



A Comparative Study of Hybrid Analytical and Laplace Transform Approaches for Solving Partial Differential Equations in Python

P. Jalili^a, M. Mahboob^b, A. Shateri^a, B. Jalili^a, D. Domiri Ganji^{*b}

^a Department of Mechanical Engineering, North Tehran Branch, Islamic Azad University, Tehran, Iran

^b Department of Mechanical Engineering, Babol Noshirvani University of Technology, Babol, Iran

PAPER INFO

Paper history:

Received 25 August 2023

Received in revised form 25 September 2023

Accepted 26 September 2023

Keywords:

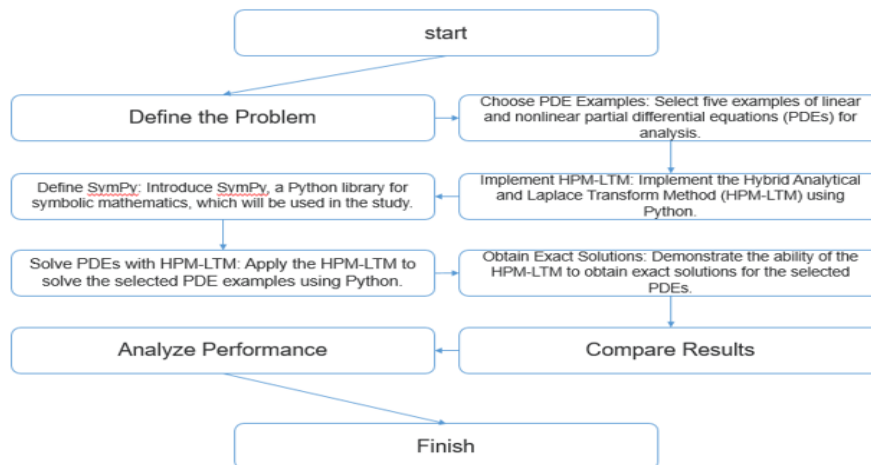
Partial Differential Equations
Homotopy Perturbation Method
Laplace Transform Method
Python Programming

ABSTRACT

This research presents a rigorous and innovative approach, the Homotopy Perturbation Method-Laplace Transform Method (HPM-LTM), implemented in Python, for the efficient solution of linear and nonlinear partial differential equations (PDEs). By combining the Homotopy Perturbation technique (HPM) with the Laplace Transform Method (LTM), our method successfully addresses the significant challenges posed by equations with nonlinear components. Through the utilization of He's polynomials, the HPM-LTM approach effectively handles nonlinear terms, resulting in accurate and reliable solutions. To demonstrate the efficacy of our method, we extensively apply it to five representative PDE scenarios, including heat and wave equations. Our comprehensive results substantiate the remarkable accuracy and reliability of the HPM-LTM approach, highlighting its superiority in comparison to conventional approaches that require restrictive assumptions or discretization, which can introduce round-off errors. Furthermore, our method overcomes the limitations imposed by numerical errors inherent in traditional HPM techniques. The robustness, effectiveness, and adaptability of our proposed approach are further validated by its successful application to a wide range of PDE problems across various fields. This research presents a significant contribution to the development of a powerful computational tool for resolving diverse PDE problems, with particular relevance to the engineering discipline.

doi: 10.5829/ije.2024.37.02b.11

Graphical Abstract



*Corresponding Author Email: mirgang@nit.ac.ir (D. Domiri Ganji)

Please cite this article as: Jalili P, Mahboob M, Shateri A, Jalili B, Domiri Ganji D. A Comparative Study of Hybrid Analytical and Laplace Transform Approaches for Solving Partial Differential Equations in Python, International Journal of Engineering, Transactions B: Applications. 2024;37(02):352-64.

1. INTRODUCTION

Partial differential equations (PDEs) are fundamental for modeling various physical phenomena, such as fluid dynamics, heat transfer, and electromagnetic fields. Analytic solutions for PDEs are usually not available, leading to the need for numerical methods to obtain approximate solutions. The Homotopy Perturbation Method (HPM) has recently become an efficient and reliable method for solving linear and nonlinear PDEs (1-3). However, the HPM can suffer from round-off errors in some cases. Therefore, the focus of this study is to propose a hybrid approach that combines the HPM with the Laplace Transform Method (LTM) to improve the accuracy of the numerical solutions. Various methods have been developed for solving PDEs, such as Finite Difference Method (FDM), Finite Element Method (FEM), Differential Transform Method (4) and Spectral Method. However, these methods often require intricate discretization techniques that can lead to inaccurate solutions (5). The HPM, developed by He, offers a promising alternative approach that avoids these issues by finding a solution in the form of a series of homotopy functions that the Taylor series can rapidly and effortlessly approximate. The HPM has recently been used to solve several PDEs (6-8).

Although highly effective, the HPM has some limitations, particularly in dealing with nonlinear terms. To address this issue, a new method called the He-Laplace method has been introduced, which combines the HPM with He's polynomials to handle nonlinearities in PDEs (8, 9). The proposed method has been applied to linear and nonlinear PDEs, exhibiting superior accuracy and efficiency to traditional numerical methods (10, 11). Researchers have favored the homotopy perturbation method (HPM) as an analytical technique to solve a range of Heat transport and fluid mechanics differential equations in recent studies (12-17). In a recent paper by Jalili et al. (18), the HPM and AGM method was used to resolve thermal problems for micro-polar nanofluids. Alternatively, Jalili et al. (19) applied the HPM method to tackle problems related to the movement of a plate submerged in a Newtonian fluid examined through Caputo fractional differential equations. Jalili et al. (20) also demonstrated a novel fractional analytical approach to solve time-space fractional equations from oil pollution analysis using HPM.

Python programming has become a common tool in scientific computing due to its simplicity, flexibility, and efficient numerical package libraries, such as NumPy, SciPy, and Matplotlib. The availability of open-source libraries for Python has enabled the development of fast and efficient algorithms for solving PDEs using numerical methods (21-23). The HPM has recently been used with other methods to improve its accuracy and efficiency. The HPM-Laplace method, which combines

the HPM with the Laplace Transform Method (LTM) to increase the accuracy of the numerical solutions, has been proposed. The method has been used to solve various linear and nonlinear PDEs, exhibiting superior accuracy and efficiency compared to traditional numerical methods (24, 25). The HPM-Laplace method could be implemented in Python to solve PDE problems with high accuracy and efficiency. Another hybrid approach involving the HPM is He's polynomial-based decomposition method, which uses He's polynomials to decompose the solution of a PDE into several parts, for which each part can be solved by a different method (26). The decomposition method involves the solution of smaller sub-problems, which can easily be implemented using Python functions. Also, much research has been done on the analytical and Python approach (27-34).

In regards to the applications of solving PDEs, the following articles shed light on diverse areas of research, including the dynamics of modified Peyrard-Bishop DNA models in bio-fluids, the analysis of protein oscillations through Langevin and Fokker-Planck equations, and the optimization of heat exchanger design for enhanced heat transfer rates and temperature distribution. Lap-Arparat and Tuchinda (35) explored the behavior of the modified Peyrard-Bishop DNA model within a thermostat as a bio-fluid. They investigate the soliton-like solutions in this system and discuss their implications for understanding the dynamics of DNA molecules. Baughman and Sharma (36) focused on the entropic analysis of protein oscillations. They employed Langevin and Fokker-Planck equations to study the dynamics of proteins and analyzed the role of entropy in protein oscillations. Their findings contribute to a deeper understanding of the complex behavior of proteins. In the study conducted by Sutantyo et al. (37), they have investigated the impact of heat exchanger design on heat transfer rate and temperature distribution. They analyzed different heat exchanger designs and evaluate their heat transfer efficiency and temperature distribution performance. The results of their study provided valuable insights for optimizing heat exchanger design to enhance heat transfer processes.

In the realm of solving partial differential equations (PDEs), significant difficulties and challenges hinder the attainment of accurate and reliable solutions. One of the primary obstacles lies in effectively handling equations with nonlinear components, which often lead to complex mathematical formulations. To overcome this challenge, we have developed a unique hybrid technique, HPM-LTM, which combines the advantages of the Homotopy Perturbation technique (HPM) and the Laplace Transform Method (LTM). By employing He's polynomials, our approach successfully tackles nonlinear terms, enabling the efficient and accurate solution of PDEs. This original achievement allows us to bypass the limitations imposed by conventional approaches that are

often plagued by numerical errors and restrictive assumptions. By implementing the HPM-LTM method, we have achieved remarkable accuracy and reliability in solving PDEs without discretization, thereby eliminating potential round-off errors. We develop Python functions to implement the proposed method and use the numerical package libraries to solve five PDE examples in various engineering fields. The examples have been chosen to demonstrate the applicability and flexibility of our method in various domains. Specifically, the examples include linear PDEs, such as the heat and wave equation, and nonlinear PDEs, such as the Burgers' Equation, the MHD flow equation, and the Klein-Gordon equation. By including a diverse set of examples, ranging from linear to nonlinear PDEs, we aim to showcase the effectiveness of our method across various engineering domains. Adopting this approach allows us to assess the method's performance under different types of PDEs, capturing a wider scope of real-world scenarios.

The article is structured as follows: Section 1 provides an introduction, offering an overview of the problem at hand and a review of the relevant literature. Section 2 focuses on the mathematical formulation, explaining the fundamental concepts of the HPM and LTM methods. Section 3, titled methodology, delves into the basic idea of the HPM-LTM method and provides a detailed explanation of SymPy and differential equations. Section 4 presents five examples and their solutions, showcasing the results obtained for both linear and nonlinear partial differential equations. Finally, section 5 concludes the article, summarizing the findings and highlighting the key advantages of the HPM-LTM method.

2. MATHEMATICAL FORMULATION

Numerous scientific investigations rely heavily on mathematical formulations because they give researchers a mechanism to properly define and measure the physical processes under investigation. We shall examine how our current problem is mathematically formulated in this part. We shall start by describing the issue statement and stating our goals. The key mathematical theories and equations required to resolve the issue will then be covered. We aim to give a clear and concise grasp of the underlying ideas and techniques required to solve by decomposing the issue into its constituent mathematical parts.

2.1. The Basic Idea of HPM We examined the subsequent equation to elucidate the underlying concepts:

$$w(k) - h(j) = 0, \quad j \in \Gamma \tag{1}$$

With the B.Cs. of:

$$B \left(k, \frac{\partial k}{\partial n} \right) = 0, \quad j \in \delta \tag{2}$$

where B is a boundary operator, δ is a domain (Γ) boundary, $h(j)$ is a known analytic function, and $w(k)$ is a differential operator. Therefore, Equation 1 can be rewritten as Equation 3, $L(k)$ is linear, and $N(k)$ is nonlinear.

$$L(k) + N(k) - h(j) = 0, \quad j \in \Gamma \tag{3}$$

The HPM configuration looks like Equation 4:

$$H(q, p) = (1 - p)[L(q) - L(k_0)] + p[w(q) - h(j)] = 0 \tag{4}$$

$$q(j, p): \Gamma \times [0, 1] \rightarrow R \tag{5}$$

The initial approximation in Equation 5 that satisfies the B.Cs. is k_0 , and p is an embedded parameter that falls between $[0, 1]$. The answer to Equation 4 can be expressed as follows in powers of p :

$$q = q_0 + pq_1 + p^2q_2 + \dots \tag{6}$$

And the best guess for the answer is:

$$k = \lim_{p \rightarrow 1} q = q_0 + q_1 + q_2 + \dots \tag{7}$$

2.2. The Basic Idea of LTM Operators, also known as transformations, turn one function into another. Although transformations alter the original function, they can maintain or change the independent variable. For example, when a function $f(t)$ is multiplied by a number n using the operator O , the resulting function $g(t)$ keeps the original function's independent variable. Formally, such an operator may be stated as:

$$O[f(t)] = n \cdot f(t) = g(t) \tag{8}$$

The inverse Laplace transform acts on $F(s)$ to get $f(t)$, whereas the direct Laplace transform takes a function $f(t)$ that relies on time t and converts it into a different function $F(s)$ that depends on the complex variable s . The Laplace transform is described mathematically as

$$F(s) = \mathcal{L}[f(t)] = \int_0^\infty f(t) \cdot e^{-s \cdot t} dt \tag{9}$$

For the transformation to occur, the integral defining the Laplace transform must converge, which necessitates that the initial function $f(t)$ meet the following prerequisites:

- 1) If the interval can be divided into a finite number of nonintersecting intervals, then $f(t)$ must be piecewise continuous across the interval $0 < t < \infty$, which means it must be continuous on any subinterval of that interval, and it must have finite bounds at the endpoints of each subinterval.
- 2) Given that $f(t)$ must be of exponential order, it must meet the equality for real constant values σ

that is greater than a critical value, σ_c , often known as the abscissa of convergence.

Formally, the inverse Laplace transform, which is often needed to return the original (generally unknown) time-domain function $f(t)$, is calculated as:

$$f(t) = \mathcal{L}^{-1}[F(s)] \tag{10}$$

3. METHODOLOGY

3.1. The Basic Idea of HPM-LTM Up to step (4), HPM-LTM follows the same stages as HPM. Then, we apply the Laplace transform to both sides of the homotopy Equation 4:

$$\mathfrak{S}\{L(K) - L(k_0) + p[L(k_0) + N(K) - h(j)]\} = 0 \tag{11}$$

The possibility exists that HPM-LTM could replace (k_0) , with any arbitrary function $z(x)$, provided that this function is defined by certain unspecified parameters A, B, C, etc. These parameters would need to be

$$\sum_{n=0}^{\infty} p^n q_n = \mathfrak{S}^{-1} \left\{ \left(\frac{1}{s^n} \right) \{s^{n-1}K(0) + s^{n-2}K'(0) + \dots + K^{(n-1)}(0)\} + \left(\frac{1}{s^n} \right) \mathfrak{S}\{L(k_0) - pL(k_0) + p[-N(\sum_{n=0}^{\infty} p^n q_n) + h(j)]\} \right\} \tag{16}$$

Comparing p coefficients of equal power results in

$$\begin{aligned} p^0: q_0 &= \mathfrak{S}^{-1} \left\{ \left(\frac{1}{s^n} \right) (s^{n-1}K(0) + s^{n-2}K'(0) + \dots + K^{(n-1)}(0)) + \mathfrak{S}\{L(k_0)\} \right\}, \\ p^1: q_1 &= \mathfrak{S}^{-1} \left\{ \left(\frac{1}{s^n} \right) (\mathfrak{S}\{-N_0(q_0) - L(k_0) + h(rj)\}) \right\}, \\ p^2: q_2 &= \mathfrak{S}^{-1} \left\{ \left(\frac{1}{s^n} \right) \mathfrak{S}\{-N_1(q_0, q_1)\} \right\} \\ p^3: q_3 &= \mathfrak{S}^{-1} \left\{ \left(\frac{1}{s^n} \right) \mathfrak{S}\{-N_2(q_0, q_1, q_2)\} \right\}, \\ p^j: q_j &= \mathfrak{S}^{-1} \left\{ \left(\frac{1}{s^n} \right) \mathfrak{S}\{-N_{j-1}(q_0, q_1, q_2, \dots, q_{j-1})\} \right\} \end{aligned} \tag{17}$$

Assuming that $K(0) = k_0 = \alpha_0, K'(0) = \alpha_1, \dots, K^{n-1}(0) = \alpha_{n-1}$; the exact solution may be obtained as follows:

$$k = \lim_{p \rightarrow 1} K = q_0 + q_1 + q_2 + \dots \tag{18}$$

The notation presented above indicates the potential for extending the nonlinear operator N as a formal series with respect to the embedding parameter p .

3.2. SymPy and Differential Equations In our study, we harnessed the capabilities of Python in conjunction with the SymPy module, a powerful tool for conducting symbolic computations. Particularly in algebra, discrete mathematics, and calculus, we found SymPy exceptionally useful, primarily due to its proficiency in formatting and presenting outcomes in LaTeX format. We employed a multi-step procedure to address differential equations using SymPy.

appropriately determined using the flexibility afforded by the homotopy formulation. Then, we have:

$$s^n \mathfrak{S}\{K\} - s^{n-1}K(0) - s^{n-2}K'(0) - \dots - K^{(n-1)}(0) = \mathfrak{S}\{L(k_0) - pL(k_0) + p[-N(K) + h(j)]\} \tag{12}$$

Or

$$\mathfrak{S}(K) = \left(\frac{1}{s^n} \right) \{s^{n-1}K(0) + s^{n-2}K'(0) + \dots + K^{(n-1)}(0)\} + \left(\frac{1}{s^n} \right) \mathfrak{S}\{L(k_0) - pL(k_0) + p[-N(K) + h(j)]\}, \tag{13}$$

When we apply the inverse Laplace transform to Equation 13, we get:

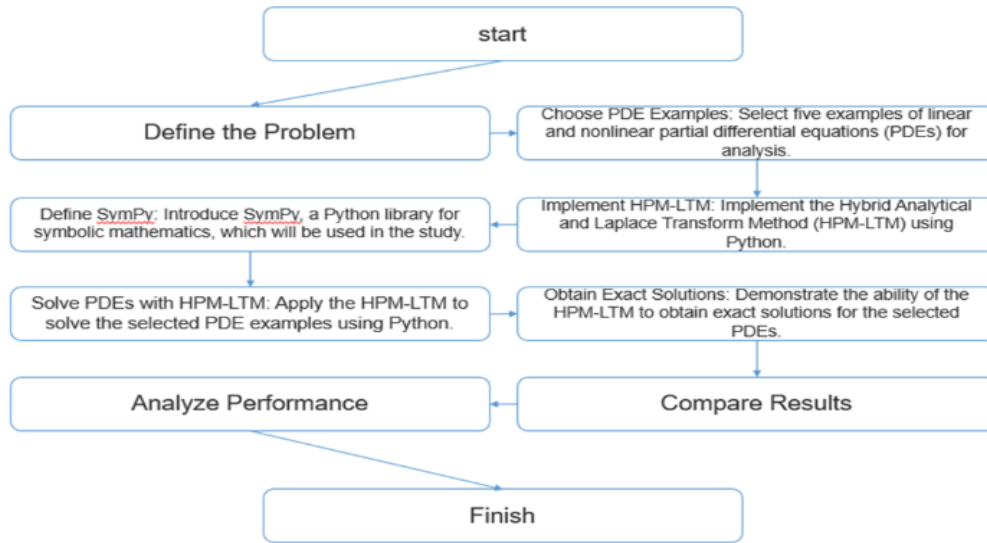
$$K = \mathfrak{S}^{-1} \left\{ \left(\frac{1}{s^n} \right) \{s^{n-1}K(0) + s^{n-2}K'(0) + \dots + K^{(n-1)}(0)\} + \left(\frac{1}{s^n} \right) \mathfrak{S}\{L(k_0) - pL(k_0) + p[-N(K) + h(j)]\} \right\}. \tag{14}$$

Assuming that a power series of p may be used to describe the answers to Equation 3:

$$K = \sum_{n=0}^{\infty} p^n q_n \tag{15}$$

then we obtain by replacing Equation 15 with Equation 14.

Commencing with importing the SymPy library and associated symbols, we defined the dependent variable as a function of the independent counterpart. The incorporation of boundary or initial conditions was achieved by utilizing the '.subs' method. We then embarked on the creation of the differential equation itself. Subsequently, the crux of our endeavor lay in solving this intricate differential equation by applying the 'dsolve' function. To streamline our efforts, we availed ourselves of the 'from sympy import *' and 'from sympy.abc import *' directives facilitated the importation of requisite modules and symbols. The complex differential equations central to our research were successfully unraveled, all thanks to the robust support provided by SymPy. This, in turn, furnished us with profound insights that greatly enriched our understanding. To clarify the methodology, we provide the following flowchart:



4. ILLUSTRATIVE EXAMPLES

We have applied the HPM-LTM method to demonstrate its effectiveness in two linear and three nonlinear partial differential equations.

Example 4.1. The homogeneous linear PDE (26, 36, 38) is as follows:

$$\frac{\partial y}{\partial t} + \frac{\partial y}{\partial x} - \frac{\partial^2 y}{\partial x^2} = 0 \tag{19}$$

with the following B.Cs.:

$$y(x, 0) = e^x - x, \quad y(0, t) = 1 + t, \quad \frac{\partial y}{\partial x}(1, t) = e - 1 \tag{20}$$

Applying the procedure above under the presumption that existed initially, we have:

$$y(x, s) = \frac{e^x - x}{s} - \frac{1}{s} L \left[\frac{\partial y}{\partial x} - \frac{\partial^2 y}{\partial x^2} \right] \tag{21}$$

The Laplace transform's inverse suggests that:

$$y(x, t) = e^x - x - L^{-1} \left[\frac{1}{s} L \left[\frac{\partial y}{\partial x} - \frac{\partial^2 y}{\partial x^2} \right] \right] \tag{22}$$

Now, we apply the HPM; we have

$$\sum_{n=0}^{\infty} p^n y_n(x, t) = e^x - x - p \left(L^{-1} \left[\frac{1}{s} L \left[\frac{\partial y}{\partial x} - \frac{\partial^2 y}{\partial x^2} \right] \right] \right) \tag{23}$$

The coefficient of similar powers of p is compared, and we have:

$$\begin{aligned} p^0: y_0(x, t) &= e^x - x \\ p^1: y_1(x, t) &= -L^{-1} \left[\frac{1}{s} L \left[\frac{\partial y_0}{\partial x} - \frac{\partial^2 y_0}{\partial x^2} \right] \right] = t \\ p^2: y_2(x, t) &= -L^{-1} \left[\frac{1}{s} L \left[\frac{\partial y_1}{\partial x} - \frac{\partial^2 y_1}{\partial x^2} \right] \right] = 0 \end{aligned} \tag{24}$$

$$\begin{aligned} p^3: y_3(x, t) &= 0, \\ p^4: y_4(x, t) &= 0, \end{aligned} \tag{25}$$

so the solution $y(x, t)$ is given by

$$\begin{aligned} y(x, t) &= e^x - x + t + 0 + 0 \dots \\ &= e^x - x + t \end{aligned} \tag{26}$$

We begin by noting that Equation 26 represents the exact solution of the problem, as reported in literature (26, 36, 38). Different approximate solutions are obtained in some experiments using HPM (36, 38). By applying HPM-LTM, we obtain exact solutions.

The problem was solved using Python, leveraging the capabilities of the symbolic mathematics library Sympy. The results obtained from the solution are presented in Figures 1 and 2. Figure 1 depicts the exact solution of $y(t) = e^x - x + t$ for various values of t . It is evident that as the value of x increases, the solution of $y(t)$ exhibits exponential growth. The different lines in the figure correspond to different values of t in the exact solution of the equation. This visualization provides valuable insights into the behavior of the solution as it varies with different input parameters. Figure 2 presents a 2D contour plot that offers a clearer visualization of the exponential increase in t as x increases. The contour lines represent equal $y(t)$ values for different combinations of x and t . By examining this contour plot, we understand how the solution behaves across the entire parameter space. The contour lines help identify regions where the solution exhibits significant changes or remains relatively constant. Both Figures 1 and 2 collectively demonstrate the exponential behavior of the exact solution of the given equation. These visual representations provide valuable insights and enhance our understanding of the problem. Employing the

Matplotlib library in Python facilitated the creation of these plots, enabling the effective communication of the obtained results.

The results obtained through the utilization of Python, Sympy, and Matplotlib highlight the exponential growth of the exact solution of $y(t) = e^x - x + t$. The figures presented in this section provide a comprehensive visualization of this behavior, showcasing the impact of varying input parameters on the solution. These findings contribute to a deeper understanding of the problem and validate the effectiveness of this study's hybrid analytical and Laplace transform methods.

Example 4.2. Take into account the homogeneous linear PDE (Klein-Gordon equation) shown below (26, 36, 38):

$$\frac{\partial^2 y}{\partial t^2} + y - \frac{\partial^2 y}{\partial x^2} = 0 \tag{27}$$

with the following B.Cs.:

$$y(x, 0) = e^{-x} + x, \quad \frac{\partial y}{\partial t}(x, 0) = 0 \tag{28}$$

Applying the procedure above by the I.Cs., we have:

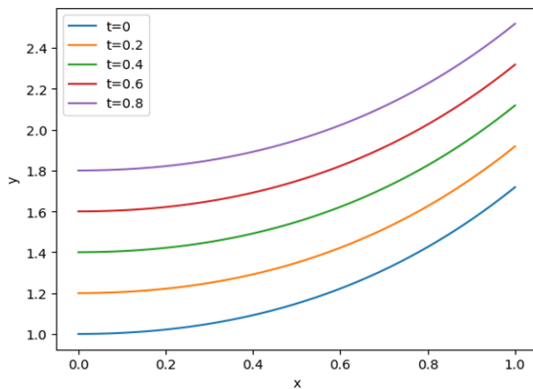


Figure 1. Exact solution of example 4.1 using HPM-LTM method for varying values of t, plotted in Python using Sympy and Matplotlib for different values of t

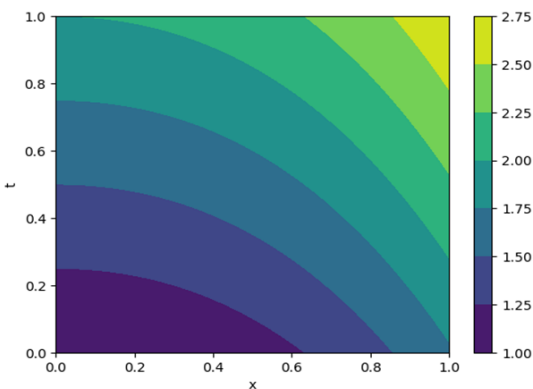


Figure 2. 2D contour plot of the exact solution of example 4.1 using the HPM-LTM method, plotted in Python using Sympy and Matplotlib

$$y(x, s) = \frac{e^{-x+x}}{s} - \frac{1}{s^2} L \left[y - \frac{\partial^2 y}{\partial x^2} \right] \tag{29}$$

The Laplace transform's inverse suggests that:

$$y(x, t) = e^{-x} + x - L^{-1} \left[\frac{1}{s^2} L \left[y - \frac{\partial^2 y}{\partial x^2} \right] \right] \tag{30}$$

Now, we apply the HPM; we have

$$\sum_{n=0}^{\infty} p^n y_n(x, t) = e^{-x} + x - p \left(L^{-1} \left[\frac{1}{s^2} L \left[y - \frac{\partial^2 y}{\partial x^2} \right] \right] \right). \tag{31}$$

The coefficient of similar powers of p is compared, and we have:

$$p^0: y_0(x, t) = e^{-x} + x$$

$$p^1: y_1(x, t) = -L^{-1} \left[\frac{1}{s^2} L \left[y_0 - \frac{\partial^2 y_0}{\partial x^2} \right] \right] = \frac{-xt^2}{2!}, \tag{32}$$

$$p^2: y_2(x, t) = -L^{-1} \left[\frac{1}{s^2} L \left[y_1 - \frac{\partial^2 y_1}{\partial x^2} \right] \right] = \frac{xt^4}{4!}.$$

$$p^3: y_3(x, t) = \frac{-xt^6}{6!}$$

$$p^4: y_4(x, t) = \frac{xt^8}{8!} \tag{33}$$

$$p^n: y_n(x, t) = \frac{(-1)^n xt^{2n}}{2n!}$$

so that the solution $y(x, t)$ is given by:

$$y(x, t) = y_0 + y_1 + y_2 + y_3 + \dots$$

$$= e^{-x} + x - \frac{xt^2}{2!} + \frac{xt^4}{4!} - \frac{xt^6}{6!} + \dots + \frac{(-1)^n xt^{2n}}{2n!}$$

$$= e^{-x} + x \left(1 - \frac{t^2}{2!} + \frac{t^4}{4!} - \dots + \frac{(-1)^n t^{2n}}{2n!} + \dots \right) \tag{34}$$

$$= e^{-x} + x \cos(t),$$

Equation 34 is the exact solution to the problem (26, 36, 38), which is a homogeneous linear PDE (Klein-Gordon equation), depicted in Figures 3 and 4, using the HPM-LTM method in Python (26, 36, 38). Different approximate solutions are obtained in some experiments using HPM (36, 38). By applying HPM-LTM, we obtain exact solutions.

Figure 3 specifically focuses on the behavior of the solution over time or for different boundary conditions. It presents a line graph illustrating the exact solution of the equation for different values of t. The graph clearly demonstrates how the solution evolves over time, providing insights into its temporal dynamics. As t increases, the corresponding value of y decreases. This decreasing trend suggests that the amplitude of the solution diminishes as time progresses. Notably, a particularly noticeable drop in the solution occurs at t = 0.8. This observation indicates that the solution becomes less pronounced as time advances. By visualizing the solution's behavior for various values of t, Figure 3 effectively showcases the temporal evolution and sensitivity of the solution to different time-dependent factors. This plot provides valuable insights into how the solution changes over time and in response to varying boundary conditions. Figure 4, on the other hand, offers a 2D contour plot that represents different solution levels

based on y and x values. This contour plot allows us to identify variations in the solution across the $y - x$ parameter space, enabling the identification of patterns, trends, and critical points. We can observe regions of equal solution values by examining the contour lines in Figure 4. This visualization helps us identify areas where the solution exhibits significant changes or remains relatively constant. Consequently, we can pinpoint critical points, such as regions with steep gradients or areas where the solution transitions rapidly. The 2D contour plot presented in Figure 4 enhances our understanding of the solution by providing a comprehensive overview of its variations across different y and x values.

Figures 3 and 4 provide a detailed analysis of the exact solution of the homogeneous linear PDE obtained using the HPM-LTM method in Python. Figure 3's line graph demonstrates the solution's temporal behavior, showcasing a decreasing trend in the solution as t increases. On the other hand, Figure 4's 2D contour plot offers a comprehensive visualization of the solution's variations across the $y - x$ parameter space, allowing us to identify patterns, trends, and critical points. Both figures contribute to a deeper understanding of the solution and further validate the effectiveness of the HPM-LTM method for solving the considered homogeneous linear PDE.

Example 4.3. Take into account the homogeneous nonlinear PDE (Burger equation) shown below (26, 36, 38):

$$\begin{aligned}
 p^0: y_0(x, t) &= 1 - x \\
 p^1: y_1(x, t) &= L^{-1} \left[\frac{1}{s} \left\{ L \left[\frac{\partial^2 y_0}{\partial x^2} \right] + L[H_0(y)] \right\} \right] = -(1 - x)t \\
 p^2: y_2(x, t) &= L^{-1} \left[\frac{1}{s} \left\{ L \left[\frac{\partial^2 y_1}{\partial x^2} \right] + L[H_1(y)] \right\} \right] = (1 - x)t^2 \\
 p^3: y_3(x, t) &= -(1 - x)t^3 \\
 p^4: y_4(x, t) &= (1 - x)t^4
 \end{aligned}$$

so that the solution $y(x, t)$ is given by:

$$\begin{aligned}
 y(x, t) &= y_0 + y_1 + y_2 + y_3 + \dots \\
 &= (1 - x) - (1 - x)t + (1 - x)t^2 - (1 - x)t^3 + \dots \\
 &= (1 - x)[1 - t + t^2 - t^3 + t^4 - \dots] \\
 &= (1 - x)(1 + t)^{-1} = \frac{(1-x)}{(1+t)},
 \end{aligned}$$

which is the exact solution to the problem (26, 36, 38). In some experiments, different approximate solutions are obtained by using HPM (36, 38). By applying HPM-LTM, we obtain exact solutions. Figure 5 presents a line graph that showcases the behavior of the solution for various values of t (0, 0.2, 0.4, 0.6, 0.8) in relation to the value of y . The graph provides valuable insights into the dynamics of the solution and its sensitivity to changes in t .

The observed trend in Figure 5 indicates that the solution curve is decreasing, with y decreasing as t increases. This behavior implies that the amplitude of the solution

$$\frac{\partial y}{\partial t} - y \frac{\partial y}{\partial x} - \frac{\partial^2 y}{\partial x^2} = 0 \tag{35}$$

with the following B.Cs.:

$$y(x, 0) = 1 - x, y(0, t) = \frac{1}{(1+t)}, y(1, t) = 0 \tag{36}$$

Applying the procedure above while keeping in mind the starting state, we have:

$$y(x, s) = \frac{1-x}{s} + \frac{1}{s} L \left[\frac{\partial^2 y}{\partial x^2} + y \frac{\partial y}{\partial x} \right] \tag{37}$$

The Laplace transform's inverse suggests that:

$$y(x, t) = 1 - x + L^{-1} \left[\frac{1}{s} L \left[\frac{\partial^2 y}{\partial x^2} + y \frac{\partial y}{\partial x} \right] \right] \tag{38}$$

Now, we apply the HPM; we have

$$\begin{aligned}
 \sum_{n=0}^{\infty} p^n y_n(x, t) &= 1 - x + p \left(L^{-1} \left[\frac{1}{s} \left\{ L \left[\frac{\partial^2 y}{\partial x^2} \right] + \right. \right. \right. \\
 &\left. \left. \left. L \left[\sum_{n=0}^{\infty} p^n H_n(y) \right] \right\} \right] \right),
 \end{aligned} \tag{39}$$

where $H_n(y)$ are He's polynomials. The first few components of He's polynomials are:

$$\begin{aligned}
 H_0(y) &= y_0 \frac{\partial y_0}{\partial x} = -(1 - x) \\
 H_1(y) &= y_0 \frac{\partial y_1}{\partial x} + y_1 \frac{\partial y_0}{\partial x} = 2(1 - x)t \\
 H_2(y) &= y_0 \frac{\partial y_2}{\partial x} + y_1 \frac{\partial y_1}{\partial x} + y_2 \frac{\partial y_0}{\partial x} = -3(1 - x)t^2
 \end{aligned} \tag{40}$$

The coefficient of similar powers of p is compared, and we have:

diminishes as time progresses. This decreasing trend is evident in the lines representing $t = 0.2, t = 0.4, t = 0.6,$ and $t = 0.8$. However, an interesting anomaly is observed in the curve for $t = 0$, where the solution starts with $y = 1$ and decreases as t increases. This reversed trend in the line plot suggests a unique behavior or specific conditions that influence the solution's initial behavior. Another noteworthy observation in Figure 5 is the significant increase in the solution's value at $t = 0.8$. This sharp increase indicates a sudden change or a critical point in the solution's behavior. It suggests that the solution

becomes more pronounced or exhibits a higher amplitude at this specific time point.

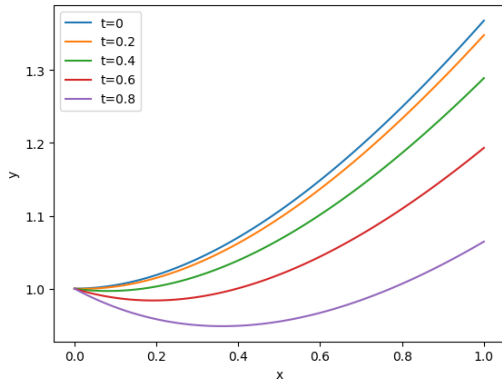


Figure 3. Exact solution of example 4.2 using HPM-LTM method for varying values of t , plotted in Python using Sympy and Matplotlib for different values of t

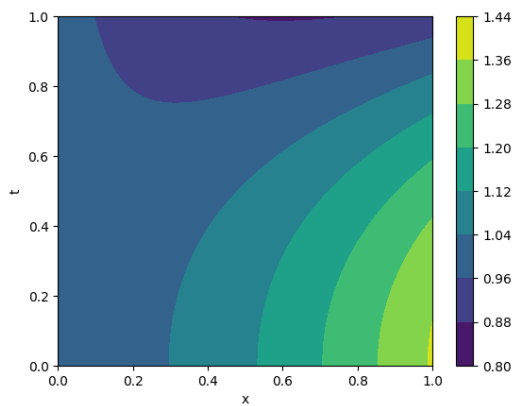


Figure 4. 2D contour plot of the exact solution of example 4.2 using the HPM-LTM method, plotted in Python using Sympy and Matplotlib

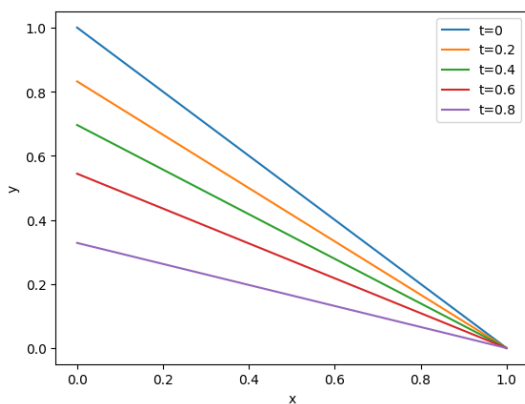


Figure 5. Exact solution of example 4.3 using HPM-LTM method for varying values of t , plotted in Python using Sympy and Matplotlib for different values of t

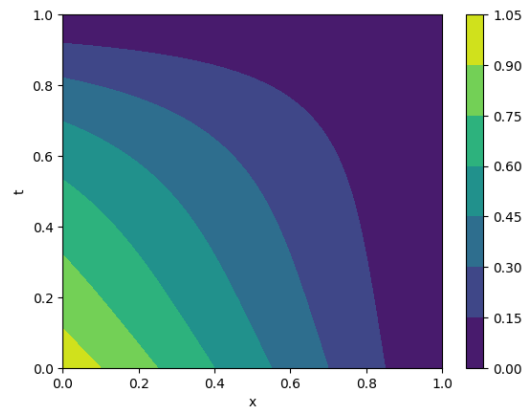


Figure 6. 2D contour plot of the exact solution of example 4.3 using the HPM-LTM method, plotted in Python using Sympy and Matplotlib

Additionally, all the lines in the graph converge at $x = 1$, indicating that the behavior of the solution remains consistent across different values of t as it approaches this point. This convergence implies that the solution reaches a stable state or exhibits similar characteristics near $x = 1$, regardless of the specific value of t . It suggests that the solution may have a specific behavior or property that is invariant with respect to changes in t near this point.

Figure 6 appears to be a 2D contour plot that provides further insights into the behavior of the solution for different values of t and x . The contour lines on the plot represent different levels of the solution and allow us to visualize how the values change across various regions of the $x - t$ axes. By examining the contour lines in Figure 6, we can identify variations in the solution across the $x - t$ parameter space. This visualization lets us identify regions with similar solution values and observe patterns, trends, and critical points. The contour plot helps us understand how the solution changes across different values of t and x . It provides a comprehensive overview of the solution's behavior and allows us to identify areas where the solution exhibits significant changes or remains relatively constant. By analyzing the contour lines, we can identify regions with steep gradients or areas where the solution transitions rapidly. This information is crucial for understanding the behavior and characteristics of the solution in different parts of the $x-t$ parameter space.

Figure 5 illustrates the behavior of the solution for various values of t , showing a decreasing trend overall with an interesting reversal in the curve for $t = 0$. Figure 6, on the other hand, provides a 2D contour plot that offers insights into the behavior of the solution across different values of t and x . The contour lines help to identify the solution's variations, patterns, and critical points. These visualizations enhance our understanding of the physical behavior and properties of the solution in the context of the given problem.

Example 4.4. Take into account the aforementioned homogeneous nonlinear PDE (26, 36, 38):

$$\frac{\partial y}{\partial t} - y - y \frac{\partial^2 y}{\partial x^2} - \left(\frac{\partial y}{\partial x}\right)^2 = 0 \tag{44}$$

with the following B.Cs.:

$$y(x, 0) = \sqrt{x}, y(0, t) = 0, y(1, t) = e^t. \tag{45}$$

Applying the procedure above while keeping in mind the starting state, we have:

$$y(x, s) = \frac{\sqrt{x}}{s} + \frac{1}{s} L \left[y + y \frac{\partial^2 y}{\partial x^2} + \left(\frac{\partial y}{\partial x}\right)^2 \right] \tag{46}$$

$$H_0(y) = y_0 \frac{\partial^2 y_0}{\partial x^2} + \left(\frac{\partial y_0}{\partial x}\right)^2 = 0$$

$$H_1(y) = y_0 \frac{\partial^2 y_1}{\partial x^2} + y_1 \frac{\partial^2 y_0}{\partial x^2} + 2 \frac{\partial y_0}{\partial x} \frac{\partial y_1}{\partial x} = 0$$

$$H_2(y) = y_0 \frac{\partial^2 y_2}{\partial x^2} + y_1 \frac{\partial^2 y_1}{\partial x^2} + y_2 \frac{\partial^2 y_0}{\partial x^2} + \left(\frac{\partial y_1}{\partial x}\right)^2 + 2 \frac{\partial y_0}{\partial x} \frac{\partial y_2}{\partial x} = 0$$

The coefficient of similar powers of p is compared, and we have:

$$p^0: y_0(x, t) = \sqrt{x}$$

$$p^1: y_1(x, t) = L^{-1} \left[\frac{1}{s} \{L[y_0] + L[H_0(y)]\} \right] = \sqrt{x}t \tag{50}$$

$$p^2: y_2(x, t) = L^{-1} \left[\frac{1}{s} \{L[y_1] + L[H_1(y)]\} \right] = \frac{\sqrt{x}t^2}{2!}.$$

$$p^3: y_3(x, t) = \frac{\sqrt{x}t^3}{3!}, \tag{51}$$

$$p^4: y_4(x, t) = \frac{\sqrt{x}t^4}{4!},$$

so the solution $y(x, t)$ is given by:

$$y(x, t) = y_0 + y_1 + y_2 + y_3 + \dots$$

$$= \sqrt{x} + \frac{\sqrt{x}t}{1!} + \frac{\sqrt{x}t^2}{2!} + \frac{\sqrt{x}t^3}{3!} + \dots$$

$$= \sqrt{x} \left(1 + \frac{t}{1!} + \frac{t^2}{2!} + \frac{t^3}{3!} + \dots + \frac{t^n}{n!} + \dots \right)$$

$$= \sqrt{x}e^t, \tag{52}$$

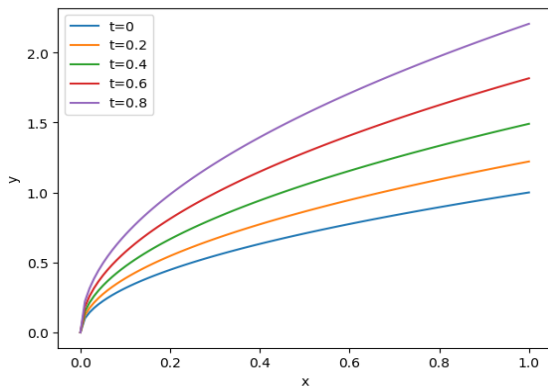


Figure 7. Exact solution of example 4.4 using HPM-LTM method for varying values of t, plotted in Python using Sympy and Matplotlib for different values of t

The Laplace transform's inverse suggests that:

$$y(x, t) = \sqrt{x} + L^{-1} \left[\frac{1}{s} L \left[y + y \frac{\partial^2 y}{\partial x^2} + \left(\frac{\partial y}{\partial x}\right)^2 \right] \right] \tag{47}$$

Now, we apply the HPM; we have

$$\sum_{n=0}^{\infty} p^n y_n(x, t) = \sqrt{x} + p \left(L^{-1} \left[\frac{1}{s} \{L[y] + L[\sum_{n=0}^{\infty} p^n H_n(y)]\} \right] \right) \tag{48}$$

where $H_n(y)$ are He's polynomials. The first few components of He's polynomials are given by:

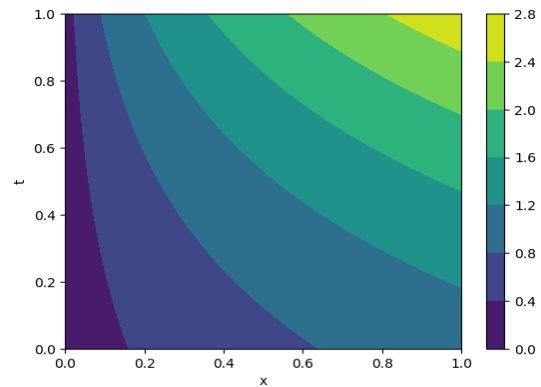


Figure 8. 2D contour plot of the exact solution of example 4.4 using the HPM-LTM method, plotted in Python using Sympy and Matplotlib

which is the exact solution to the problem (26, 36, 38). In some experiments, different approximate solutions are obtained by using HPM (36, 38). By applying HPM-LTM, we obtain exact solutions.

Example 4.5. Take into account the subsequent non-homogeneous nonlinear PDE (26, 36, 38):

$$\frac{\partial^2 y}{\partial t^2} + \frac{\partial^2 y}{\partial x^2} + \left(\frac{\partial y}{\partial x}\right)^2 = 2x + t^4 \tag{53}$$

with the following B.Cs.:

$$y(x, 0) = 0, \frac{\partial y}{\partial t}(x, 0) = a, y(0, t) = at, \frac{\partial y}{\partial x}(0, t) = t^2 \tag{54}$$

By applying the HPM subject to the I.Cs., we have:

$$y(x, s) = \frac{a}{s^2} - \frac{1}{s^2} L \left[\frac{\partial^2 y}{\partial x^2} + \left(\frac{\partial y}{\partial x}\right)^2 \right] + \frac{1}{s^2} L[2x + t^4]$$

$$= \frac{a}{s^2} + \frac{2x}{s^3} + \frac{4!}{s^7} - \frac{1}{s^2} L \left[\frac{\partial^2 y}{\partial x^2} + \left(\frac{\partial y}{\partial x}\right)^2 \right]. \tag{55}$$

The Laplace transform's inverse suggests that:

$$y(x, t) = at + xt^2 + \frac{t^6}{30} - L^{-1} \left[\frac{1}{s^2} L \left[\frac{\partial^2 y}{\partial x^2} + \left(\frac{\partial y}{\partial x} \right)^2 \right] \right]. \quad (56)$$

Now, we apply the HPM, we have:

$$\sum_{n=0}^{\infty} p^n y_n(x, t) = at + xt^2 + \frac{t^6}{30} - p \left(L^{-1} \left[\frac{1}{s^2} \left\{ L \left[\frac{\partial^2 y}{\partial x^2} \right] + L \left[\sum_{n=0}^{\infty} p^n H_n(y) \right] \right\} \right] \right) \quad (57)$$

where $H_n(y)$ are He's polynomials. The first few components of He's polynomials are given by:

$$\begin{aligned} H_0(y) &= \left(\frac{\partial y_0}{\partial x} \right)^2 = t^4, \\ H_1(y) &= 2 \left(\frac{\partial y_0}{\partial x} \right) \times \left(\frac{\partial y_1}{\partial x} \right) = 0, \\ H_2(y) &= \left(\frac{\partial y_1}{\partial x} \right)^2 + 2 \frac{\partial y_0}{\partial x} \frac{\partial y_2}{\partial x} = 0, \end{aligned} \quad (58)$$

The coefficient of similar powers of p is compared, and we have:

$$\begin{aligned} p^0: y_0(x, t) &= at + xt^2 + \frac{t^6}{30}, \\ p^1: y_1(x, t) &= -L^{-1} \left[\frac{1}{s^2} \left\{ L \left[\frac{\partial^2 y_0}{\partial x^2} \right] + L[H_0(y)] \right\} \right] = -\frac{t^6}{30}, \\ p^2: y_2(x, t) &= -L^{-1} \left[\frac{1}{s^2} \left\{ L \left[\frac{\partial^2 y_1}{\partial x^2} \right] + L[H_1(y)] \right\} \right] = 0. \end{aligned} \quad (59)$$

$$\begin{aligned} p^3: y_3(x, t) &= 0, \\ p^4: y_4(x, t) &= 0, \end{aligned} \quad (60)$$

so that the solution $y(x, t)$ is given by:

$$\begin{aligned} y(x, t) &= y_0 + y_1 + y_2 + y_3 + \dots \\ &= at + xt^2 + \frac{t^6}{30} - \frac{t^6}{30} + 0 + 0 + 0 + \dots \\ &= at + xt^2 \end{aligned} \quad (61)$$

which is the exact solution to the problem (26, 36, 38). In some experiments, different approximate solutions are obtained by using HPM (36, 38). By applying HPM-LTM, we obtain exact solutions. Figure 9 displays the exact solution of Example 4.5 for various values of a (0, 0.2, 0.4, 0.6, 0.8), with y represented on the vertical axis and x on the horizontal axis. The equation for the exact solution is given as $at + xt^2$, where a is a parameter that influences the behavior of the solution. Analyzing the graph in Figure 9, we observe that as the value increases, the corresponding y values also increase. This relationship indicates that higher values of a result in larger magnitudes of y . The curve for $a = 0.8$ exhibits the highest y values, while the curve for $a = 0$ demonstrates the smallest values. This trend suggests that the parameter a significantly impacts the amplitude or magnitude of the solution.

Figure 10 presents a 2D contour plot illustrating the behavior of the exact solution of Example 4.5 for different values of a and x . The vertical axis represents the values of a , while the horizontal axis represents the values of x . The contour lines depicted on the plot represent different levels of the solution, enabling us to visualize how the values change across various regions of the $x - a$ parameter space.

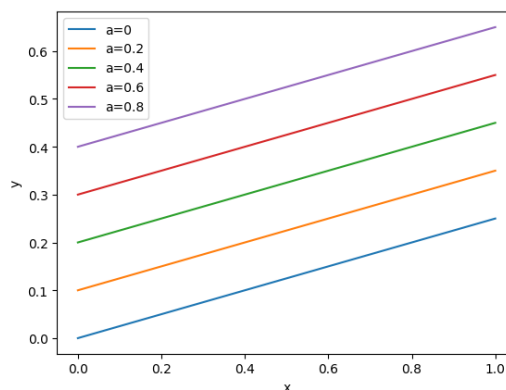


Figure 9. Exact solution of example 4.5 using HPM-LTM method for varying values of t , plotted in Python using Sympy and Matplotlib for different values of a

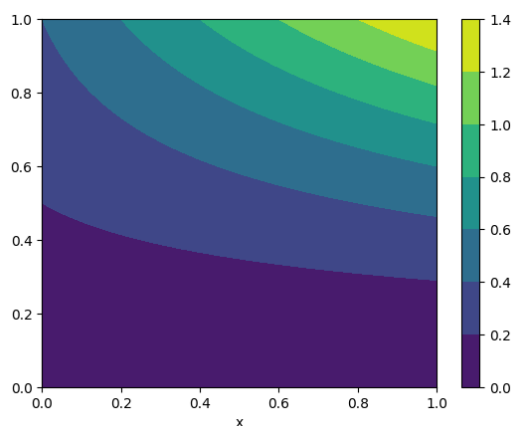


Figure 10. 2D contour plot of the exact solution of example 4.5 using the HPM-LTM method, plotted in Python using Sympy and Matplotlib

By examining the contour lines in Figure 10, we can identify regions where the solution changes more rapidly. In areas where the contour lines are closer together, the solution exhibits significant variations or transitions over a small range of x and a values. Conversely, regions with widely spaced contour lines correspond to areas where the solution changes more gradually or remains relatively constant. This 2D contour plot demonstrates that the behavior of the exact solution is highly dependent on the values of a and x . The contour lines' proximity indicates the solution's sensitivity to changes in these parameters. It suggests that small variations in a and x can substantially change the solution's magnitude or behavior.

Despite the complexity of the solution, the contour plot provides an easily interpretable visual representation of the data. It allows us to identify critical regions, observe trends, and understand the relationship between the solution levels and the values of a and x . This visualization aids in gaining insights into the behavior and properties of the exact solution in the context of Example 4.5.

In our research paper, we compared the efficiency of the HPM-LTM and traditional HPM methods. We performed this comparison by solving example 4.5 using both approaches. While both methods yielded the exact solution, our observations indicated that the HPM-LTM method significantly outperformed the HPM method. Using the HPM-LTM method, we achieved the desired result with just seven blocks of code. In contrast, the HPM method required 48 blocks of code. This substantial difference in coding requirements demonstrates the efficiency gain offered by the HPM-LTM method. The improved efficiency of the HPM-LTM method is attributed to its optimized design, which provides a streamlined and less convoluted approach to problem-solving. The HPM-LTM method simplifies coding by eliminating unnecessary complexities and reducing coding requirements.

Our study unequivocally demonstrates that the HPM-LTM method is superior in efficiently and effectively solving complex problems compared to traditional methods. Our research findings highlight the advantages of adopting the HPM-LTM method for improved efficiency and effectiveness in problem-solving endeavors.

5. CONCLUSION

The proposed HPM-LTM method offers a novel and effective approach for solving linear and nonlinear partial differential equations, as demonstrated through illustrative heat and wave equations examples. By combining the Homotopy Perturbation Method with the Laplace Transform Method, the HPM-LTM method can handle nonlinear terms in equations while avoiding the limitations of traditional methods. Furthermore, Python's computational capabilities enable fast and accurate implementation of the HPM-LTM method. It is a practical computational tool for solving PDE problems in various domains, especially engineering, where efficient calculation time and large data volumes are essential. This study contributes to the advancement of computational tools for solving PDEs, and future research could explore the potential of the HPM-LTM method for more complex PDE problems. In summary, the HPM-LTM method offers a promising approach for solving PDEs with high accuracy and efficiency.

6. REFERENCES

- He J-H. Homotopy perturbation technique. *Computer methods in applied mechanics and engineering*. 1999;178(3-4):257-62. [https://doi.org/10.1016/S0045-7825\(99\)00018-3](https://doi.org/10.1016/S0045-7825(99)00018-3)
- He J-H. Homotopy perturbation method for solving boundary value problems. *Physics letters A*. 2006;350(1-2):87-8. <https://doi.org/10.1016/j.physleta.2005.10.005>
- He J-H. Homotopy perturbation method: a new nonlinear analytical technique. *Applied Mathematics and computation*. 2003;135(1):73-9. [https://doi.org/10.1016/S0096-3003\(01\)00312-5](https://doi.org/10.1016/S0096-3003(01)00312-5)
- Shateri A, Moghaddam MM, Jalili B, Khan Y, Jalili P, Ganji DD. Heat transfer analysis of unsteady nanofluid flow between moving parallel plates with magnetic field: Analytical approach. *Journal of Central South University*. 2023;30(7):2313-23. <https://doi.org/10.1007/s11771-023-5388-3>
- LeVeque RJ. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*: SIAM; 2007.
- Khater M, Anwar S, Tariq KU, Mohamed MS. Some optical soliton solutions to the perturbed nonlinear Schrödinger equation by modified Khater method. *AIP Advances*. 2021;11(2). <https://doi.org/10.1063/5.0038671>
- Sedighi HM, Shirazi KH, Zare J. An analytic solution of transversal oscillation of quintic non-linear beam with homotopy analysis method. *International Journal of Non-Linear Mechanics*. 2012;47(7):777-84. <https://doi.org/10.1016/j.ijnonlinmec.2012.04.008>
- El-Shahed M. Application of He's homotopy perturbation method to Volterra's integro-differential equation. *International Journal of Nonlinear Sciences and Numerical Simulation*. 2005;6(2):163-8. <https://doi.org/10.1515/IJNSNS.2005.6.2.163>
- Van Der Walt S, Colbert SC, Varoquaux G. The NumPy array: a structure for efficient numerical computation. *Computing in science & engineering*. 2011;13(2):22-30. <https://doi.org/10.1109/MCSE.2011.37>
- Nadeem M, Li F. He-Laplace method for nonlinear vibration systems and nonlinear wave equations. *Journal of Low Frequency Noise, Vibration and Active Control*. 2019;38(3-4):1060-74. <https://doi.org/10.1177/1461348418818>
- Filobello-Nino U, Vazquez-Leal H, Sarmiento-Reyes A, Cervantes-Perez J, Perez-Sesma A, Jimenez-Fernandez V, et al. Laplace transform-homotopy perturbation method with arbitrary initial approximation and residual error cancellation. *Applied Mathematical Modelling*. 2017;41:180-94. <https://doi.org/10.1016/j.apm.2016.08.003>
- Liu Z-J, Adamu MY, Suleiman E, He J-H. Hybridization of homotopy perturbation method and Laplace transformation for the partial differential equations. *Thermal Science*. 2017;21(4):1843-6. <https://doi.org/10.2298/TSCI160715078L>
- Jalili B, Ganji AD, Jalili P, Nourazar SS, Ganji D. Thermal analysis of Williamson fluid flow with Lorentz force on the stretching plate. *Case Studies in Thermal Engineering*. 2022;39:102374. <https://doi.org/10.1016/j.csite.2022.102374>
- Jalili B, Sadighi S, Jalili P, Ganji DD. Numerical analysis of MHD nanofluid flow and heat transfer in a circular porous medium containing a Cassini oval under the influence of the Lorentz and buoyancy forces. *Heat Transfer*. 2022;51(7):6122-38. <https://doi.org/10.1002/htj.22582>
- Jalili B, Aghaee N, Jalili P, Ganji DD. Novel usage of the curved rectangular fin on the heat transfer of a double-pipe heat exchanger with a nanofluid. *Case Studies in Thermal Engineering*. 2022;35:102086. <https://doi.org/10.1016/j.csite.2022.102086>
- Endalew MF, Sarkar S. Numerical exploration of forced convection hydromagnetic hyperbolic tangent nanofluid flow over a permeable wedge with melting heat transfer. *Scientific Reports*. 2023;13(1):3515. <https://doi.org/10.1038/s41598-023-30656-2>
- Chamkha A, Ismael M, Kasaeipoor A, Armaghani T. Entropy generation and natural convection of CuO-water nanofluid in C-shaped cavity under magnetic field. *Entropy*. 2016;18(2):50. <https://doi.org/10.3390/e18020050>
- Jalili B, Mousavi A, Jalili P, Shateri A, Domiri Ganji D. Thermal analysis of fluid flow with heat generation for different

- logarithmic surfaces. *International Journal of Engineering, TRANSACTIONS C: Aspects*. 2022;35(12):2291-6. 10.5829/IJE.2022.35.12C.03
19. Jalili P, Narimisa H, Jalili B, Shateri A, Ganji D. A novel analytical approach to micro-polar nanofluid thermal analysis in the presence of thermophoresis, Brownian motion and Hall currents. *Soft Computing*. 2023;27(2):677-89. <https://doi.org/10.1007/s00500-022-07643-2>
 20. Jalili B, Jalili P, Shateri A, Ganji DD. Rigid plate submerged in a Newtonian fluid and fractional differential equation problems via Caputo fractional derivative. *Partial Differential Equations in Applied Mathematics*. 2022;6:100452. <https://doi.org/10.1016/j.padiff.2022.100452>
 21. Jalili P, Jalili B, Shateri A, Domiri Ganji D. A novel fractional analytical technique for the time-space fractional equations appearing in oil pollution. *International Journal of Engineering, TRANSACTIONS C: Aspects*. 2022;35(12):2386-94. 10.5829/IJE.2022.35.12C.15
 22. Abbasbandy S. Iterated He's homotopy perturbation method for quadratic Riccati differential equation. *Applied Mathematics and Computation*. 2006;175(1):581-9. <https://doi.org/10.1016/j.amc.2005.02.014>
 23. Kiusalaas J. *Numerical methods in engineering with Python 3*: Cambridge university press; 2013.
 24. Johansson R. *Numerical Python: Scientific Computing and Data Science Applications with Numpy, SciPy and Matplotlib*, Apress, Berkeley, CA, 2019. DOI: <https://doi.org/10.1007/978-1-4842-4246-9>.
 25. Hunter JD. *Matplotlib: A 2D graphics environment*. *Computing in science & engineering*. 2007;9(03):90-5. 10.1109/MCSE.2007.55
 26. Liao S, Tan Y. A general approach to obtain series solutions of nonlinear differential equations. *Studies in Applied Mathematics*. 2007;119(4):297-354. <https://doi.org/10.1111/j.14679590.2007.00387.x>
 27. Momani S, Odibat Z. Homotopy perturbation method for nonlinear partial differential equations of fractional order. *Physics Letters A*. 2007;365(5-6):345-50. <https://doi.org/10.1016/j.physleta.2007.01.046>
 28. Kumar Mishra H, Nagar AK. He-Laplace method for linear and nonlinear partial differential equations. *Journal of Applied Mathematics*. 2012;2012. <https://doi.org/10.1155/2012/180315>
 29. Shijing D, Hongru C, Xudong W, Deshi W, Yongyong Z. Modal optimization design of supporting structure based on the improved particle swarm algorithm. *International Journal of Engineering, TRANSACTIONS A: Basics*. 2022;35(4):740-9. 10.5829/IJE.2022.35.04A.14
 30. Jalili P, Shateri A, Ganji AM, Jalili B, Ganji DD. Analytical analyzing mixed convection flow of nanofluid in a vertical channel using python approach. *Results in Physics*. 2023;52:106908. <https://doi.org/10.1016/j.rinp.2023.106908>
 31. Hamidi H, Seyed Lotfali S. Analysis of role of cloud computing in providing internet banking services: Case study bank melli iran. *International Journal of Engineering, TRANSACTIONS B: Applications*. 2022;35(5):1082-8. 10.5829/IJE.2022.35.05B.23
 32. Jalili B, Ganji AM, Shateri A, Jalili P, Ganji DD. Thermal analysis of non-Newtonian visco-elastic fluid MHD flow between rotating disks. *Case Studies in Thermal Engineering*. 2023;49:103333. <https://doi.org/10.1016/j.csite.2023.103333>
 33. Marchione F. Analytical investigation on the stress distribution in structural elements reinforced with laminates subjected to axial loads. *International Journal of Engineering, TRANSACTIONS A: Basics*. 2022;35(4):692-7. 10.5829/IJE.2022.35.04A.08
 34. Agrawal V, Khairnar H. Analytical Modeling of Heat Transfer Coefficient Analysis in Dimensionless Number of an Electric Parking Brake Using CFD. *International Journal of Engineering, TRANSACTIONS B: Applications*. 2023;36(2):276-88. 10.5829/IJE.2023.36.02B.08
 35. Lap-Arparat P, Tuchinda K. Computational Study of Excitation Controlling Parameters Effect on Uniform Beam Deformation under Vibration. *International Journal of Engineering, TRANSACTIONS A: Basics*. 2023;36(1):60-70. 10.5829/IJE.2023.36.01A.08
 36. Baughman G, Sharma P. Entropic analysis of protein oscillations through langevin equations & fokker-planck equations. *Journal of Human, Earth, and Future*. 2022;3:56-70. 10.28991/HEF-SP2022-01-05
 37. Santantyo T, Ripai A, Abdullah Z, Hidayat W, Zen FP. Soliton-like solution on the dynamics of modified Peyrard-bishop DNA model in the thermostat as a bio-fluid. *Emerg Sci J (ISSN: 2610-9182)*. 2022;6(4). 10.28991/ESJ-2022-06-04-01
 38. Alrwashdeh SS, Ammari H, Madanat MA, Al-Falahat AaM. The effect of heat exchanger design on heat transfer rate and temperature distribution. *Emerging Science Journal*. 2022;6(1):128-37. 10.28991/ESJ-2022-06-01-010

COPYRIGHTS

©2024 The author(s). This is an open access article distributed under the terms of the Creative Commons Attribution (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, as long as the original authors and source are cited. No permission is required from the authors or the publishers.

**Persian Abstract****چکیده**

این تحقیق یک روش دقیق و نوآورانه به نام روش اختلالی هماتویی - روش تبدیل لاپلاس (HPM-LTM) که با استفاده از زبان پایتون پیاده‌سازی شده است، برای حل کارآمد معادلات دیفرانسیل جزئی خطی و غیرخطی (PDEs) معرفی می‌کند. با ترکیب تکنیک اختلالی هماتویی (HPM) با روش تبدیل لاپلاس (LTM)، روش ما با موفقیت به معاملات مهمی که توسط معادلات دارای اجزای غیرخطی ایجاد می‌شود، پاسخ می‌دهد. با استفاده از چند جمله‌ای‌های هس، رویکرد HPM-LTM با دقت اصطلاحات غیرخطی را مدیریت می‌کند و به راه‌حل‌های دقیق و قابل اعتماد منجر می‌شود. برای نشان دادن کارایی روش ما، ما آن را به طور گسترده در پنج سناریو نماینده از PDE، از جمله معادلات حرارت و موج، به کار می‌بریم. نتایج جامع ما دقت و قابلیت اعتماد قابل توجه روش HPM-LTM را تأیید می‌کند و برتری آن نسبت به روش‌های معمولی که نیاز به فرضیات محدود کننده یا گسسته‌سازی دارند و ممکن است خطاهای گردابی ایجاد کنند، را مورد تأکید قرار می‌دهد. علاوه بر این، روش ما محدودیت‌های ناشی از خطاهای عددی موجود در تکنیک‌های هماتویی سنتی را برطرف می‌کند. قدرت، کارایی و انعطاف‌پذیری روش پیشنهادی ما از طریق استفاده موفق از آن در مجموعه‌ای گسترده از مسائل PDE در زمینه‌های مختلف تأیید می‌شود. این تحقیق یک مشارکت مهم به توسعه ابزار محاسباتی قدرتمند برای حل مسائل متنوع PDE با تأکید خاص بر رشته مهندسی معرفی می‌کند.