20,40,60,80,100\}$.

Table 1. The RMSEs of the temperature $u(x,1)$ obtained by using HFIM with $N, N_t \in \{20,40,60,80,100\}$, for Example 1.

N	20	40	60	80	100
N_t	20	40	60	80	100
RMSE(u)	5.3742E-2	2.4067E-2	1.5505E-2	1.1437E-2	9.0593E-3
N	20	40	60	80	100
N_t	100	100	100	100	100
RMSE(u)	9.0630E-3	9.0605E-3	9.0599E-3	9.0595E-3	9.0593E-3
N	100	100	100	100	100
N_t	20	40	60	80	100
RMSE(u)	5.3723E-2	2.4064E-2	1.5504E-2	1.1436E-2	9.0593E-3

It is obvious to see that when increased the number of time subintervals N_t , the RMSE(u) decrease. Here, with $N_t = 100$ among various $N \in \{20,40,60,80,100\}$ gives the smallest errors, i.e. RMSE$(u) \approx 8.9 \times 10^{-3}$. This means that by using small spacing subinterval $N_t = 100$ and $N = 20$ can yield the small error among those $N \in \{20, 40, 60, 80, 100\}$. The illustration of both analytical and approximation solution $u(x,t)$ obtained using $N = 20$ and $N_t = 100$ can be seen in Figure 2 that the agreement between the numerical and the analytical solutions is excellent.

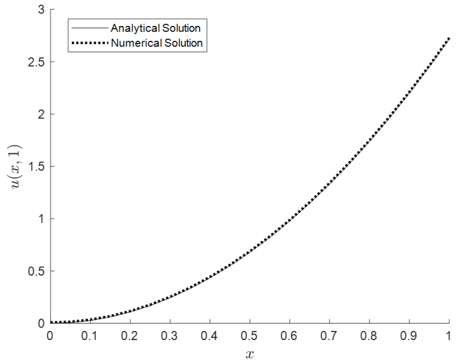


FIGURE 2. The analytical (—) and numerical (···) solutions of $u(x,1)$ obtained using HFIM with $N = 20$ and $N_t = 100$, for Example 1.

5.2. Example 2

The previous Example 1 displayed that the HFIM can solve the time dependent PDE with very good performance observed by RMSEs and the illustration of the solution. In this example, we are considering the heat equation together with the initial and boundary conditions solved by using HFIM, in comparison with the FDM and the traditional FIM. Note that the traditional FIM presented in [1] is constructed based on the FIM with ordinary linear approximation (OLA) namely the trapezoidal rule over the space domain $[0, L]$, whereas the FDM used here is backward difference in time and central difference in space.

The following heat problem is considered in Example 2,

$$\begin{aligned}
&\frac{\partial u}{\partial t}(x,t) = \frac{\partial^2 u}{\partial x^2}(x,t) - (2+4x^2)u(x,t) + \cos(t)e^{x^2}, \qquad (x,t) \in (0,1)\times(0,1)\\
&\text{I.C.} \quad u(x,0) = \varphi(x) = 0, \quad x \in (0,1)\\
&\text{B.C.} \quad \frac{\partial u}{\partial x}(0,t) = \mu_1(t) = 0, \quad \frac{\partial u}{\partial x}(1,t) = \mu_2(t) = 2e\sin(t), \quad t \in (0,1].
\end{aligned}$$

The analytical solution of this example is $u(x,t) = \sin(t)e^{x^2}$. For using HFIM to this example, all known functions and parameters are applied to equation (3.10) and solve the block matrix equation in order to get the approximate solution. The RMSEs of the numerical result $u(x,1)$ obtained using HFIM, FDM and traditional FIM are tabulated in Table 2 with $N_t \in \{10, 20, 30, \ldots, 100\}$ and the number of grid points is fixed as $N = 10$.

From Table 2, it can be observed that the RMSEs obtained using HFIM decrease with increasing of the number of time N_t, i.e. from RMSE$(u) \approx 1 \times 10^{-2}$ to RMSE$(u) \approx 5 \times 10^{-3}$ for $N_t = 10$ to $N_t = 100$, respectively. Whereas, the RMSEs obtained using FDM and traditional FIM are playing around RMSE$(u) \approx 1 \times 10^{-2}$ for all $N_t \in \{10, 20, 30, \ldots, 100\}$.

Table 2. The RMSEs of the temperature $u(x, 1)$ obtained by using HFIM, FDM and traditional FIM with $N_t \in \{10, 20, 30, \ldots, 100\}$ and $N = 10$, for Example 2.

N	N_t	HFIM	FDM	Traditional FIM
10	10	1.8396E-2	2.5463E-2	2.5279E-2
10	20	1.0785E-2	1.8299E-2	1.7663E-2
10	30	8.3658E-3	1.6107E-2	1.5240E-2
10	40	7.1785E-3	1.5057E-2	1.4050E-2
10	50	6.4739E-3	1.4443E-2	1.3343E-2
10	60	6.0075E-3	1.4041E-2	1.2874E-2
10	70	5.6761E-3	1.3758E-2	1.2541E-2
10	80	5.4286E-3	1.3548E-2	1.2292E-2
10	90	5.2367E-3	1.3385E-2	1.2098E-2
10	100	5.0836E-3	1.3256E-2	1.1944E-2

Additionally, we consider the converging performance under the error tolerance $\epsilon_s = 1 \times 10^{-2}$. Among the methods tested, i.e. HFIM, FDM and the traditional FIM, we found that the HFIM converges fastest with minimum number of steps taken at $N_t = 30$. Increasing the number of nodes M results in more accurate solutions $u(x, 1)$ for all methods but the RMSEs of the FDM and the traditional FIM are not lower than error tolerance when $N_t = 100$. We, also, note here that the algorithm of the FDM is relatively stable even for a large number of nodes N_t, the errors are at the accuracy degree of 10^{-2} when $N_t \in \{10, 20, 30, \ldots, 100\}$.

5.3. Example 3

We now consider the use of HFIM to solve the Poisson equation together with the Dirichlet boundary conditions and compare our results with results obtained by the FDM and the traditional FIM. The root mean square error (RMSE) employed in this example is defined as

$$\mathrm{RMSE}(\tilde{u}(\tilde{x}, \tilde{y})) = \sqrt{\frac{1}{NM} \sum_{p=1}^{NM} (\tilde{v}_p - \tilde{u}_p)}, \tag{5.2}$$

where $\tilde{v}_p$ and $\tilde{u}_p$ represent the analytical and numerical solutions of $\tilde{u}(\tilde{x}_p, \tilde{y}_p)$, respectively, via the global numbering system along X-axis.

We consider the following Poisson equation with $a = c = 0$ and $b = d = 1$,

$$\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) = -2x^2 \sin(\pi x) \sin(\pi y),$$

subject to the boundary conditions $u(0, y) = u(1, y) = u(x, 0) = u(x, 1) = 0$. The analytical solution of this boundary value problem is $u(x, y) = \sin(\pi x) \sin(\pi y)$. In this example we present numerical results obtained with $N = M \in \{5, 10, 15\}$.

We have tried solving system of linear equations (4.10) and found that this system is ill-conditioned since the condition number of the left-hand side matrix herein is large. The singular value decomposition (SVD) technique needs to be applied here to solve the linear system. In this study we operate this decomposition by using command in

MATLAB namely '*svd*', i.e. $[U, S, V] = svd(X)$, which produces matrices U, S, V that $X = U \times S \times V^T$, whereas S is a diagonal matrix with nonnegative diagonal elements in decreasing order and having the same dimension as matrix X, and U, V are unitary matrices with the size corresponding to $X = U \times S \times V^T$. One thing can be noted that if the inaccurate results is found by using SVD technique, then in order to improve the accuracy we suggest to truncate the diagonal matrix S by omitting its last some small singular values. This method is called the truncated SVD (TSVD), and the MATLAB command for the TSVD is '*svds*', see more in [13].

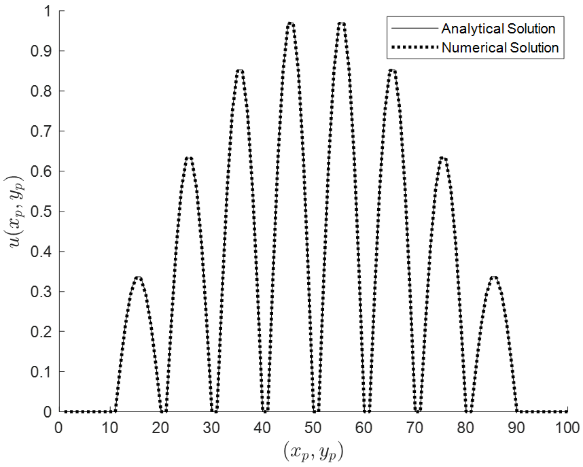


FIGURE 3. The analytical (——) and numerical (⋯⋯) solutions of $\tilde{u}(\tilde{x}_p, \tilde{y}_p)$ obtained using HFIM with $N = M = 10$, for Example 3.

Firstly we solve the problem with $N = M = 10$ and illustrate the analytical and numerical results of $\tilde{u}(\tilde{x}_p, \tilde{y}_p)$ shown in Figure 3. It indicated that the HFIM can be used to solve the Poisson equation with Dirichlet boundary conditions accurately and efficiently with RMSE($\tilde{u}$) $= 1.7305 \times 10^{-3}$. The RMSEs of numerical results with $N = M \in \{5, 10, 15\}$ are shown in Table 3 in comparison with numerical results obtained by the FDM and the traditional FIM. From this table it can be observed that, as expected, the accuracies of all results are improved when numbers of N and M are increased. Whereas the results obtained using the traditional FIM is less accurate than both results obtained using HFIM and FDM. With $N = M = 15$, it can be clearly seen that the result obtained using the HFIM is higher, with RMSE($\tilde{u}$) $\approx 10^{-4}$, than results obtained using the traditional FIM and FDM, with RMSE($\tilde{u}$) $\approx 10^{-3}$.

TABLE 3. The RMSEs of the solution $\tilde{u}(\tilde{x}_p, \tilde{y}_p)$ obtained by using HFIM, FDM and traditional FIM with $N = M \in \{5, 10, 15\}$, for Example 3.

$M = N$	HFIM	FDM	Traditional FIM
5	1.3904E-2	2.1212E-2	4.0473E-2
10	1.7305E-3	4.5972E-3	9.1105E-3
15	4.4113E-4	1.9632E-3	3.9116E-3

6. Conclutions

In this study, the HFIM has been developed to solve partial differential equations, particularly for the heat equation and Poisson equation. The use of HFIM together with FDM has been considered to solve the heat equation subjected to the Neumann boundary conditions and the initial condition. Whereas the Poisson equation subjected to the Dirichlet boundary conditions has been solved by using HFIM with transforming the numbering system from local to global. The numerical results obtained using HFIM have been investigated with very good agreement with the analytical solution in both problems of the heat equation and Poisson equation. Moreover, the accuracy of the solution obtained using HFIM found higher than that results obtained by using both FDM and traditional FIM. For the application of HFIM to solve wave equation can be studied in future work since this equation is more complicate to be solved than the heat equation or Poisson equation as it is time-dependent with two dimensions.

ACKNOWLEDGEMENTS

Authors would like to acknowledge the financial support received from the Office of Research Affairs of the Faculty of Science and Technology, Prince of Songkla University, Pattani campus, Thailand, under the Research Career Development Grant 2020.

References

[1] P.H. Wen, Y.C. Hon, M. Li, T. Korakianitis, Finite integration method for partial differential equations, Applied Mathematical Modelling, Appl Math Model. 37 (2013) 10092-10106.

[2] M. Li, C.S. Chen, Y.C. Hon, P.H. Wen, Finite integration method for solving multi-dimensional partial differential equations, Appl Math Model. 39 (2015) 4979-4994.

[3] M. Li, Z.L. Tian, Y.C. Hon, C.S. Chen, P.H. Wen, Improved finite integration method for partial differential equations, Eng Anal Bound Elem. 64 (2016) 230-236.

[4] R. Boonklurb, A. Duangpan, T. Treeyaprasert, Modified finite integration method by using Chebyshev polynomial for solving linear differential equations, J Numer Anal Ind Appl Math. 12 (2018) 1-19.

[5] A. Duangpan, R. Boonklurb, T. Treeyaprasert, Finite integration method with shifted Chebyshev polynomials for solving Time-Fractional Burgers equations, Mathematics. 7 (2019) 1-24.

[6] T. Sahakitchatchawan, R. Boonklurb, S. Singhum, Modified finite integration method by using Legendre polynomials for solving linear ordinary differential equations, Chamchuri J Math.10 (2018) 14-27.

[7] M. Li, Y.C. Hon, T. Korakianitis, P.H. Wen, Finite integration method for nonlocal elastic bar under static and dynamic loads, Eng Anal Bound Elem. 37 (2013) 842-849.

[8] Y. Yu, D. Xu, Y.C. Hon, Reconstruction of inaccessible boundary value in a side ways parabolic problem with variable coefficients-Forward collocation with finite integration method, Eng Anal Bound Elem. 61 (2015) 78-90.

[9] R. Lesmana, A. Hazanee, A. Phon-On, J. Saelee, A finite integration method for a time-dependent heat source identification of inverse problem, Proc 5th AASIC. (2017) 444-451.

[10] A. Hazanee, Finite integration method for the time-dependent heat source determination of inverse problem, Proc 6th BUU Conf. (2017) 391-401.

[11] A. Hazanee, Finite integration method based on a trapezoid rule for solving inverse source problem for the heat equation with a non-local boundary condition, Burapha Sci J. 23 (2018) 1345-1358.

[12] N. Makaje, A. Hazanee, A. Phon-on, Hybrid finite integration method for solving ordinary differential equations, Proc 46th STT Conf. (2020) 323-333.

[13] A. Hazanee, D. Lesnic, The boundary element method for solving an inverse time-dependent source problem, Proc 9th UK Conf. (2013) 5562.