International Journal of Mathematical Analysis and Modelling

(Formerly Journal of the Nigerian Society for Mathematical Biology)

Volume 7 Issue 2, 2024

ISSN (Print): 2682 - 5694
ISSN (Online): 2682 – 5708
www.tnsmb.org

Numerical solution approximation of nonlinear functional integral by Homotopy perturbation method

C.E. Chika*

Abstract

A numerical method for solving nonlinear functional integral is presented. Numerical examples are presented to show the validity of the numerical algorithm developed using Homotopy Perturbation Method (HPM). This method is simple to implement and gives good approximation at first iteration.

Keywords and phrases: Homotopy perturbation method; Nonlinear integral equations; Functional integral, Numerical Method.

1 Introduction

Finding solution to integral equations has been studied by scientists and engineers over the years; this is due to their immense applications in the field of sciences and engineering. These applications appear in the arrears of chaos, solitons, fractals, as well as scattering like the one encountered by Chika and Hooshyar [1].

Most of the methods developed work for specific assumptions, hence the need to develop more methods that include other cases that are not yet covered in the assumptions or the methods that may be simpler or easier to implement. Some of these methods solve linear functional integrals like the one presented by Abbasbandy [2] and [3] as well as other literatures. After the introduction of Homotopy Perturbation Method (HPM) by He in [4]; it has been used to solve many linear and nonlinear problems. This method has been applied also to linear functional integral equations and specific integral equations like Fredholm and Volterra [5, 6, 7, 8, 9] . In this work, the objective is to apply Homotopy Perturbation Method in solving a class of nonlinear functional integral.
In section 2, numerical algorithm for solving nonlinear functional integral is presented; in section 3, examples are presented to show the validity of the method. Finally, conclusion and further works are presented in section 4.

2 Numerical Algorithm

Functional integral can be written in two types: the Fredholm type as in Eq(1) and the Voltera type shown in Eq(2)

$$y(x) + w(x)y(h(x)) + \lambda \int_a^b K(x,t)(y(t))^m dt = g(x) \tag{1}$$

$$y(x) + w(x)y(h(x)) + \lambda \int_a^x K(x,t)(y(t))^m dt = g(x) \tag{2}$$

$x, t \in [a, b]$, $w(x)$, $y(h(x))$, $K(x,t)$ and $g(x)$ known

*E-mail: charles.chika@unn.edu.ng; Department of Mathematics, University of Nigeria, Nsukka, Enugu State, Nigeria

We form the homotopy equation using the Fredholm type

$$H(y,p) = (1-p)F(y) + pL(y) \tag{3}$$

where $p \in [0,1]$

$$L(y) = y(x) + w(x)y(h(x)) + \lambda \int_a^b K(x,t)(y(t))^m dt - g(x)$$

from Eq(3) $H(y,0) = F(y)$ and $H(y,1) = L(y)$, therefore, the solution to Eq(1) is traced when p increases from 0 to 1 by setting $H(y,p) = 0$.
Let $y(x) = \sum_{n=0}^{\infty} p^n y_n(x)$, and $F(y) = y(x) - y_0(x)$, Eq(3) becomes;

$$(1-p)\left(\sum_{n=0}^{\infty} p^n y_n(x) - y_0(x)\right) + p\left[\sum_{n=0}^{\infty} p^n y_n(x) + w(x)q(x) + \lambda \int_a^b K(x,t)(\sum_{n=0}^{\infty} p^n y_n(x))^m dt - g(x)\right] = 0 \tag{4}$$

where $q(x) = y(h(x))$
Expanding and equating like powers of p in Eq(4), we obtain
$p^0 : y_0(x) - y_0(x) = 0$
$p^1 : y_1(x) + y_0(x) + w(x)q(x) - g(x) - \lambda \int_a^b K(x,t)(y_0(t))^m dt = 0$

$$p^2 : \begin{cases} y_2(x) + \lambda \int_a^b K(x,t)(2y_0(t)y_1(t))dt = 0, & \text{if } m = 2 \\ y_2(x) + \lambda \int_a^b K(x,t)(3((y_0(t))^2 y_1(t))dt = 0, & \text{if } m = 3 \\ y_2(x) + \lambda \int_a^b K(x,t)(4(y_0(t))^3 y_1(t))dt = 0, & \text{if } m = 4 \\ y_2(x) + \lambda \int_a^b K(x,t)(5(y_0(t))^4 y_1(t))dt = 0, & \text{if } m = 5 \\ . \\ . \\ . \end{cases}$$

.
.
.

It follows that

$$p^n : \begin{cases} y_n(x) + \lambda \int_a^b K(x,t) \sum_{k=0}^{n-1} (y_k(t) y_{n-k-1}(t))dt = 0, & \text{if } m = 2 \\ y_n(x) + \lambda \int_a^b K(x,t) \sum_{i=0}^{n-1} \sum_{k=0}^{n-i-1} (y_i(t) y_k(t) y_{n-k-i-1}(t))dt = 0, & \text{if } m = 3 \\ y_n(x) + \lambda \int_a^b K(x,t) \sum_{i=0}^{n-1} \sum_{k=0}^{n-i-1} \sum_{l=0}^{n-i-k-1} (y_i(t) y_k(t) y_l(t) y_{n-l-k-i-1}(t))dt = 0, & \text{if } m = 4 \\ . \\ . \\ . \end{cases}$$

.
.
.

Therefore, the algorithm for Fredholm type can be written as follows:
$y_0(x) = g(x) - w(x)y(h(x))$, since $q(x) = y(h(x))$
for $n > 0$

$$y_n(x) = \begin{cases} -\lambda \int_a^b K(x,t) \sum_{k=0}^{n-1} (y_k(t) y_{n-k-1}(t))dt, & \text{if } m = 2 \\ -\lambda \int_a^b K(x,t) \sum_{i=0}^{n-1} \sum_{k=0}^{n-i-1} (y_i(t) y_k(t) y_{n-k-i-1}(t))dt, & \text{if } m = 3 \\ -\lambda \int_a^b K(x,t) \sum_{i=0}^{n-1} \sum_{k=0}^{n-i-1} \sum_{l=0}^{n-i-k-1} (y_i(t) y_k(x) u_l(t) y_{n-l-k-i-1}(t))dt, & \text{if } m = 4 \\ . \\ . \\ . \end{cases}$$

.
.
.

Similarly, the algorithm for Volterra type is

$y_0(x) = g(x) - w(x)y(h(x))$, since $q(x) = y(h(x))$
for $n > 0$

$$y_n(x) = \begin{cases} -\lambda \int_a^x K(x,t) \sum_{k=0}^{n-1} (y_k(t) y_{n-k-1}(t))dt, & \text{if } m = 2 \\ -\lambda \int_a^x K(x,t) \sum_{i=0}^{n-1} \sum_{k=0}^{n-i-1} (y_i(t) y_k(t) y_{n-k-i-1}(t))dt, & \text{if } m = 3 \\ -\lambda \int_a^x K(x,t) \sum_{i=0}^{n-1} \sum_{k=0}^{n-i-1} \sum_{l=0}^{n-i-k-1} (y_i(t) y_k(x) u_l(t) y_{n-l-k-i-1}(t))dt, & \text{if } m = 4 \\ . \\ . \\ . \end{cases}$$

.
.
.

Since, the solution is traced as p increases from 0 to $1 \Rightarrow y(x) = \lim_{p \to 1^-} \sum_{n=1}^{\infty} p^n y_n(x)$, so we take $y(x) = \sum_{n=1}^{\infty} y_n(x)$

3 Examples

Here we present examples to validate the algorithms stated above.

Example 1: Solve for $y(x)$ in the equation

$$y(x) + e^{-x}(1 - \cos^2 x) + \frac{1}{10} \int_0^{\frac{1}{2}} xt(y(t))^2 = (1 - x^2) + e^{-x}(1 - \cos^2 x) + \frac{37}{3840} x$$

Solution:
The equation in example 1 is of Fredholm type with $w(x) = e^{-x}$, $y(h(x)) = 1 - \cos^2 x$ and $g(x) = (1 - x^2) + e^{-x}(1 - \cos^2 x) + \frac{37}{3840} x$
Therefore,
$y_0(x) = (1 - x^2) + \frac{37}{3840} x$
for $n > 1$

$$y_n(x) = -\frac{1}{10} \int_0^{\frac{1}{2}} xt \sum_{k=0}^{n-1} (y_k(t) y_{n-k-1}(t))dt$$

The exact solution is $1 - x^2$
Using Matlab to compute and compare the solutions show that the exact and approximate solution have good agreement by just taking the first partial sum for the approximate solution. This can be seen in Figure 1 and Table 1.

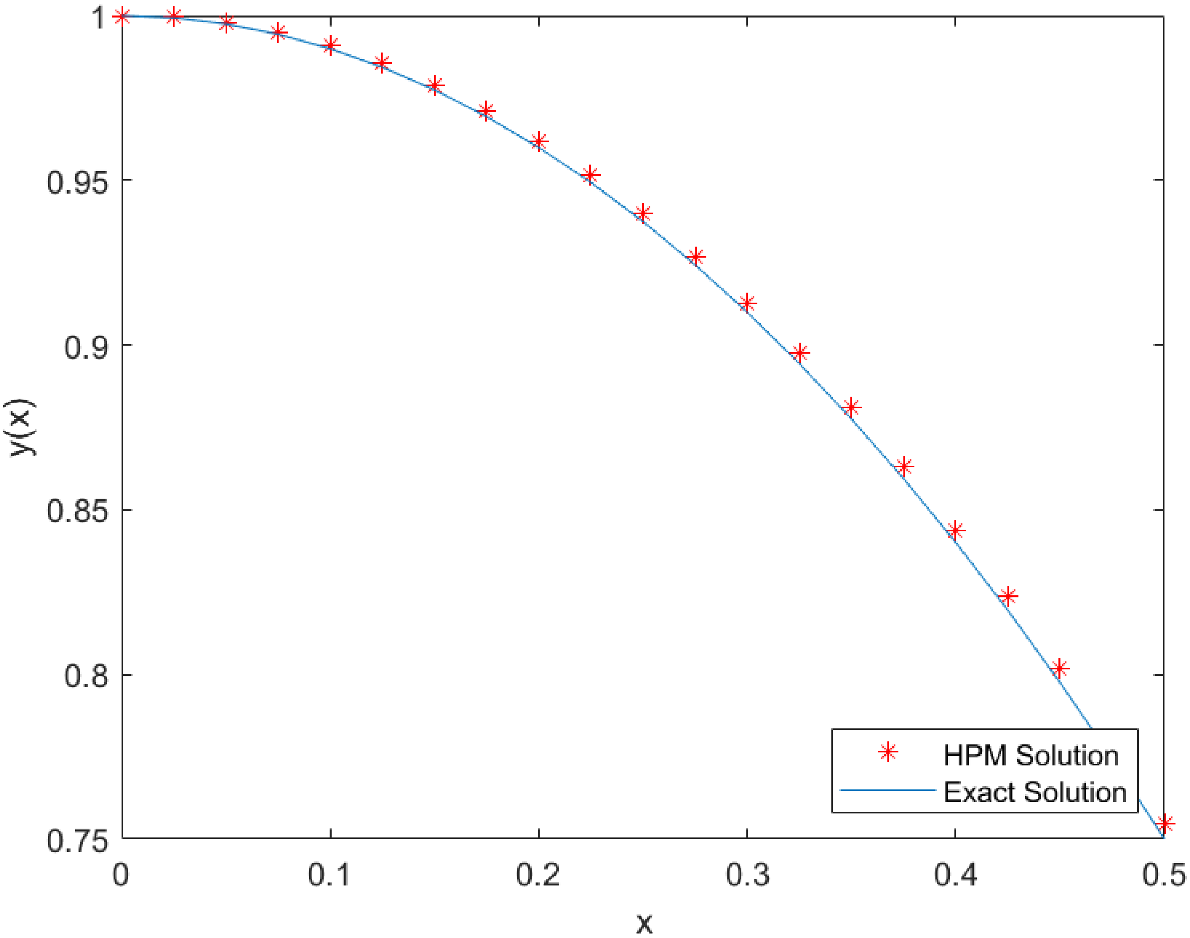

Figure 1: The graph of exact and HPM solution in Example 1

Table 1: Numerical values for example 1

x_i	Approximate $y(x_i)$	Exact $y(x_i)$	Relative error (%)
0	1.0000	1.0000	0
0.0500	0.9980	0.9975	0.0483
0.1000	0.9910	0.9900	0.0973
0.1500	0.9789	0.9775	0.1479
0.2000	0.9619	0.9600	0.2007
0.2500	0.9399	0.9375	0.2569
0.3000	0.9129	0.9100	0.3177
0.3500	0.8809	0.8775	0.3843
0.4000	0.8439	0.8400	0.4588
0.4500	0.8018	0.7975	0.5437
0.5000	0.7548	0.7500	0.6424

Example 2: Solve for y(x) such that
$y(x) + xe^{-3x} + \frac{1}{100}\int_0^x x(y(t))^3 dt = x(e^{-3x} + \frac{1}{300}) + e^x(1 - \frac{1}{300}e^{2x}), x \in [0, \frac{1}{2}]$

Solution: The equation in example 2 is of Volterra type with $w(x) = x$, $y(h(x)) = e^{-3x}$ and

$g(x) = x(e^{-3x} + \frac{1}{300}) + e^{x}(1 - \frac{1}{300}e^{2x})$
$y_0 = x(e^{-3x} + \frac{1}{300}) + e^{x}(1 - \frac{1}{300}e^{2x}) - xe^{-3x}$
for $n > 0$
$y_n = -\frac{1}{100}\int_0^x x\sum_{i=0}^{n-1}\sum_{k=0}^{n-i-1}(y_i(t)y_k(t)y_{n-k-i-1}(t))dt$
The exact solution to this problem is $y(x) = e^{x}$.
Figure 2 shows the exact and approximate solution plotted and Table 2 displays their errors for different values of x, as can be seen, there is a good agreement between the exact and approximate solution.

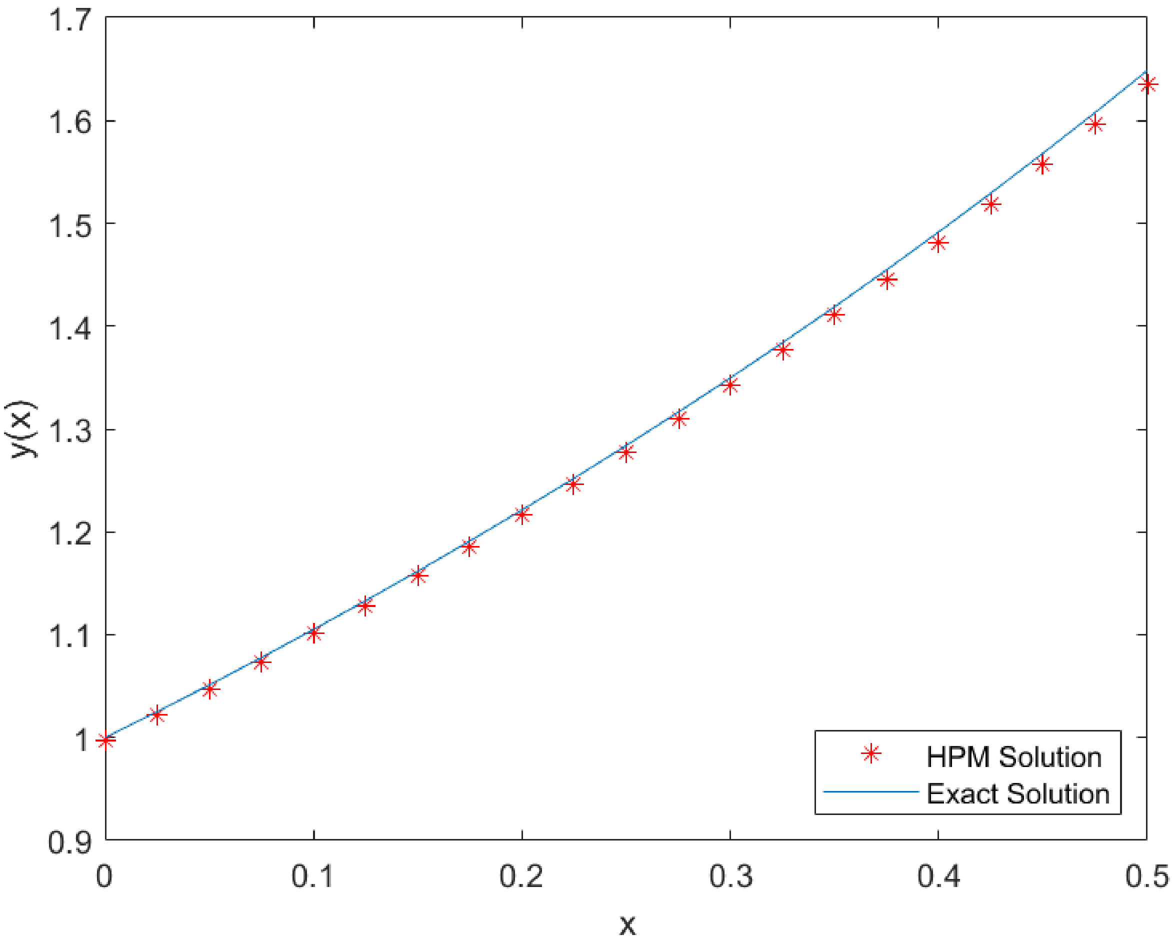

Figure 2: The graph of exact and HPM solutions in Example 2

Table 2: Numerical values for example 2

x_i	Approximate $y(x_i)$	Exact $y(x_i)$	Relative error (%)
0	0.9967	1.0000	0.3333
0.0500	1.0476	1.0513	0.3525
0.1000	1.1010	1.1052	0.3770
0.1500	1.1571	1.1618	0.4069
0.2000	1.2160	1.2214	0.4427
0.2500	1.2778	1.2840	0.4847
0.3000	1.3427	1.3499	0.5333
0.3500	1.4107	1.4191	0.5890
0.4000	1.4821	1.4918	0.6525
0.4500	1.5570	1.5683	0.7242
0.5000	1.6354	1.6487	0.8050

4 Conclusion

The numerical method works well according to the shown examples, only first partial sum is needed to get good agreement with the exact result. The maximum relative error for example 1 and 2 are 0.64% and 0.805% respectively; this further confirms how good the numerical solution approximates exact solution. Since this gives good agreement, we intend carrying out further analysis on the method to explain some of its numerical behavior and convergence.

References

[1] Chika CE, Hooshyar MA. , "An electromagnetic inverse scattering problem for dielectrics that depend on two spatial variables via Eikonal approximation," Microw Opt Technol Lett., 2019;1–7. https://doi.org/10.1002/mop.31752

[2] S. Abbasbandy, "Application of He's homotopy perturbation method to functional integral equations," Chaos, Solitons and Fractals, 31 (2007) 1243-1247.

[3] Abdul-Majid Wazwaz , " *Linear and Nonlinear Integral Equations,*" Higher Education Press; Springer, Beijing (2011).

[4] J.H. He, "Homotopy perturbation Technique," Computer Methods in Applied Mechanics and Engineering, 178 (1999) 257-262.

[5] J. Biazar, H. Ghazvini , "Numerical solution for special non-linear Fredholm integral equation by HPM," Appl. Math. Comput. 195 (2008) 681-687.

[6] Edyta Hetmaniok, Damian Slota, Roman Witula , "Convergence and error estimation of homotopy perturbation method for Fredholm and Volterra integral equations," Appl. Math. Comput. 218 (2012) 10717-10725.

[7] Behzad Ghanbari, "On the Convergence of the Homotopy Analysis Method for Solving Fredholm Integral Equations," Walailak J Sci & Tech, 2013; 10(4):395–403.

[8] S. Abbasbandy, "Numerical solutions of the integral equations: Homotopy perturbation method and Adomain's decomposition method," Applied Mathematics and Computation, 173 (2006) 493-500.

[9] Charles Ekene Chika, Chinedu Junior Okere, "Iterative Solution and Convergence of Nonlinear Volterra Integral Equations of The Second Second Kind Using HPM," International Journal of Engineering Research and Technology (IJERT) Volume 09, Issue 09 (2020).