Advances in Engineering Software 170 (2022) 103160

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Application of He's homotopy and perturbation method to solve heat transfer equations: A python approach

Pankaj Dumka [*], Parth Singh Pawar, Abhay Sauda, Gaurav Shukla, Dhananjay R. Mishra

Department of Mechanical Engineering, Jaypee University of Engineering and Technology, Guna, 473226, Madhya Pradesh, India

ARTICLE INFO

Keywords:
Heat transfer
Homotopy and perturbation method (HPM)
Nonlinear differential equations
Python
SymPy

ABSTRACT

In this research article, an attempt has been made to solve the linear/nonlinear and steady/unsteady heat transfer equations using the Homotopy and Perturbation method (HPM). Moreover, the implementation of HPM has been done by using SymPy, a library in python, to solve problems symbolically. Total three problems were dealt viz. steady-state conduction with heat generation, lumped capacitance analysis with a variable specific heat of the material, and heat transfer in uniform rectangular fin with radiation from the surface. In all the cases, the HPM has given excellent results compared to the analytical and numerical. Finally, the execution of SymPy has been explained, and a detailed procedure to implement HPM through python has been presented for all three cases.

1. Introduction

Everything is self-explanatory by some mathematical relationship in the core of all the different engineering specialists. Most of these problems and situations are characterized by linear and nonlinear equations. As in case of heat transfer most of the problems are of nonlinear nature. So, except for a small number of these problems, it isn't easy to find specific analytical solutions. Consequently, appropriate analytical and delivery solutions are sought, most effective and easy to use, namely the Adomian Decomposition Method (ADM) [1–3], Variational Therapy Method (VIM) [4–6], and the Homotopy Perturbation Method (HPM) [7] in both weak and robust nonlinear equations. The perturbation method is one of the most flexible tools available in analysing the problem of indirect engineering, but its limitations limit its use [8]. The perturbation method is based on the assumption of a small parameter [9,10] because stability and integration should be considered numerically to avoid different or negative results. Weighted Residual Methods (WRMs) and Least Square Methods (LSM) [11] are some simple and accurate measurement techniques for solving various nonlinear calculations.

Dr Ji Huan He initially developed a method of HPM [12–17], which has been used to solve many types of linear and nonlinear functional operations. This approach combines topology, homotopy [18], and general perturbation techniques[19], allowing us to find analytical or speculative solutions to many different problems from different domains. The technique produces solutions for a dynamic series of non-linear separation systems using the concept of homotopy [20]. These methods include a complete selection of the control parameter, initial measurement, an auxiliary line operator, and an auxiliary function in the use of HPM for non-linear separation problems [21]. As a result, all the most suitable HPM strategies significantly accelerated the integration of series solutions into different non-linear measurements [22]. Ganji et al. [27] have reported the use of HPM to solve the Burgers equation observed in fluid mechanics. For nonlinear oscillators, Yu et al. [23] have reported a review in which they have focused on a linear term embedded auxiliary parameter. To study the nonlinear oscillator containing damping, He and El-Dib [24] have reported modifications in the HPM. For solving fractional differential equations, Nadeem et al. [25] have reported the use of HPM. Ashrafi et al. [26] have reported the use of HPM to solve for the fluid flow (Newtonian Fluid) past as flat porous plate.

When it comes to implementation of any numerical model or analytical/semi-analytical model a lot of pen and paper work which has to be done to setup the differential equations and then they are discretized and solved either numerically or analytically [28]. This process requires a lot of mechanical work. Here comes the importance of python language to reduce the solution and implementation task [29,30]. SymPy is a module which is used in python (as well written in python) to solve mathematics symbolically [31–33]. This is so powerful tool that

Abbreviations: ADM, Adomian Decomposition Method; VIM, Variational Therapy Method; HPM, Homotopy Perturbation Method; WRMs, Weighted Residual Methods; LSM, Least Square Methods.

[*] Corresponding author.
E-mail address: p.dumka.ipec@gmail.com (P. Dumka).

https://doi.org/10.1016/j.advengsoft.2022.103160
Received 11 May 2022; Received in revised form 21 May 2022; Accepted 26 May 2022
Available online 4 June 2022

Nomenclature

A	General differential operator
B	Boundary operator
u	Dependent variable
f(r)	Any known function
L	Linear operator
N	Nonlinear operator
H(y,p)	Homotopy
p	Homotopy parameter
$\boldsymbol{Y}_0$	Initial guess
T	Temperature
t	Time
q	Heat generation
k	Thermal conductivity
V	volume
c	Specific heat
h	Convective heat transfer coefficient
A'	Surface area

Greek

Ω	Domain
Γ	Domain boundary
ρ	Density
θ	Nondimensional temperature

one has to just plug in the differential model in symbolic form (as we write on paper) and it will return the output. With the help of SymPy one can write a comprehensible code which can be easily extended while keeping the code as simple as possible. With the help of sympy multiple tasks like expression simplification, derivative & integral computations, limit evaluation, equation solutions, matrix operation, ordinary differential equation (ODE) solution and much more can be done symbolically with only few lines of code [34]. Several authors have reported the use of Python for solving different types of problems efficiently. To facilitate cross-institutional collaboration and democratise research progress, Sulzer et al. [35] have developed and reported the Python package PyBaMM. Marowka [36] has reported the use of Python and its importance for parallel programming. A Python-based comparison analysis of different feature importance measures has been written by Saarela and Jauhiainen [37]. Pawar et al. [38] have reported the use of Python to handle a complex problems like viscous fluid flow. They have developed modules that can solve the parallel fluid flow problems quickly and with no error.

After studying the HPM, we realize that the concept is simple, but everyone has dealt with each of their problems heuristically by using various strategies. While this shows the flexibility of the method, the reader experiences difficulties when using it. Therefore, in this manuscript attempts have been done to make the concept more general and provide some guidance to implement the HPM computationally. The indirect equation of continuous heat transfer with apparent flexible parameters is solved using the HPM. The results thus obtained are compared with the analytical/numerical solution. The implementation of HPM is done with the help of SymPy and the codes written to solve the equations are presented so that reader be benefitted.

2. Homotopy perturbation method

In this manuscript, Homotopy Perturbation Method (HPM), which was originally proposed by He [12,39] has been applied to solve differential equations which are seen in heat transfer. To understand HPM methodology, consider the following differential equation [39]:

$$\mathrm{A}(u) - \mathrm{f}(r) = 0, \mathrm{r} \in \Omega \tag{1}$$

with boundary conditions,

$$B(u, \delta u / \delta n) = 0,\ r \in \Gamma \tag{2}$$

where A is any general differential operator, *f*(*r*) is some known function, B is any boundary operator, and Ω is the domain which is bounded by Γ.

In HPM the operator A can be generally divided into two parts L the linear and N the nonlinear.

So, Eq. (1) can now be rewritten as:

$$L(u) + N(u) - (r) = 0 \tag{3}$$

The good aspect of homotopy method is that it is independent of ‘small parameter’. Now a homotopy can be constructed with the help of a homotopy parameter *p*, auxiliary function *L*& N, and initial guess $\boldsymbol{Y}_0$as follows:

$$H(y,p) = L(y) - L(\boldsymbol{Y}_0) + p[L(y_0) + N(y) - f(r)] = 0 \tag{4}$$

According to the HPM, an approximation solution of the above equation i.e., Eq. (4) can be written a power series of p as follows:

$$u = \mathrm{p}^0\mathrm{y}_0 + \mathrm{p}^1\mathrm{y}_1 + \mathrm{p}^2\mathrm{y}_2 + \ldots \tag{5}$$

Which convents to the following approximate solution of Eq. (1) when p → 1:

$$u = \lim_{\mathrm{p}\to 1} y = \mathrm{y}_0 + \mathrm{y}_1 + \mathrm{y}_2 + \ldots \tag{6}$$

The point to note here is that the Eq. (6) is convergent in most of the cases resulting in the exact solution of original governing equation viz. Eq. (1). For obtaining the approximate solution, one can truncate the series if closed form solution is not arrived at.

While developing homotopy equation the most important part to care of is the selection of initial guess $\boldsymbol{Y}_0$and the *L* i.e., the auxiliary operator. The choice of L should be done in such a way that it does not pose any problem while solving resulting system of equations. Hence, it is also said that the choice of *L* is not necessarily be linear [40]. Moreover, the better option is to select some portion of the governing equation as *L*. The choice of $\boldsymbol{Y}_0$ is done in such a way it is the solution of some part of the original governing equation. Also, the choice of $\boldsymbol{Y}_0$should also reduce the complication of the resulting equations.

3. SymPy and differential equations

When it comes to symbolic computations in python, the SymPy module comes into picture. Ondrej Certik [31] was the first to introduce SymPy to the scientific community. This module is open source and can be used freely for symbolic computations. Sympy is very powerful when it comes to algebra, discrete mathematics, calculus etc. One of the attractive features of SymPy is its capability to format and present the results in LaTeX format.

For solving differential equations with the help of SymPy the following steps are adopted:

i. Import SymPy into the code (from sympy import *)
ii. Importing all the Greek and Latin letters as symbols (from sympy.abc import *)
iii. Create dependent variable as function of independent variable (eg. y = Function(‘y’)(x))

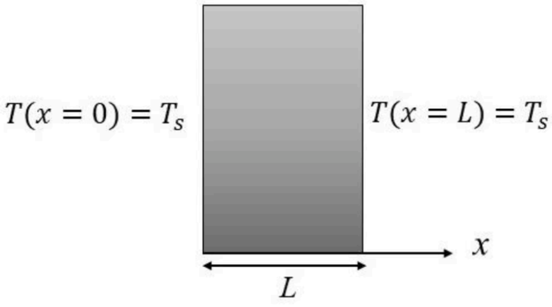


Fig. 1. Schematic of 1D slab.

iv. Setting up the boundary/initial conditions using‘.subs’ method (eq. ics={y.subs(x,0)=1})
v. Create a differential equation (eg. eqn=Eq(y.diff(x)+y,0))
vi. Using ‘dsolve’ for solving the differential equation (dsolve(eqn, y))

Let us understand the above procedure with the help of an example. Consider a differential equation $\frac{dy}{dx}+y=2$, with the boundary condition y(0)=1. Now the following few lines of code will solve this equation:

```
Code                                                   Output
# Step 1
from sympy import *
# Step 2
from sympy.abc import *
# Step 3
y=Function('y')(x)
# Step 4
eqn=Eq(y.diff(x)+y,2)
# Step 5
bc={y.subs(x,0):1}
# Step 6
dsolve(eqn,ics=bc)                                     Out[52]: y(x) = 2 − e^(−x)
```

In this manuscript SymPy has been used to implement Homotopy and Perturbation. This will not only ease the application procedure but will also help in systematic computation without any errors. All the functions and codes are written in the web-based interactive environment viz. Jupyter notebook.

4. Applications of HPM on heat transfer problems

In this manuscript in total three problems are solved to demonstrate the solution of 1-D heat conduction equations by HPM. These problems are:

- Steady state heat transfer in a slab with constant heat generation and thermal conductivity.
- Cooling of a system (assuming its specific heat is a function of temperature) considering it as a lumped system. This will make the governing equation of non-linear nature.
- Rectangular fin with radiation into ambient.

4.1. Heat transfer in a slab with constant heat generation and thermal conductivity

Fig. 1 shows a slab (in steady state) of two-meter length ($L=2$), constant thermal conductivity (k), and constant heat source per unit volume (q).

The generalized heat conduction equation in cartesian coordinates can be written as:

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{\partial^2 T}{\partial z^2}+\frac{q}{k}=\left(\frac{1}{\alpha}\right)\frac{\partial T}{\partial t} \tag{7}$$

As the problem is 1D and steady state so the second derivative terms in y and z along with time derivative will vanish which will result in:

$$\frac{\partial^2 T}{\partial x^2}+\frac{q}{k}=0 \tag{8}$$

With a boundary condition that $T(x=0)=Ts$ and $T(x=1)=Ts$. Where, Ts is some fixed temperature at both the ends.

To develop homotopy of the above equation let us assume the linear operator (L) to be $\frac{\partial^2}{\partial x^2}$. So, the homotopy equation becomes:

$$L(\theta)-L(\theta_0)+p\times L(\theta_0)+p\times\left(\frac{q}{k}\right)=0 \tag{9}$$

where, θ_0 is taken as Ts.

Let say the final approximate solution be $\theta=y_0+py_1+p^2y_2+p^3y_3\ldots.$

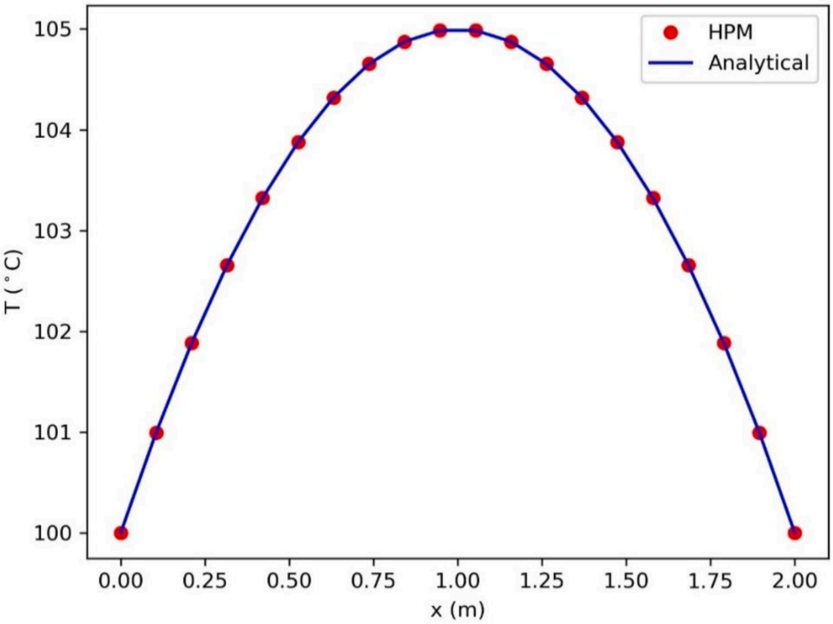


Fig. 2. Temperature variation along the length of slab.

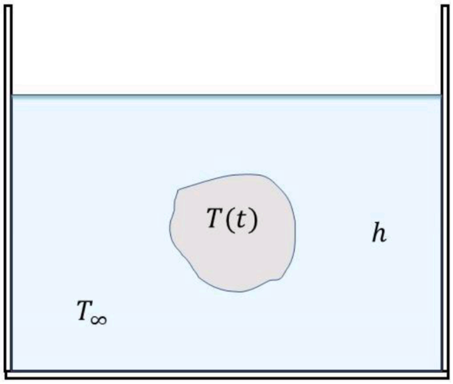


Fig. 3. Cooling of lumped system.

Now SymPy will be used in its full capacity to develop and the homotopy equation and its solution to arrive at the final answer.

```
from sympy import *
from sympy.abc import *
# Approximate solution Functions
y0,y1,y2=symbols('y0,y1,y2')
y0=Function('y0')(x)
y1=Function('y1')(x)
y2=Function('y2')(x)
theta=Function('theta')(x)
Ts=symbols('Ts')
theta0=Ts
# Linear differential function
L=lambda f: diff(f,x,x)
# Homotopy equation
eqn1=Eq(L(theta)-L(theta0)+p*L(theta0)+p*(q/k),0)
# Creating approximate solution
theta_prime=y0+p*y1+p**2*y2
eqn2=eqn1.subs(theta,theta_prime)
# Expanding the above equation for p and derivatives
eqn3=expand(eqn2.doit())
pprint(eqn3)
# Segregating coefficients of p
de0=Eq(y0.diff(x,x),0);ics0={y0.subs(x,0):Ts,y0.subs(x,2):
  Ts};
de1=Eq(eqn3.lhs.coeff(p**1),0); ics1={y1.subs(x,0):0,y1.subs
  (x,2):0};
de2=Eq(eqn3.lhs.coeff(p**2),0); ics2={y2.subs(x,0):0,y2.subs
  (x,2):0};
```

Solution of coefficients:

```
#for p^0
dsolve(de0,ics=ics0)                          Out [18]: y0(x) = Ts
# for p^1
de1=de1.subs(y0,Ts)
dsolve(de1,ics=ics1)                          Out [19]: y1(x) =
                                              x(-qx/2+q)
                                              ----------
                                                  k
# for p^2
de2=de2.subs({y0:Ts,y1:(x/k)*(-q*x/
  2+q)})
dsolve(de2,ics=ics2)                          Out [20]:y2(x) = 0
```

Therefore for $p = 1$the approximate solution of the Eq. (7) from HPM is:

$$T(x) = T_s + \left(\frac{x}{k}\right) \times \left(-\frac{qx}{2} + q\right) = T_s + \left(\frac{qx}{k}\right) \times \left(1 - \frac{x}{2}\right) \tag{10}$$

Which is exactly equal to the analytical solution of the problem [41]. Fig. 2 show the variation of temperature obtained by exact analytical solution and approximate solution obtained by HPM.

While plotting the results the values of k, q, and T_s are assumed to be 100.0 W/m-K, 1000 W/m^3, and 100°C respectively. The temperature variation is parabolic and symmetric about the centre line, this is due to the fact that the uniform heat source is there, and both ends of the slab are at the same temperature. One can see that the temperature obtained from the HPM is in very good agreement with the analytical result.

4.2. Analysis of lumped system

Let us consider a solid of surface area A, volume V, density ρ, and specific heat c is initially at some high temperature (T_0) and suddenly kept in a fluid (of convective heat transfer h) of lower temperature (T_∞). If the solid is having very small thermal resistance (infinite thermal conductivity), then it can be treated as a lumped system. The schematic of the problem is shown in Fig. 3. Then the governing equation of the whole lumped system can be written as:

$$\rho Vc\frac{dT}{d\tau} = \mathrm{hA}^{'}(T - T_\infty) \tag{11}$$

with the boundary condition that $T(\tau = 0) = T_0$

To make the equation nonlinear let us assume that the c is a linear function temperature which can be written as:

$$c = c_0[1 - \gamma(T - T_\infty)] \tag{12}$$

where,$C_0$is the specific heat, at temperature T_∞ and γ is a constant. By introducing below mentioned parameters the Eq. (11) takes the shape of Eq. (13).

$$\begin{aligned} &\theta = \frac{T - T_\infty}{T_0 - T_\infty}, \mathrm{t} = \frac{\tau hA'}{\rho Vc_0} \in = \gamma(T - T_\infty) \\ &\frac{d\theta}{d\mathrm{t}} + \in \theta\frac{d\theta}{d\mathrm{t}} + \theta = 0 \end{aligned} \tag{13}$$

With the boundary condition that $\theta(0) = 1$.

To develop homotopy of the above equation, let us assume the linear operator (L) to be $\frac{\partial}{\partial t} + 1$. So, the Homotopy equation becomes:

$$L(\theta) - L(\theta_0) + p \times L(\theta_0) + p \times \theta \times \in \times\frac{d\theta}{dt} = 0 \tag{14}$$

where θ_0 is taken as e^{-t} and the final approximate solution can be given by $\theta = y_0 + py_1 + p^2y_2 + p^3y_3\ldots.$

Now using sympy to solve the above Homotopy.

```
from sympy import *
from sympy.abc import *
# Approximate solution coefficients
y0,y1,y2=symbols('y0,y1,y2')
y0=Function('y0')(t)
y1=Function('y1')(t)
y2=Function('y2')(t)
theta=Function('theta')(t)
theta0=exp(-t)
# Linear differential function
L=lambda f: f.diff(t)+f
# Homotopy equation
eqn1=Eq(L(theta)-L(theta0)+p*L(theta0)+\
p*epsilon*theta*theta.diff(t),0)
# Creating approximate solution and its substitution
eqn2=eqn1.subs(theta,y0+p*y1+p**2*y2)
# Expanding the above equation for p and derivatives
eqn3=expand(eqn2.doit())
# Segregating coefficients of p
de0=Eq(y0.diff(t)+y0,0); ics0={y0.subs(t,0):1};
de1=Eq(eqn3.lhs.coeff(p**1),0); ics1={y1.subs(t,0):0};
de2=Eq(eqn3.lhs.coeff(p**2),0); ics2={y2.subs(t,0):0};
```

Solution of coefficients:

`#for p^0`	
`dsolve(de0,ics=ics0)`	Out [171] : $y_0(t) = e^{-t}$
`# for p^1`	
`de1=de1.subs(y0,exp(-t))`	
`dsolve(de1,ics=ics1)`	Out [177] : $y_1(t) = (\in - \in e^{-t}) e^{-t}$
`# for p^2`	

`de2=de2.subs({y0:exp(-t),y1:`
`  epsilon*(1-exp(-t))*exp(-t)})`	

`dsolve(de2,ics=ics2)`	Out [185]: $y_2(t) = \left(\frac{\in^2}{2} - 2\in^2 e^{-t} + \frac{3\in^2 e^{-2t}}{2}\right) e^{-t}$

So, for $p = 1$ the approximate solution of Eq. (11) is:

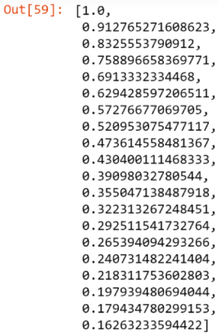
$$\theta(\mathrm{t}) = \mathrm{e}^{-\mathrm{t}} + (\in - \in \ \mathrm{e}^{-\mathrm{t}})\mathrm{e}^{-\mathrm{t}} + \left(\frac{\in^2}{2} - 2 \times \in^2 \mathrm{e}^{-\mathrm{t}} + \left(\frac{3}{2}\right) \times \in^2 \mathrm{e}^{-2\mathrm{t}}\right) \times \mathrm{e}^{-\mathrm{t}} \tag{15}$$

The exact analytical solution of the problem is [42]:

$$\ln(\theta) - \in (1 - \theta) = -t \tag{16}$$

At $\in =0$both the analytical and exact solution arrive at the same result i.e., e^{-t}.To solve the Eq. (16) again SymPy has been used. Considering $\in =0.1$the code and its result are as follows:

Code	Output

`from sympy import *`
`from sympy.abc import *`
 	
`t=0`
`g=[]`
 	
`while t<=2:`
`g.append(solve(Eq(log(theta)+0.1*theta,0.1-t),theta))`
`    t=t+2/20`
	 Out[59]: [1.0,
	0.912765271608623,
	0.8325553790912,
	0.758896658369771,
	0.6913332334468,
	0.629428597206511,
	0.57276677069705,
	0.520953075477117,
	0.473614558481367,
	0.430400111468333,
	0.39098032780544,
	0.355047138487918,
	0.322313267248451,
	0.292511541732764,
	0.265394094293266,
	0.240731482241404,
	0.218311753602803,
	0.197939480694044,
	0.179434780299153,
	0.16263233594422]

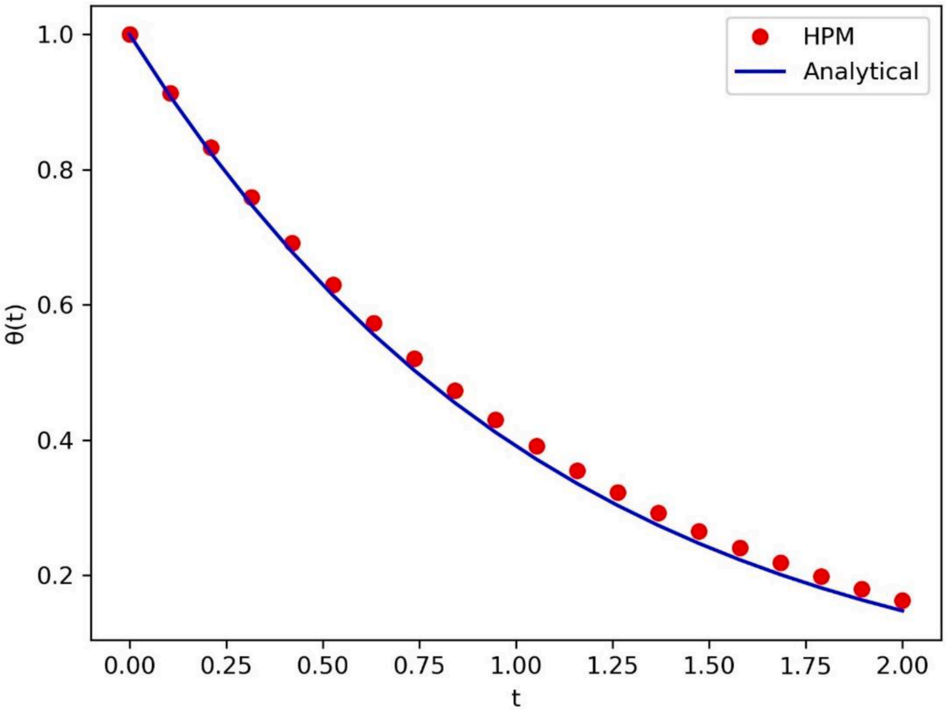


Fig. 4. Temperature variation for $\in =0.1$.

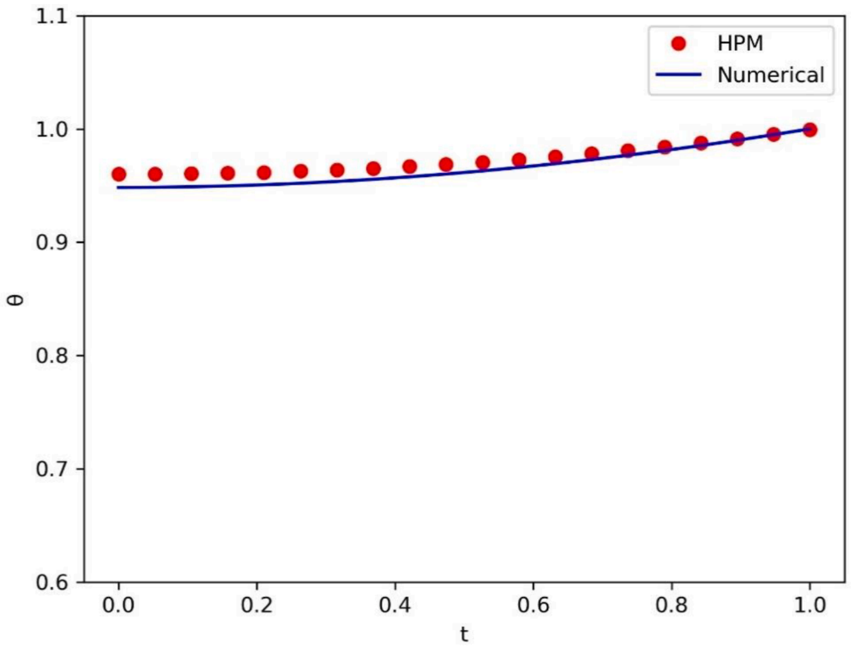


Fig. 5. Temperature variation from HPM and 4th order Runge Kutta.

Fig. 4 shows the variation of analytical solution and the solution obtained by HPM. As the time marches, the temperature of the lump reduces exponentially due to the assumption of infinite thermal conductivity of the lump. The physics of the problem has been captured well by the HPM as the results are very close to the analytical results.

4.3. Rectangular fin with radiation into ambient

Let us consider a fin which radiates into free space. The governing equation comments out to be highly nonlinear in nature [43] which is show in Eq. (17)

$$\frac{d^2\theta}{dx^2} - \in \theta^4 = 0 \tag{17}$$

Subject to the boundary condition that $\theta(1) = 1$ and $\frac{d\theta(0)}{dx} = 0$.

To develop homotopy of the above equation let us assume the linear operator (L) to be $\frac{\partial^2}{\partial x^2}$. So, the Homotopy equation becomes:

$$L(\theta) - L(\theta_0) + p \times L(\theta_0) - p \times \theta^4 \times \in = 0 \tag{18}$$

where θ_0 is taken as 1 and the final approximate solution can be given by $\theta = y_0 + py_1 + p^2y_2 + p^3y_3 \ldots$.

SymPy procedure to solve above homotopy equation is as follows:

```
from sympy import *
from sympy.abc import *
# Approximate solution coefficients
y0,y1,y2=symbols('y0,y1,y2')
y0=Function('y0')(x)
y1=Function('y1')(x)
y2=Function('y2')(x)
theta=Function('theta')(x)
theta0=1.0
# Linear differential function
L=lambda f: diff(f,x,x)
# Homotopy equation
eqn1=Eq(L(theta)-L(theta0)+p*L(theta0)-p*epsilon*theta**4,0)
# Creating approximate solution
theta_prime=y0+p*y1+p**2*y2
eqn2=eqn1.subs(theta,theta_prime)
# Expanding the above equation for p and derivatives
eqn3=expand(eqn2.doit())
```

(continued on next column)

(continued)

```
# Segregating coefficients of p
de0=Eq(y0.diff(x,x),0); ics0={y0.subs(x,1):1,y0.diff(x).subs
  (x,0):0};
de1=Eq(eqn3.lhs.coeff(p**1),0); ics1={y1.subs(x,1):0,y1.diff
  (x).subs(x,0):0};
de2=Eq(eqn3.lhs.coeff(p**2),0); ics2={y2.subs(x,1):0,y2.diff
  (x).subs(x,0):0};
```

Solution of coefficients:

```
#for p^0
dsolve(de0,ics=ics0)
```

Out [210]: $y_0(x) = 1$

```
# for p^1
de1=de1.subs(y0,1)
dsolve(de1,ics=ics1)
```

Out [211]: $y_1(x) = \frac{\in x^2}{2} - \frac{\in}{2}$

```
# for p^2
de2=de2.subs({y0:1,y1:epsilon*(1-
  x**2)/2})
dsolve(de2,ics=ics2)
```

Out [212]: $y_2(x) = \frac{\in^2 x^2 (6 - x^2)}{6} - \frac{5\in^2}{6}$

Therefore, the final output of approximate solution for p=1 is:

$$\theta = 1 + \frac{\in}{2}\left(x^2 - 1\right) + \frac{\in^2}{6}\left(x^2\left(6 - x^2\right) - 5\right) \tag{19}$$

As Eq. (17) is very difficult to solve analytically so it has been done with the help of scipy.integrate package in python which uses 4th order Runge Kutta. The results of HPM and numerical results are plotted in Fig. 5 for ∈ =0.09. It can be observed that as the time marches the temperature gradually increases this is due to the transfer of heat due to the radiation. The results are very encouraging as the HPM and numerical results are very close.

5. Conclusions

Based on the application of numerical analysis done in this research article the following conclusions can be drawn:

- HPM can be efficiently used to solve linear as well as non linear differential equations.
- HPM almost eliminates the use of complex numerical techiniques to solve the governing equations.
- HPM can solve approximate solutions to the problems whose analytical solution is very difficult.
- SymPy reduces the effort of manual derivations and with very less effort one can apply any scheme on it.
- A methodology has been shown to apply HPM in python via SymPy.

CRediT authorship contribution statement

Pankaj Dumka: Conceptualization, Methodology, Software, Investigation, Investigation, Writing – original draft. **Parth Singh Pawar:** Software, Writing – original draft. **Abhay Sauda:** Conceptualization, Methodology. **Gaurav Shukla:** Conceptualization, Methodology. **Dhananjay R. Mishra:** Conceptualization, Writing – review & editing.

Declaration of Competing Interest

The authors whose names are listed immediately below certify that they have No affiliations with or involvement in any organization or entity with any financial interest (such as honoraria; educational grants; participation in speakers' bureaus; membership, employment, consultancies, stock ownership, or other equity interest; and expert testimony or patent-licensing arrangements), or non-financial interest (such as personal or professional relationships, affiliations, knowledge or beliefs) in the subject matter or materials discussed in this manuscript.

References

[1] Hajmohammadi MR, Nourazar AH, Manesh Seyed Salman. Semi-analytical treatments of conjugate heat transfer. Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci. 2013;227(3):492–503.

[2] Hajmohammadi MR, Nourazar SS. On the solution of characteristic value problems arising in linear stability analysis; semi analytical approach. Appl. Math. Comput. 2014;239:126–32.

[3] Hajmohammadi MR, Nourazar SS. Conjugate forced convection heat transfer from a heated flat plate of finite thickness and temperature-dependent thermal conductivity. Heat Transf. Eng. 2014;35(9):863–74.

[4] Kumar D, Singh J, Kumar S, Sushila. Numerical computation of Klein-Gordon equations arising in quantum field theory by using homotopy analysis transform method. Alexandria Eng. J. 2014;53(2):469–74.

[5] Khan N, Hashmi MS, Iqbal S, Mahmood T. Optimal homotopy asymptotic method for solving Volterra integral equation of first kind. Alexandria Eng. J. 2014;53(3):751–5.

[6] Chakraverty S, Behera D. Dynamic responses of fractionally damped mechanical system using homotopy perturbation method. Alexandria Eng. J. 2013;52(3):557–62.

[7] Fakour M, Vahabzadeh A, Ganji DD. Scrutiny of mixed convection flow of a nanofluid in a vertical channel,". Case Stud. Therm. Eng. 2014;4:15–23.

[8] A. Aziz and T. Y. Na, "Perturbation methods in heat transfer," *Washington*, 1984.

[9] Sayevand K, Fardi M, Moradi E, Hemati Boroujeni F. Convergence analysis of homotopy perturbation method for Volterra integro-differential equations of fractional order. Alexandria Eng. J. 2013;52(4):807–12.

[10] Kumar S. A new fractional modeling arising in engineering sciences and its analytical approximate solution. Alexandria Eng. J. 2013;52(4):813–9.

[11] Pawar PS, Mishra DR, Dumka P. Solving first order ordinary differential equations using least square method : a comparative study. Int. J. Innov. Sci. Res. Technol. 2022;7(3):857–64.

[12] He JH. Homotopy perturbation technique. Comput. Methods Appl. Mech. Eng. 1999;178(3-4):257–62.

[13] He J-H. A coupling method of a homotopy technique and a perturbation technique for non-linear problems. Int. J. Non. Linear. Mech. 2000;35(1):37–43.

[14] He JH. Homotopy perturbation method for bifurcation of nonlinear problems. Int. J. Nonlinear Sci. Numer. Simul. 2005;6(2):207–8.

[15] Abbasbandy S. Iterated He's homotopy perturbation method for quadratic Riccati differential equation. Appl. Math. Comput. 2006;175(1):581–9.

[16] Biazar J, Ghazvini H, Eslami M. He's homotopy perturbation method for systems of integro-differential equations. Chaos, Solitons and Fractals 2009;39(3):1253–8.

[17] Obi CN. Semi-analytic solution of the nonlinear advection problem using homotopy perturbation method. Earthline J. Math. Sci. 2022;8(2):305–12.

[18] Ghane H, Hamed Z, Mashayekhy B, Mirebrahimi H. Topological homotopy groups. Bull. Belgian Math. Soc. - Simon Stevin 2008;15(3):455–64.

[19] M. Robnik-Šikonja and M. Bohanec, "Perturbation-Based Explanations of Prediction Models," in *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*, J. Zhou and F. Chen, Eds. Cham: Springer International Publishing, 2018, pp. 159-175.

[20] Abolvafaei M, Ganjefar S. Maximum power extraction from wind energy system using homotopy singular perturbation and fast terminal sliding mode method. Renew. Energy 2020;148:611–26.

[21] Loyinmi AC, Akinfe TK. Exact solutions to the family of Fisher's reaction-diffusion equation using Elzaki homotopy transformation perturbation method. Eng. Reports 2020;2(2):1–32.

[22] Huang X, Zhang Y, Moradi Z, Shafiei N. Computer simulation via a couple of homotopy perturbation methods and the generalized differential quadrature method for nonlinear vibration of functionally graded non-uniform micro-tube. Eng. Comput. 2021.

[23] Yu DN, He JH, Garcıa AG. Homotopy perturbation method with an auxiliary parameter for nonlinear oscillators. J. Low Freq. Noise Vib. Act. Control 2019;38(3-4):1540–54.

[24] He JH, El-Dib YO. Homotopy perturbation method with three expansions. J. Math. Chem. 2021;59(4):1139–50.

[25] Nadeem M, He J-H, Islam A. The homotopy perturbation method for fractional differential equations: part 1 Mohand transform. Int. J. Numer. Methods Heat Fluid Flow 2021;31(11):3490–504. Jan.

[26] Ghanbari Ashrafi T, Hoseinzadeh S, Sohani A, Shahverdian MH. Applying homotopy perturbation method to provide an analytical solution for Newtonian fluid flow on a porous flat plate. Math. Methods Appl. Sci. 2021;44(8):7017–30.

[27] Ganji DD, Ganji SS, Karimpour S, Ganji ZZ. Numerical study of homotopy-perturbation method applied to Burgers equation in fluid. Numer. Methods Partial Differ. Equ. 2010;26(4):917–30.

[28] Ames WF. Differential Equations. Courier Corporation; 2004.

[29] Huei YC. Benefits and introduction to python programming for freshmore students using inexpensive robots. In: Proceedings of IEEE International Conference on Teaching, Assessment and Learning for Engineering: Learning for the Future Now, TALE 2014; 2015. p. 12–7.

[30] Lin JWB. Why python is the next wave in earth sciences computing. Bull. Am. Meteorol. Soc. 2012;93(12):1823–4.

[31] Meurer A, et al. SymPy: Symbolic computing in python. PeerJ Comput. Sci. 2017;2017(1):1–27.

[32] Rocklin M, Terrel AR. Symbolic statistics with SymPy. Comput. Sci. Eng. 2012;14(3):88–93.

[33] Rocklin M. Uncertainty modeling with SymPy stats. In: *Proc. 11th Python Sci. Conf.*, no. Scipy; 2012. p. 51–5.

[34] C. Huang, "Python Solver for Stochastic Differential Equations," vol. 34, pp. 1-13, 2011.

[35] Sulzer V, Marquis SG, Timms R, Robinson M, Chapman SJ. Python battery mathematical modelling (PyBaMM). J. Open Res. Softw. 2021;9:1–8.

[36] Marowka A. On parallel software engineering education using python. Educ. Inf. Technol. 2018;23(1):357–72.

[37] Saarela M, Jauhiainen S. Comparison of feature importance measures as explanations for classification models. SN Appl. Sci. 2021;3(2):272.

[38] Pawar PS, Mishra DR, Dumka P, Pradesh M. Obtaining exact solutions of visco-incompressible parallel flows using python. Int. J. Eng. Appl. Sci. Technol. 2022;6(11):213–7.

[39] He JH. An elementary introduction to the homotopy perturbation method. Comput. Math. with Appl. 2009;57(3):410–2.

[40] Babolian E, Azizi A, Saeidian J. Some notes on using the homotopy perturbation method for solving time-dependent differential equations. Math. Comput. Model. 2009;50(1-2):213–24.

[41] Incropera FP, DeWitt DP, Bergman TL, Lavine AS, Incropera A, Dewitt FP, Bergman DP, Lavine T. Fundamentals of Heat and Mass Transfer. 8th ed. Hoboken, NJ: Wiley's; 2007.

[42] Ganji DD. The application of He's homotopy perturbation method to nonlinear equations arising in heat transfer. Phys. Lett. Sect. A Gen. At. Solid State Phys. 2006;355(4-5):337–41.

[43] Filobello-Nino U. The study of heat transfer phenomena using PM for approximate solution with dirichlet and mixed boundary conditions. Appl. Comput. Math. 2013;2(6):143.