Research Article

Two Numerical Methods for Solving the Schrödinger Parabolic and Pseudoparabolic Partial Differential Equations

Mahmut Modanli 1, Bushra Bajjah 2, and Sevgi Kuşulay 1,2

1 Harran University, Faculty of Arts and Sciences, Department of Mathematics, Sanliurfa 63300, Turkey
2 Thamar University, Faculty of Applied Sciences, Department of Mathematics, Dhamar, Yemen

Correspondence should be addressed to Bushra Bajjah; bushra.bajjah@tu.edu.ye

Received 4 November 2021; Revised 17 January 2022; Accepted 4 February 2022; Published 21 March 2022

Academic Editor: Zine El Abiddine Fellah

Copyright © 2022 Mahmut Modanli 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.

In this work, the initial-boundary value problems for one-dimensional linear time-dependent Schrödinger parabolic and pseudoparabolic partial differential equations are studied. The modified double Laplace decomposition method is applied to get the semianalytic solutions and the explicit finite difference method to get the approximate solutions of the problems. The von Neumann stability analysis of the presented problems is also investigated.

1. Introduction

The parabolic partial differential equations (PPDEs) take place in several areas of applied mathematics, for instance in the heat diffusion equation and in fluid mechanics [1]. Pseudoparabolic partial differential equations (PPPDEs) arise in modelling various phenomena, such as diffusion, heat conduction, thermoelectricity, chemical engineering, subsurface water flow, population dynamics, and plasma physics [2]. Several analytical approximation methods have been structured to treat such problems. For instance, the modified double Laplace decomposition method (MDLDM) has been used to solve the singular one-dimensional PPPDE with initial conditions by Gadain [3]. Ilhan et al. [4] constructed a family of travelling wave solutions to the nonlinear PPPDEs utilizing the modified exponential function method. In [5], the authors employed the variational iteration method with He’s polynomials to solve a coupled PPPDE. A three-layer finite difference scheme (FDS) to a linear PPPDE with a delay has been dealt by Amirali and Gabil [6]. In [7], Yang has studied a finite volume element approximation of PPPDEs in three spatial dimensions. Singh et al. [8] solved approximately the two-dimensional diffusion equation with Dirichlet boundary conditions using an Euler matrix method. The authors in [9] studied a nonlinear PPPDE with initial and Dirichlet boundary conditions using spectral discretization in space. Huntul et al. [10] solved the inverse initial-Neumann boundary value problem for a third-order PPPDE using the CB-spline collocation method. In [11], the researcher used a finite time blow-up criterion to solve a semilinear PPPDE. The Jacobi pseudospectral method has been used to solve numerically the two-dimensional linear heat diffusion equations subject to the Neumann and Robin boundary conditions by Yang et al. [12].

The famous time-dependent one-dimensional linear Schrödinger partial differential equation (SPDE) in quantum physics is known as,

\[ i\zeta u_t(t, x) = -\frac{\zeta^2}{2m}u_{xx}(t, x) + f(t, x)u(t, x), \]

where \( i = \sqrt{-1} \) is the imaginary number, \( \zeta \) is Planck’s constant, and \( u(t, x) \) and \( f(t, x) \) represent the wave function and the potential function, respectively. Equation (1) has been reformulated by the use of the path integral approach considering the Gaussian probability distribution by Feynman and Hibbs [13].

In this work, an initial-boundary value problem for one-dimensional time-dependent linear Schrödinger parabolic
and pseudoparabolic partial differential equation in the domain $[0, T] \times [0, L]$ is considered as follows:

\[
\begin{aligned}
\begin{cases}
    iu_t (t, x) = au_{xxx} (t, x) + u_{xxx} (t, x) + f (t, x), & \alpha \geq 0, 0 < x < L, 0 < t < T, \\
    u(0, x) = \psi(x), & 0 \leq x \leq L, \\
    u(t, 0) = u(t, L) = 0, & 0 \leq t \leq T,
\end{cases}
\end{aligned}
\]

where $f$ and $\psi$ are known sufficiently smooth functions, which may be complex-valued. Equation (2) is referred to as a Schrödinger parabolic partial differential equation (SPPDE) when $\alpha = 0$, and as a Schrödinger pseudoparabolic partial differential equation (SPPPDE) when $\alpha > 0$.

The MDLDM will be employed to find the semianalytic solution of problem (2). Many researchers consider it to be one of the best integral transforms. Jaradat et al. [14] used a Laplace–Adomian decomposition method (LADM) to obtain the semianalytical solution of the nonlinear SPDE with a harmonic oscillator in one and two dimensions. A nonlinear SPDE has been solved utilizing a double Laplace decomposition method by Gündoğdu and Ömer F [15]. The double Laplace transform of $u(t, x)$ is called MDLDM.

On the other hand, the approximate solution of problem (2) will be obtained using the explicit FDS. Ghafouri et al. [16] have applied FDS to find the approximate solution of SPDE in a nanoscale side-contacted field effect diode. For solving multi-applied FDS to find the approximate solution of SPDE in a one-dimensional case, the MDLDM will be employed to find the semianalytic solution for problems of type (2). In Section 3, the MDLDM is explained with two examples to get the semianalytic solution for problems of type (2).

2. Preliminaries on Double Laplace Transform

Some important definitions and theorems of the double Laplace transform are given which are used further in this work.

**Definition 1.** Let us assume $u(t, x)$, a function of two variables $t$ and $x$. The double Laplace transform of $u(t, x)$ is defined by the following double improper integral:

\[
L_s L \alpha [u(t, x)] := U(s, \alpha) = \int_0^\infty \int_0^\infty e^{-(s+\alpha)p} \, dx \, dt
\]

\[
= \lim_{n,m \to \infty} \int_0^n \int_0^m e^{-(s+\alpha)p} \, dx \, dp,
\]

where $s$ and $\alpha$ are complex numbers and $\Re (s) > 0$ and $\Re (\alpha) > 0$.

**Definition 2.** The inverse double Laplace transform of $U(s, \alpha)$ is defined as follows:

\[
u(t, x) = L^{-1}_s L^{-1}_\alpha [U(s, \alpha)] = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} e^{\pi i \frac{1}{2}t} \int_{\delta-i\infty}^{\delta+i\infty} e^{\pi x U(s, \alpha) \, dp},
\]

where $\sigma$ and $\delta$ are large enough that $U(s, \alpha)$ is defined for the real parts of $s > \sigma$ and $\alpha > \delta$.

**Theorem 1.** If $u(t, x)$ is of an exponential order and a piecewise continuous function on $(0, T)$ and $(0, X)$, then its double Laplace transform exists for $\Re (s) > 0$ and $\Re (\alpha) > 0$.

**Theorem 2.** Let $u \in C^r (\mathbb{R}^+ \times \mathbb{R}^+)$ and $\tilde{t} = \max(k, n)$, there exist $M, \alpha, \beta > 0$ such that

\[
\left| \frac{\partial^{j+i} u(t, x)}{\partial t^j \partial x^i} \right| \leq M e^{\alpha t + \beta x}, \quad j = 0, 1, \ldots, k, i = 0, 1, \ldots, n.
\]

Then, the following formulas yield:
3. Modified Double Laplace Decomposition Method

In this section, the MDLDM will be constructed to solve the following initial-boundary value problem for one-dimensional time-dependent linear SPPPDE in the domain \([0, T] \times [0, L] \times [0, L],\)

\[
\begin{align*}
iu_t (t, x) &= au_{xxt} (t, x) + ux_t (t, x) + f (t, x), \\
(t, x) & \in (0, T) \times (0, L), \alpha > 0,
\end{align*}
\]  
(7)

with the following initial and boundary conditions:

\[
\begin{align*}
u (0, x) &= \psi (x), \quad x \in [0, L], \\
u (t, 0) &= u (t, L) = 0, \quad t \in [0, T].
\end{align*}
\]  
(8)

Firstly, the double Laplace transform with respect to \(t\) and \(x\) is applied to both sides of (7), and the following is obtained:

\[
L_t L_x [iu_t (t, x)] = L_t L_x [au_{xxt} (t, x) + ux_t (t, x) + f (t, x)].
\]  
(9)

In the next step, using the linearity and the differentiation properties of the double Laplace transform to equation (9), we obtain,

\[
L_t L_x [u (t, x)] = \frac{G (p)}{s} + \frac{1}{is} F (s, p) + \frac{1}{is} L_t L_x [au_{xxt} (t, x) + ux_t (t, x)],
\]  
(10)

where \(G (p)\) and \(F (s, p)\) symbolize the double Laplace transforms of \(\psi (x)\) and \(f (t, x)\), respectively.

The LADM is used to define the solution of one-dimen-
sional time-dependent linear SPPPDE \(u (t, x)\) by the infinite series as follows:

\[
u (t, x) = \sum_{n=0}^{\infty} u_n (t, x).
\]  
(11)

After this step, by applying the inverse double Laplace transform to both sides of equation (10) and using equation (11), as well as the linearity of the inverse transform, it holds,

\[
\begin{align*}
\sum_{n=0}^{\infty} u_n (t, x) &= \psi (x) - i L_t^{-1} L_x^{-1} \left[ \frac{F (s, p)}{s} \right] - i L_t^{-1} L_x^{-1} \left[ \frac{1}{s} L_t L_x \left[ \alpha \frac{\partial^3}{\partial t^3 \partial x^3} \sum_{n=0}^{\infty} u_n (t, x) + \frac{\partial^2}{\partial x^2} \sum_{n=0}^{\infty} u_n (t, x) \right] \right].
\end{align*}
\]  
(12)

From equation (12), the following recursive relationship can be acquired:

\[
\begin{align*}
u_0 (t, x) &= \psi (x) - i L_t^{-1} L_x^{-1} \left[ \frac{F (s, p)}{s} \right], \\
u_{n+1} (t, x) &= -i L_t^{-1} L_x^{-1} \left[ \frac{1}{s} L_t L_x \left[ \alpha \frac{\partial^3}{\partial t^3 \partial x^3} u_n (t, x) + \frac{\partial^2}{\partial x^2} u_n (t, x) \right] \right], \quad n \geq 0.
\end{align*}
\]  
(13)

(14)
Finally, the resulting components \( u_0, u_1, u_2, \ldots \) are substituted into the decomposition series for \( u(t, x) \), and then the required solution \( u(t, x) \) is obtained. Equations (13) and (14) are called the MDLDM of \( u(t, x) \).

3.1. Applications of the Method. We will now analyze some examples to demonstrate the applicability of the MDLDM for solving the one-dimensional time-dependent linear SPPDEs and SPPDEs.

**Example 1.** We consider the following initial-boundary value problem for a one-dimensional linear time-dependent SPPDE:

\[
\begin{align*}
\partial_t u(t, x) &= a\partial_{xx} u(t, x) + u_x(t, x) + f(t, x), \quad \alpha > 0, \\
f(t, x) &= [(i + a)\cos t + \sin t]\sin x, \quad 0 < t, x < \pi, \\
\partial_t u(0, x) &= 0, \quad 0 \leq x \leq \pi, \\
u(t, 0) &= u(t, \pi) = 0, \quad 0 \leq t \leq \pi.
\end{align*}
\]

When both sides of equation (16) are transformed with the inverse double Laplace transform, the result is as follows:

\[
u(t, x) = \left[(-a^2 + 1)\sin t - i(1 - \cos t)\sin x \right. - iL_t^{-1}L_x^{-1}\left[ \frac{1}{s}L_tL_x\left[ \alpha \frac{\partial^3}{\partial t^3} + \frac{\partial^2}{\partial x^2} \sum_{n=0}^{\infty} u_n(t, x) \right] + \frac{\partial^2}{\partial x^2} \sum_{n=0}^{\infty} u_n(t, x) \right].
\]

Putting the expansion (11) into equation (17), it leads to the following:

\[
\sum_{n=0}^{\infty} u_n(t, x) = [(a + 1)\sin t - i(1 - \cos t)]\sin x,
\]

Using the above mentioned recursive relation, the other components are given as follows:

\[
\begin{align*}
u_0(t, x) &= [(a + 1)\sin t - i(1 - \cos t)]\sin x, \\
u_1(t, x) &= [(a^2 - 3a^2i + x^2 i + 1)\sin t + (2a^2i + ai - 2\alpha - i)(1 - \cos t) + (3\alpha - 1)t + \frac{t^2}{2!}]\sin x, \\
u_2(t, x) &= \left[(-a^2 - 3ai + 2\alpha - 1)\sin t + (2a^2i + ai - 2\alpha - i)(1 - \cos t) + (3\alpha - 1)t + \frac{t^2}{2!} \right] \sin x, \\
u_3(t, x) &= \left[(a^2i - a^3 + 2\alpha^2 + 2\alpha + 1)\sin t + (5a^2 - 3ai - \alpha + 1)t - \frac{t^3}{3!} + (2a^3 - 3a^2i - 2\alpha^2 + 4\alpha + i)(1 - \cos t) - (4a + i^2) \frac{t^3}{2!}\right] \sin x.
\end{align*}
\]
The noise terms \( \pm ai \sin t \sin x \) and \( \pm i(1 - \cos t) \sin x \) between the components \( u_0(t, x) \) and \( u_1(t, x) \) can be taken away, and system (15) is still satisfied by the remaining term of \( u_0(t, x) \). As a result, the required solution is given by,

\[
u(t, x) = \sum_{n=0}^{\infty} u_n(t, x) = \sin t \sin x.
\]

(21)

For the SPPDE of system (15), that is, \( \alpha = 0 \), the same solution has been obtained.

**Example 2.** Consider the following initial-boundary value problem for a one-dimensional linear time-dependent SPPDE:

\[
\begin{aligned}
\left\{
\begin{array}{l}
u_t(t, x) = a \nu_{xx}(t, x) + \nu_{xx}(t, x) + f(t, x), \quad \alpha > 0, \\
f(t, x) = 2t(\alpha^2 - \alpha + 2t^2 - 1), \quad 0 < x, t < 1, \\
u(0, x) = x - x^2, \quad 0 \leq x \leq 1, \\
u(t, 0) = u(t, 1) = 0, \quad 0 \leq t \leq 1.
\end{array}
\right.
\end{aligned}
\]

(22)

Using the steps mentioned in the previous example, the following components are obtained:

\[
\nu_0(t, x) = (x^2 - x)(t^2 - 1) + \frac{2}{3} i t(t^2 + 3\alpha t - 3),
\]

\[
\nu_1(t, x) = -\frac{2}{3} i t(t^2 + 3\alpha t - 3),
\]

\[
\nu_2(t, x) = \nu_3(t, x) = \cdots = 0.
\]

Thus, the desired solution is as follows:

\[
u(t, x) = \sum_{n=0}^{\infty} u_n(t, x) = (x^2 - x)(t^2 - 1).
\]

(24)

For the SPPDE of system (22), we obtain the same solution.

### 4. Finite Difference Scheme and Its Stability

#### 4.1. Construction of the FDS

Let \( \Omega = (0, T) \times (0, L) \). For some nonnegative integers \( M \) and \( N \), the uniformly grid steps are introduced as,

\[
\Omega^t = \{t_k: t_k = k\tau, k = 1, \ldots, N - 1\}, \quad \tau = \frac{T}{N},
\]

\[
\Omega^h = \{x_n: x_n = nh, n = 1, \ldots, M - 1\}, \quad h = \frac{L}{M}.
\]

(25)

The values of the function \( \nu(t, x) \) at the grid points are represented as \( \nu(t_k, x_n) \equiv \nu_{nk} \).

In order to construct an explicit FDS, \( (u_k)_n \) is approximated by the forward Euler scheme and the two derivatives \( (u_{nk})_{n} \) and \( (u_{nk})_{n} \) by the backward Euler scheme. Because the solution \( \nu(t, x) \) is assumed to be smooth enough, the following three-layer explicit FDS at any point \( (t_k, x_n) \in \Omega \) can be written as follows:

\[
\begin{aligned}
\alpha \frac{u_{k+1}^n - u_k^n}{\tau} + \left( \frac{2}{h^2} - \frac{2\alpha}{h^2} \right) u_k^n + \frac{2\alpha}{h^2} i u_{k+1}^n + \frac{i}{\tau} u_{k-1}^n + \left( \frac{2\alpha}{h^2} - \frac{2\alpha}{h^2} \right) u_k^n + \frac{2\alpha}{h^2} i u_{k+1}^n + \frac{i}{\tau} u_{k-1}^n = f_{nk}^n,
\end{aligned}
\]

(26)

Ignoring discretization errors of order \( O(\tau^2, h^2) \), the explicit FDS for the SPPDE of problem (2) can be written as,
\[
\begin{align*}
\alpha & + 1 = a \\
\alpha & + 1 = b \\
\end{align*}
\]
So, explicit FDS (28) is stable at \( \| r_1 \| = 0 \), unconditionally unstable at \( \| r_2 \| > 1 \), and marginally stable at \( \| r_2 \| = 1 \).

5. Numerical Experiments

In this part, the explicit FDM is applied to solve the algebraic systems (15) and (22). MATLAB code is presented in the Appendix. To analyze the error between the obtained analytical \((u(t, x))\) and approximate \((u^N_k)\) solutions, we use,

\[
\varepsilon = \max_{k=0,1,\ldots,N} \max_{n=0,1,\ldots,M} |u(t, x) - u^N_k|.
\] (35)

We have shown that the analytic solution of system (15) is as follows:

\[
u(t, x) = \sin t \sin x,
\] (36)

and of system (22) is as follows:

\[
u(t, x) = (x^2 - x)(t^2 - 1).
\] (37)

We ran multiple simulations with different input parameters to compare the approximate and analytical solutions to quantitatively demonstrate the accuracy and efficiency of our schemes. The analytical solution was

\[
Table 1: Error analysis for system (15) with \( \tau = \pi / N \) and \( h = \pi / M \).
\[
\begin{array}{cccc}
N, M & \alpha & \varepsilon \\
\hline
40, 50 & 0 & 0.1365 \\
& 0.00001 & 0.1362 \\
& 0.001 & 0.1147 \\
65, 65 & 0 & 0.0857 \\
& 0.00001 & 0.0853 \\
& 0.001 & 0.0678 \\
70, 90 & 0 & 0.0795 \\
& 0.00001 & 0.0789 \\
& 0.001 & 0.0900 \\
150, 100 & 0 & 0.0386 \\
& 0.00001 & 0.0382 \\
& 0.001 & 0.0655 \\
\end{array}
\]

\[
Table 2: Error analysis for system (22) with \( \tau = 1 / N \) and \( h = 1 / M \).
\[
\begin{array}{cccc}
N, M & \alpha & \varepsilon \\
\hline
50, 50 & 0 & 0.0231 \\
& 0.00002 & 0.0231 \\
& 0.00005 & 0.0230 \\
70, 70 & 0 & 0.0173 \\
& 0.00002 & 0.0171 \\
& 0.00005 & 0.0169 \\
120, 108 & 0 & 0.0106 \\
& 0.00002 & 0.0103 \\
& 0.00005 & 0.0098 \\
250, 230 & 0 & 0.0053 \\
& 0.00002 & 0.0052 \\
& 0.00005 & 0.0063 \\
\end{array}
\]

Figure 1: Profile evolution solution for system (15) at \( N = 70 \) and \( M = 90 \). (a) Approximate simulation for \( \alpha = 0 \). (b) Approximate simulation for \( \alpha = 0.001 \). (c) Analytic solution.
approximated using the MDLDM, while the approximate solution was approximated using the explicit FDS. The error analysis (ε) between the two solutions for systems (15) and (22) is shown in Tables 1 and 2, respectively.

Figures 1 and 2 show a comparison of the profile evolution of the analytical and approximate solutions for systems (15) and (22), respectively.

Since the exact solutions obtained for the mentioned systems did not contain the parameter (α), then the graph of the analytic solution does not change as the value of (α) changes.

6. Conclusion

In this study, the initial-boundary value problems for one-dimensional linear time-dependent SPPPDE and SPPDE are discussed. The semianalytic and approximate solutions for the SPPPDE and SPPDE of problem (2) were obtained using MDLDM and explicit FDS, respectively. The von Neumann stability analysis of the submitted problems was investigated. The abovementioned tables and figures indicate that the values of (α) have an impact on the solutions.

Appendix

MATLAB code

```matlab
1 clc; clear all; close all;
2 % 1D time-dependent linear Schrodinger pseudo-parabolic PDE in the domain [0, pi] x [0, pi]
3 % iut = Alpha * uxxx + uxx + f(t, x), 0t, x < pi
4 % f(t, x) = [(i + Alpha) * cost + sint]sinx
5 % Initial condition u(0, x) = 0, 0 ≤ x ≤ pi
6 % Boundary conditions u(t, 0) = u(t, pi) = 0, 0 ≤ t ≤ pi
7 % Approximate solution by using explicit FDS
8 N = 70;
9 M = 90;
10 Alpha = 0.001;
11 h = pi/M;
12 tau = pi/N;
13 a = (Alpha/(tau * (h2)));
14 r = (-Alpha/(tau * (h2))-(1/(h2)));
15 b = 0;
16 c = (-complex (0, 1)/tau + 2/(h2)−2 * Alpha/(tau * (h2))); % complex (0, 1) represents the imaginary number i
17 d = (complex (0, 1)/tau + 2 * Alpha/(tau * (h2)));
18 for i = 2: N + 1;
19 A (i, i−1) = a;
20 A (i, i) = r;
21 end; A;
22 C = A;
23 for i = 2: N + 1;
24 K (i, i−1) = b;
25 end;
26 for i = 2: N + 1;
27 K (i, i) = c;
28 end;
29 for i = 2: N;
30 K (i, i + 1) = d;
31 end;
32 K (1, 1) = 1;
33 for i = 1: N + 1;
34 D (i, i) = 1;
35 end;
36 D;
37 % Finding fii (j);
38 for j = 1: M + 1;
39 x = (j−1) * h;
40 fii (1, j) = 0; % Initial condition of the given problem
41 for k = 2: N + 1;
42 t = tau * (k);
43 fii (k, j) = ((Alpha + complex (0, 1) * cos(t) + sint) * sin(x)); % The given function f(t, x)
44 end;
45 end;
46 alpha (N + 1, N + 1, 1:1) = 0;
47 beta (N + 1, 1:1) = 0;
```

Figure 2: Profile evolution solution for system (22) at N = 120 and M = 108. (a) Approximate simulation for α = 0. (b) Approximate simulation for α = 0.00005. (c) Analytic solution.
Advances in Mathematical Physics

84 set (gca, ‘ZLim’, [q w]); rotate3d; xlabel (‘x’); ylabel
83 figure; surf (Xs, Ts, abs (app)); title (‘Approximate
80 q
79 w
78 table (1 : 2: end,:)
77 table (1 : 2: end,:)
76 table [ex; app];
75 ex;
74 % Plotting exact and approximate solutions
73 Answer = [maxes, maxapp, maxerror, relativeerror]
72 relativeerror = maxerror/maxapp;
71 maxerror = max (max (abs (ex–app)));
70 maxapp = max (max (abs (app)));
69 maxes = max (max (abs (ex)));
68 % Absolute error analysis;
67 ex;
66 end;
65 ex;
64 (k, j: j) = sin(t) * sin(x);
63 t = tau * (k–1);
62 x = (j–1) * h;
61 for k = 1: N + 1;
60 for j = 1: M + 1;
59 % Analytic solution;
58 end;
57 [Xs, Ts] = meshgrid (0: h: pi, 0: tau: pi);
56 for z = 1: M;
55 end;
54 U (:, z+1) = alpha (:, z+1: z+1) * U (:, z+1: z+1) + beta (:, z+1: z+1);
53 for z = M–1:–1:1;
51 end;
50 beta (:, j+1: j+1) = inv (K + C * alpha(:,j:j)) * (-A);
49 alpha (:,j+1: j+1) = inv (K + C * alpha(:,j:j)) * (D * fi((j,j))-C * beta(:,j:j));
48 for j = 1: M–1;
47 % Absolute error analysis;
46 end;
45 end;
44 end;
43 end;
42 for j = 1: M+1;
41 for k = 1: N+1;
40 x = (j–1) * h;
39 t = tau * (k–1);
38 ex = sin(t) * sin(x);
37 end;
36 end;
35 end;
34 figure; surf (Xs, Ts, abs (app)); title (‘Approximate
33 figure; surf (Xs, Ts, abs (app)); title (‘Approximate
32 set (gca, ‘ZLim’, [q w]); rotate3d; xlabel (‘x’); ylabel (‘t’);
31 set (gca, ‘ZLim’, [q w]); rotate3d; xlabel (‘x’); ylabel (‘t’);
30 w = max (max (table));
29 q = min (min (table));
28 table = [ex; app];
27 table (1 : 2: end,:) = ex;
26 table (1 : 2: end,:) = app;
25 for j = 1: M+1;
24 for k = 1: N+1;
23 x = (j–1) * h;
22 t = tau * (k–1);
21 ex = sin(t) * sin(x);
20 end;
19 end;
18 end;
17 end;
16 ex;
15 % Absolute error analysis;
14 end;
13 end;
12 for j = 1: M+1;
11 for k = 1: N+1;
10 x = (j–1) * h;
9 t = tau * (k–1);
8 ex = sin(t) * sin(x);
7 end;
6 end;
5 end;
4 end;
3 for j = 1: M+1;
2 for k = 1: N+1;
1 x = (j–1) * h;
0 t = tau * (k–1);
–8 maxes = max (max (abs (ex)));
–7 maxapp = max (max (abs (app)));
–6 maxerror = max (max (abs (ex–app)));
–5 relativeerror = maxerror/maxapp;
–4 Answer = [maxes, maxapp, maxerror, relativeerror]
–3 % Plotting exact and approximate solutions
–2 separately;
–1 [Xs, Ts] = meshgrid (0: h: pi, 0: tau: pi);
0 table = [ex; app];
–9 table (1 : 2: end,:) = ex;
–8 table (1 : 2: end,:) = app;
–7 q = min (min (table));
–6 w = max (max (table));
–5 figure; surf (Xs, Ts, abs (ex)); title (‘Exact Solution’);
–4 set (gca, ‘ZLim’, [q w]); rotate3d; xlabel (‘x’); ylabel (‘t’);
–3 figure; surf (Xs, Ts, abs (app)); title (‘Approximate
–2 set (gca, ‘ZLim’, [q w]); rotate3d; xlabel (‘x’); ylabel (‘t’);
–1

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References


