International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 09, September-2020

Iterative Solution and Convergence of Nonlinear Volterra Integral Equations of The Second Kind using HPM

Charles Ekene Chika
Department of Mathematical Sciences,
University of Nigeria, Nsukka, Nigeria.

Chinedu Junior Okere
China University of Petroleum,
Beijing, China.

***Abstract*:- An iterative method is developed to solve a class of nonlinear Volterra integral equations. This method uses the concept of homotopy perturbation to approximate the exact solution of the integral equation. The convergence is discussed and illustrated with examples. Examples are used to show the validity of the algorithm and theorems presented which shows that the method is effective and easy to implement.**

Keywords: Homotopy perturbation method; Nonlinear Volterra integral; Convergence, Numerical Method

Subject Class: Numerical Method

1 INTRODUCTION

Solution to integral equations is of high interest in science since these equations appear in many applied problems; which in many cases takes the form of the so called Fredholm or Volterra Integral Equations. This equations result from direct model or from solving another problem like the one encountered by Chika and Hooshyar [5]. Di_erent methods have been developed to solve these integral equations, some which are presented in [1]. These methods have some assumptions like requiring separable kernel, linearity etc. hence, there is need to develop more methods that will solve di_erent classes of these equations.

Homotopy Perturbation Method (HPM) has been good in solving nonlinear equations ever since it was introduced by He [7], and this has been applied to solve linear integral equations like the one by Abbasbandy [8]. The convergence of this method has been studied as well like the ones presented in [4] and [6]. J. Biazar and H. Ghazvini [2] used this method to solve nonlinear Fredholm integral which worked well. The aim of this work is to apply this method in nonlinear Volterra integral and study it's convergence.

Numerical (or iterative) algorithm developed using homotopy perturbation as well as examples showing the validity of the algorithm is presented in section 2. Section 3 discusses the convergence and section 4 has the conclusion and future works.

2 Numerical Algorithm and Examples

Let's consider the nonlinear Volterra Integral Equation:

$$u(x) = f(x) + \lambda \int_a^x K(x,t)(u(t))^m dt \quad (1)$$

x ∈ [a; b] and m is a positive integer. This type of equations occurs in applied problems like those arising from solitons, fractals, etc.

To get the solution to Eq(1), consider the expansion $u(x) = \sum_{n=1}^{\infty} P^n u_n(x)$ where p ∈ [0; 1] such that the solution u(x) to Eq(1) is u(x) $= \lim_{P\to 1^-} \sum_{n=1}^{\infty} P^n u_n(x)$

Using the homotopy equation

$$H(u,p) = (1-p)(u(x) - u_0) + p(u(x) - f(x) - \lambda \int_a^x K(x,t)(u(t))^m dt) = 0 \quad (2)$$

Substituting $u(x) = \sum_{n=1}^{\infty} P^n u_n(x)$ in Eq(2) we have

$$H(u,p) = (1-p)\left(\sum_{n=1}^{\infty} p^n u_n(x) - u_0(x)\right) + p\left(\sum_{n=1}^{\infty} p^n u_n(x) - f(x) - \lambda \int_a^x K(x,t)(\sum_{n=1}^{\infty} p^n (u_n(t))^m dt\right) = 0 \quad (3)$$

Expanding the series in Eq(3) and equating like powers of p gives

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 09, September-2020

$$p^0 : u_0(x) - u_0(x) = 0$$
$$p^1 : u_1(x) + u_0(x) - f(x) - \lambda \int_a^x K(x,t)(u_0(t))^m dt = 0$$

$$p^2 : \begin{cases} u_2(x) - \lambda \int_a^x K(x,t)(2u_0(t)u_1(t))dt = 0, & \text{if } m = 2 \\ u_2(x) - \lambda \int_a^x K(x,t)(3((u_0(t))^2 u_1(t))dt = 0, & \text{if } m = 3 \\ u_2(x) - \lambda \int_a^x K(x,t)(4(u_0(t))^3 u_1(t))dt = 0, & \text{if } m = 4 \\ u_2(x) - \lambda \int_a^x K(x,t)(5(u_0(t))^4 u_1(t))dt = 0, & \text{if } m = 5 \\ . \\ . \\ . \end{cases}$$

.
.
.

It follows that

$$p^n : \begin{cases} u_n(x) - \lambda \int_a^x K(x,t) \sum_{k=0}^{n-1} (u_k(t)u_{n-k-1}(t))dt = 0, & \text{if } m = 2 \\ u_n(x) - \lambda \int_a^x K(x,t) \sum_{i=0}^{n-1} \sum_{k=0}^{n-i-1} (u_i(t)u_k(t)u_{n-k-i-1}(t))dt = 0, & \text{if } m = 3 \\ u_n(t) - \lambda \int_a^x K(x,t) \sum_{i=0}^{n-1} \sum_{k=0}^{n-i-1} \sum_{l=0}^{n-i-k-1} (u_i(t)u_k(t)u_l(t)u_{n-l-k-i-1}(t))dt = 0, & \text{if } m = 4 \\ . \\ . \\ . \end{cases}$$

.
.
.

Therefore, the numerical algorithm for solving Eq(1) can be written as, $u_o(x) = f(x)$
for n > 0

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

.
.
.

Where

$$u(x) = \sum_{n=1}^{\infty} u_n(x) \qquad (4)$$

In practice, we take u(x) $\approx \tilde{u}(x)$

$$\tilde{u}(x) = \sum_{n=1}^{N} u_n(x) \qquad (5)$$

is the so called partial sum.
To check the feasibility of the algorithm, we look at some examples:
Example 1: Find the function u(x) such that u(x) $= e^x + \frac{1}{300}x(1 - e^{3x}) + \frac{1}{100}\int_0^x x\, u^3(t)dt,\ x \in \left[0, \frac{1}{2}\right]$ is satisfied.
Solution:
Using the algorithm

$$u_0(x) = f(x) = e^x + \frac{1}{300}x(1 - e^{3x})$$

for n > 1

$$u_n(x) = \frac{1}{100} \int_0^x x \sum_{i=0}^{n-1} \sum_{k=0}^{n-i-1} (u_i(t)u_k(t)u_{n-k-i-1}(t))dt$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 09, September-2020

We use MATLAB to compute the above quantity for di_erent values of x and N = 4. The exact solution to this problem is u(x) = ex. As can be seen in Figure 1 and Table 1, there is good agreement between the exact solution and the numerical solution.

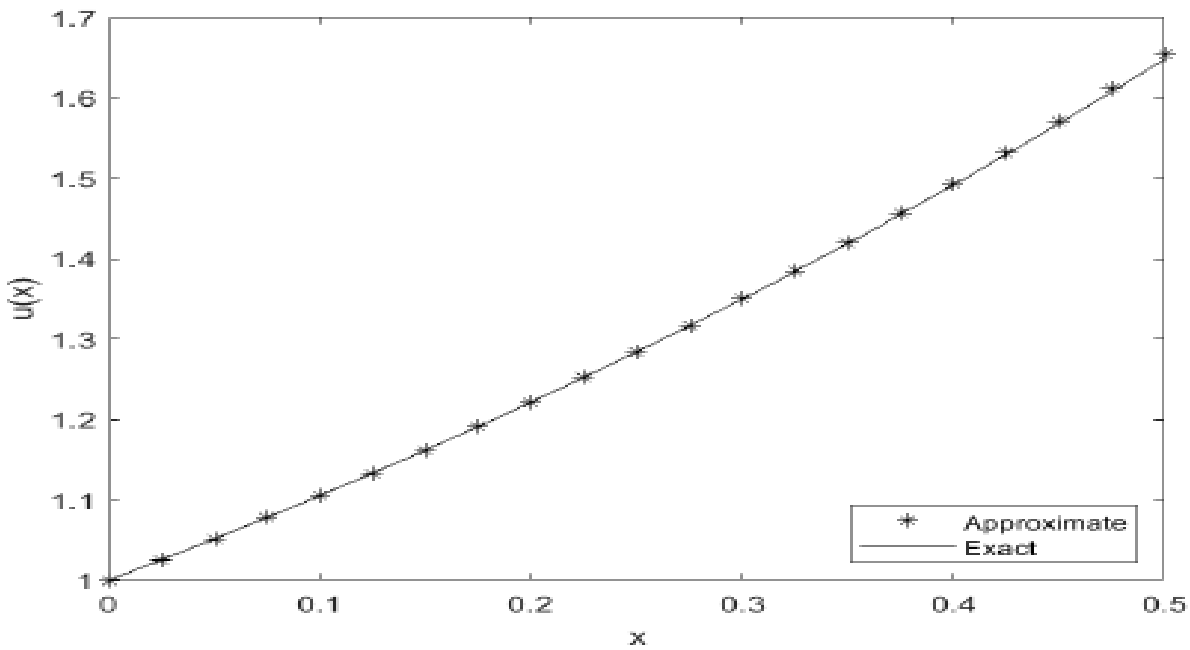

Figure 1: The graph of exact and approximate solution by HPM in Example 1

Example 2: Find u(x) that satisfied u(x) $= 1 + \frac{1}{160}x^2 - \frac{1}{480}x^4 - \frac{1}{2400}x^6 + \frac{1}{80}\int_0^x (x-t)\, u^2(t)dt$ u(x). The exact solution to this equation is u(x) = 1 + x^2. Applying the algorithm, we have; u_o(x) $= 1 + \frac{1}{160}x^2 - \frac{1}{480}x^4 - \frac{1}{2400}x^6$
for n > 1

$$u_n(x) = \frac{1}{80}\int_0^x (x-t)\sum_{k=0}^{n-1}(u_k(t)$$

Looking at Figure 2 and Table 2, the exact solution and the numerical solution are in good agreement.
Example 3: Consider the equation $u(x) = \cos(x) + \frac{1}{160}\cos(2x) - \frac{1}{80}x^2 - \frac{1}{160} + \frac{1}{20}\int_0^x (x-t)u^2\,(t)dt$, x ∈ [0; 1] whose solution is u(x) = cos (x).

Table 1: Numerical output for example 1

x_i	Exact $u(x_i)$	Approximate $u(x_i)$	Relative error (%)
0	1.0000	1.0000	0
0.0500	1.0513	1.0513	0.0002
0.1000	1.1052	1.1052	0.0017
0.1500	1.1618	1.1619	0.0059
0.2000	1.2214	1.2216	0.0149
0.2500	1.2840	1.2844	0.0308
0.3000	1.3499	1.3506	0.0564
0.3500	1.4191	1.4204	0.0953
0.4000	1.4918	1.4941	0.1516
0.4500	1.5683	1.5719	0.2308
0.5000	1.6487	1.6543	0.3394

Table 2: Numerical output for example 2

x_i	Exact $u(x_i)$	Approximate $u(x_i)$	Relative error (%)
0	1.0000	0.9938	0.6250
0.1000	1.0100	1.0038	0.6139
0.2000	1.0400	1.0340	0.5809
0.3000	1.0900	1.0843	0.5269
0.4000	1.1600	1.1548	0.4523
0.5000	1.2500	1.2455	0.3569
0.6000	1.3600	1.3568	0.2385
0.7000	1.4900	1.4886	0.0929
0.8000	1.6400	1.6414	0.0863
0.9000	1.8100	1.8156	0.3084
1.0000	2.0000	2.0117	0.5855

Applying the algorithm

$$u_0(x) = \cos(x) + \frac{1}{160}\cos(2x) - \frac{1}{80}x^2 - \frac{1}{160}$$

for $n > 0$

$$u_n(x) = \frac{1}{20}\int_0^x (x-t)\sum_{k=0}^{n-1}(u_k(t)u_{n-k-1}(t))dt$$

Figure 3 and Table 3 shows the comparison of the exact and the numerical solutions. It can be seen that the numerical approximation performed well.

3 Convergence and Error Estimation

In this section, we discuss the convergence of the algorithm presented above. We are going to use Abel's Theorem [4], i.e if $v = \sum_{n=0}^{\infty} P^n v_n$ has radius of

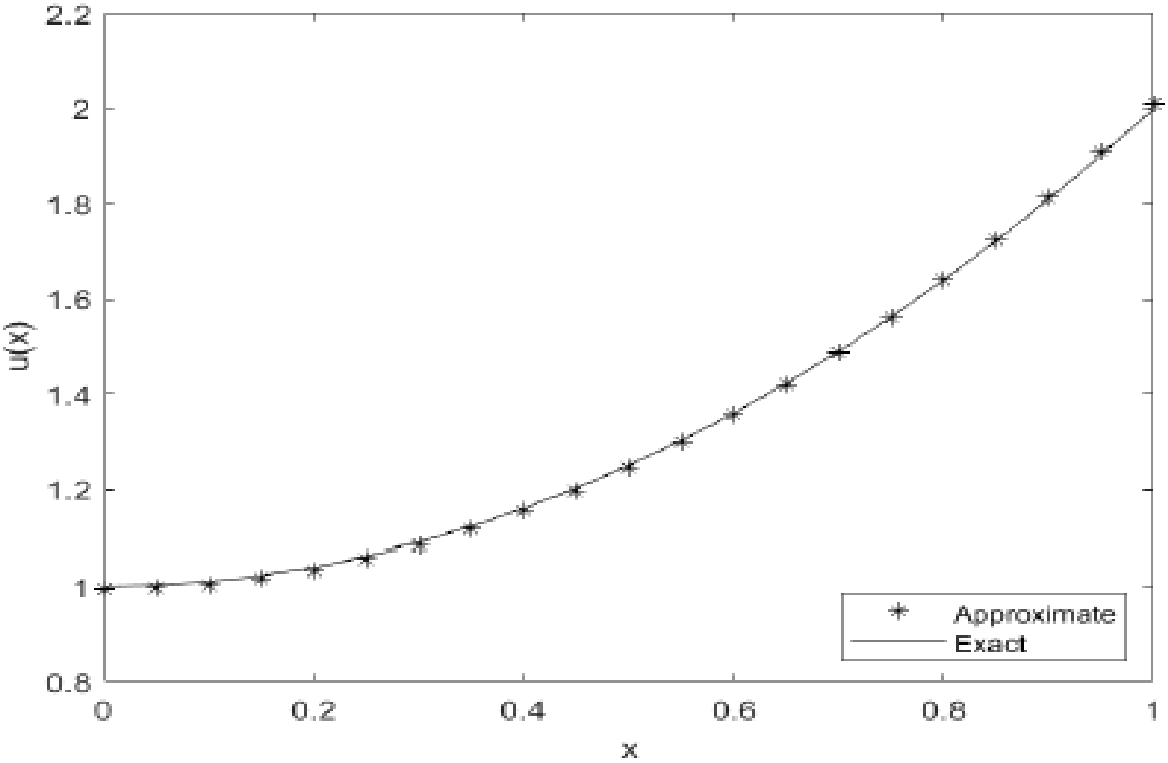

Figure 2: The graph of exact and approximate solution by HPM in Example 2

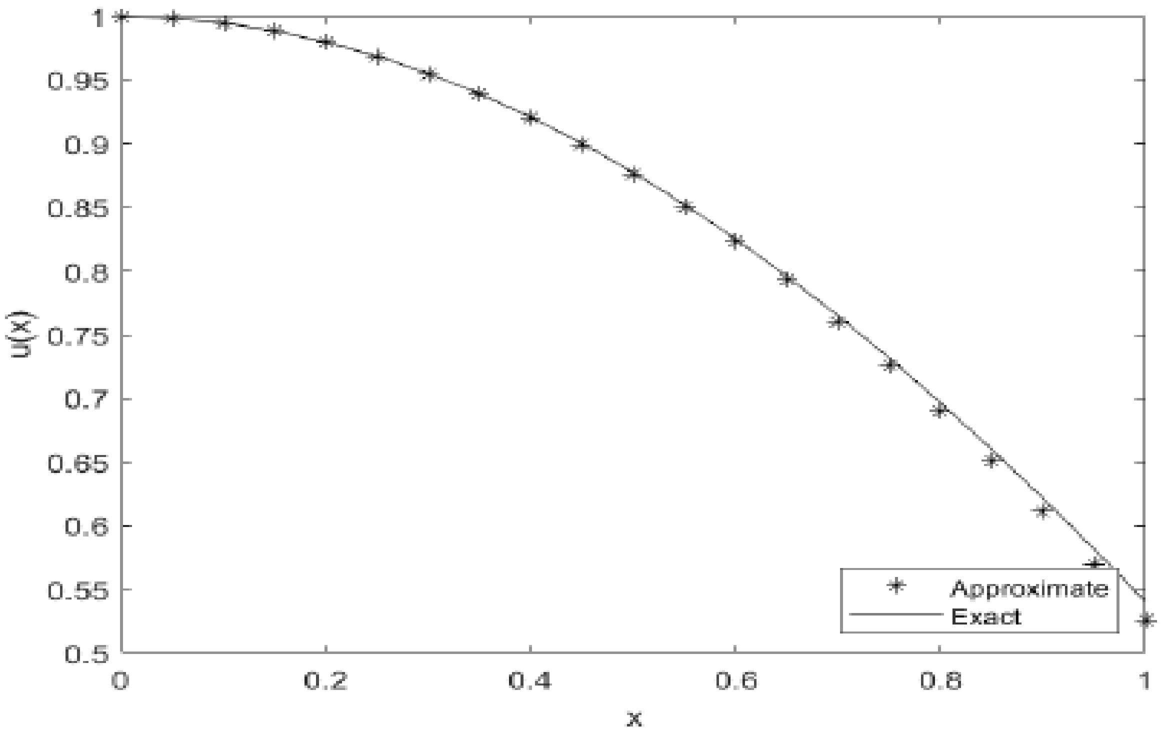

Figure 3: The graph of exact and approximate solution by HPM in Example 3

Table 3: Numerical output for example 3

x_i	Exact $u(x_i)$	Approximate $u(x_i)$	Relative error (%)
0	1.0000	1.0000	0
0.1000	0.9950	0.9950	0.0004
0.2000	0.9801	0.9800	0.0041
0.3000	0.9553	0.9552	0.0184
0.4000	0.9211	0.9205	0.0559
0.5000	0.8776	0.8764	0.1358
0.6000	0.8253	0.8230	0.2856
0.7000	0.7648	0.7607	0.5435
0.8000	0.6967	0.6900	0.9636
0.9000	0.6216	0.6115	1.6246
1.0000	0.5403	0.5260	2.6483

Convergence not less than one and the series $\sum_{n=0}^{\infty} v_n$ is absolutely convergent, then,

$$u(x) = \lim_{P\to 1^-} v = \sum_{n=0}^{\infty} P^n v_n$$

For this first work, we take m=2 to investigate the convergence.
Theorem 1: for m = 2, let K (x, t) and f(x) be continuous in the regions Ω= [a; b] × [a; b] and Ω = [a; b] respectively. K (x, t) and f(x) is bounded such that $|k\ (x,t)| \leq N_k$ and $|f(x)| \leq N_o$ for all x, t ∈ [a, b] and C = $N_k N_0$. If $|\lambda| < \frac{1}{C(b-a)}$, then the algorithm above is uniformly convergent in [a, b] for each p ∈ [0, 1].

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 09, September-2020

Proof for m=2

$$|f(x)| \le N_0 \Rightarrow |u_0(x)| \le N_0,\ |K(x,t)| \le N_k$$
$$|u_1(x)| = |\lambda \int_a^x K(x,t)(u_0(t))^m dt| \le |\lambda| \int_a^x |K(x,t)||(u_0(t))|^m dt \le |\lambda| N_k N_o^m (x-a)$$
$$|u_2(x)| = |\lambda \int_a^x K(x,t)(2u_0(t)u_1(t))dt| \le |\lambda| \int_a^x |K(x,t)|2|(u_0(t))||(u_1(t))|dt \le 2|\lambda|^2 N_k^2 N_o^{m+1} \frac{(x-a)^2}{2}$$
$$|u_3(x)| \le |\lambda| \int_a^x |K(x,t)|(2|(u_0(t))||(u_2(t))| + |(u_1(t))|^2 dt \le 6|\lambda|^3 N_k^3 N_o^{m+2} \frac{(x-a)^3}{2.3}$$
.
.
.
$$|u_n(x)| \le n!|\lambda|^n N_k^n N_o^{m+n-1} \frac{(x-a)^n}{n!} \text{ this implies}$$

$$|u_n(x)| \le |\lambda|^n N_k^n N_o^{m+n-1}(x-a)^n \tag{6}$$

Using Eq(6), for p ∈ [0, 1]

$$\sum_{n=0}^{\infty} p^n u_n(x) \le |\sum_{n=0}^{\infty} p^n u_n(x)| \le \sum_{n=0}^{\infty} |u_n(x)| \le \sum_{n=0}^{\infty} |\lambda|^n N_k^n N_o^{m+n-1}(x-a)^n \le N_0 \sum_{n=0}^{\infty} (|\lambda| N_k N_o (x-a))^n$$

Since m=2
This is geometric series, for convergence;

$$|\lambda| N_k N_o (x-a)| < 1 \Rightarrow |\lambda| < \frac{1}{N_k N_0 (x-a)}$$

Therefore,

$$|\lambda| < \frac{1}{C(x-a)} \tag{7}$$

or more strongly

$$|\lambda| < \frac{1}{C(b-a)} \tag{8}$$

Theorem 2: The Nth order approximation error of the solution to Eq(1) $E_n(x) = N_o \frac{(|\lambda| C(x-a))^{N+1}}{1-|\lambda|\, C(x-a)}$ for all x ∈ [a, b] or more strongly $E_n = N_o \frac{(C|\lambda|(b-a))^{N+1}}{1-C|\lambda|(b-a)}$

Where

$$E_n = \sup |u(x) - \tilde{u}(x)|,\ \tilde{u}(x) = \sum_{n=0}^{N} u_n(x) \text{ and } u(x) = \sum_{n=0}^{\infty} u_n(x)$$

Note: This estimates the error between partial sum and infinite sum using HPM (not exact solution).

Proof
Using Eq(6) and m = 2

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 9 Issue 09, September-2020

$$|u(x)-\tilde{u}(x)| = |\sum_{n=0}^{\infty} u_n(x) - \sum_{n=0}^{N} u_n(x)| = |\sum_{n=N+1}^{\infty} u_n(x)|$$
$$\leq \sum_{n=N+1}^{\infty} |u_n(x)| \leq N_0 \sum_{n=N+1}^{\infty} |\lambda|^n N_k^n N_o^n (x-a)^n$$
$$= N_0 \frac{(C|\lambda|(b-a))^{N+1}}{1-C|\lambda|(x-a)}$$
$$\leq N_0 \frac{(C|\lambda|(b-a))^{N+1}}{1-C|\lambda|(b-a)} \qquad (9)$$

From Theorem 1, if other quantities are constant; smaller λ, gives better convergence and from Theorem 2 reduces the error; this is illustrated in the next example.

Example 4: Using equation in example 3 and changing λ from $\frac{1}{20}$ to $\frac{1}{100}$, the equation becomes

$u(x) = \cos(x) + \frac{1}{800}\cos(2x) - \frac{1}{400}x^2 - \frac{1}{800} + \frac{1}{100}\int_0^x (x-t)u^2(t)dt,\ x \in [0,1]$

Notice: $K(x,t) = |x-t| \leq 1 = N_k$

$|f(x)| \leq |\cos x| + \frac{1}{800}|\cos 2x| + |\frac{1}{400}x^2 + \frac{1}{800}| \leq 1.005 = N_0$

$C = N_k N_0 = 1.005$

This implies that $|\lambda| < \frac{1}{C(b-a)}$, therefore, the algorithm converges.

Comparing values in Table 3 and Table 4 as well as Figure 3 and Figure 4, HPM gives a better approximation in example 4 than example 3, which confirms the validity of the Theorems.

Table 4: Numerical output for example 4

x_i	Exact $u(x_i)$	Approximate $u(x)$	Relative Error (%)
0	1.0000	1.0000	0
0.1000	0.9950	0.9950	0.0001
0.2000	0.9801	0.9801	0.0009
0.3000	0.9553	0.9553	0.0038
0.4000	0.9211	0.9210	0.0116
0.5000	0.8776	0.8773	0.0279
0.6000	0.8253	0.8249	0.0583
0.7000	0.7648	0.7640	0.1100
0.8000	0.6967	0.6954	0.1935
0.9000	0.6216	0.6196	0.3237
1.0000	0.5403	0.5375	0.5242

4 Conclusion

The algorithm developed using homotopy perturbation method performs well as shown in the examples, the percentage error in the shown examples are all less than 0.65% except in example 3 which has up to 2.65%. This is explained by the convergence theorems. Since 4th partial sum is used in all the computations, better approximation can be gotten by adding more terms. This algorithm is easy to implement and gives good approximation. Given the impressive performance, future communication will involve more analysis and extending the method to more general class of nonlinear Volterra integral.

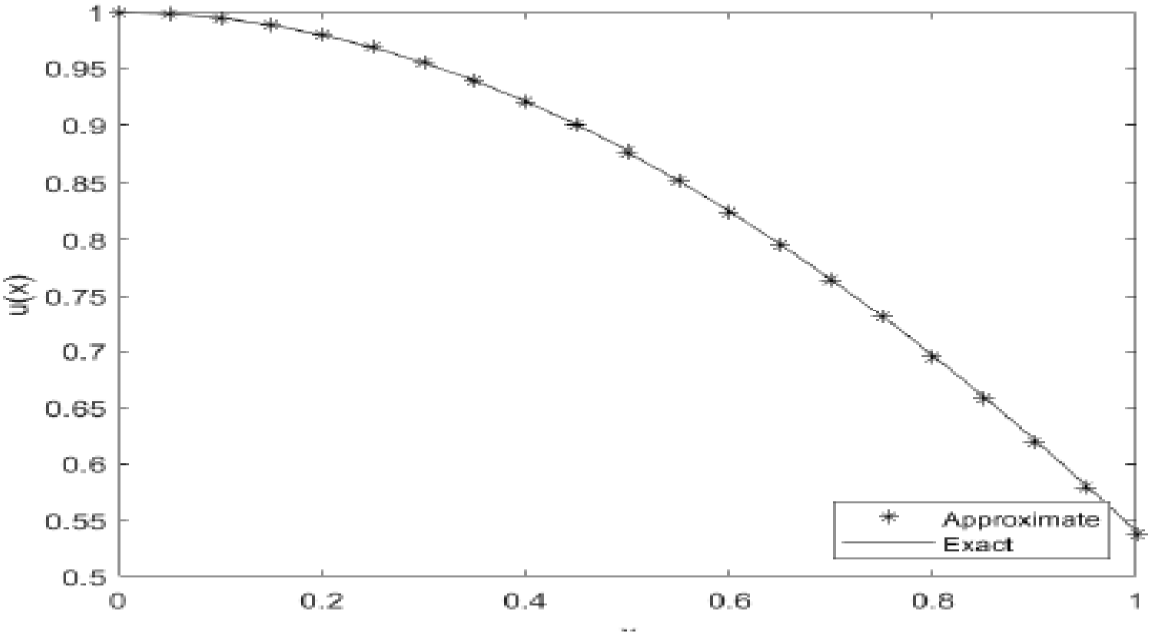

Figure 4: The graph of exact and approximate solution by HPM in Example 4

REFERENCES

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

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

[3] S. Abbasbandy , "Numerical solutions of the integral equations: Homotopy perturbation method and Adomian's decomposition method ," Appl. Math. Comput. 173 (2006) 493-500.

[4] 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.

[5] 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

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

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

[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.