Science Focus, 16 (1) 2011 pp. 12 – 20
Faculty of Pure and Applied Sciences, LAUTECH, Printed in Nigeria

Application of collocation methods for the numerical solution of integro – differential equations by Chebyshev polynomials

Taiwo, O.A[1]; Falade, K.I and Bello, A.K[2]

[1]Department of Mathematics, University of Ilorin, Ilorin, Nigeria; [2]Department of Mathematics, Kwara State College of Education, Oro, Nigeria.

Abstract

Integro – differential equations find special applicability within scientific and mathematical discipline.. In this work , the application of some collocation methods for solving Integro – Differential equations presented. We employed two collocation methods namely, Standard and Perturbed collocation methods and the following collocation points namely, Equally spaced interior collocation, Chebyshev Gauss – Lobatto collocation and Chebyshev Gauss- Radau collocation points were used . Errors analysis and illustrative examples were included to demonstrate the validity and applicability of the methods MATLAB 7 was used to carry out the computation. We conclude that collocation methods discussed can be used as a novel solver for linear Integro – differential equations.

Keywords: Collocation methods, Integro – differential equations, collocation points and error analysis.

Introduction

Integration and differentiation became the basic tools of calculus, with numerous application in science and engineering. In recent years, there has been a growing interest in integro – differential equations. A rigorous mathematical definitions and applications of both integration and differentiation have been considered by some Mathematicians among them were (Taiwo, 1991; Taiwo. and Ishola 2009; Danful and Xufeng, 2007; Karamete and Sezer, 2002) to mention a few and derivatives arise in many areas of science and technology whenever a deterministic relationship involving some continuously changing quantities (modelled by functions and their rate of change expressed as derivatives known as postulated. This is well illustrated by classical mechanics, where the motion of a body is described by its position and velocity as the time varies, this differential equation may explicitly yield the motion. Numerical analysis in mathematics emphasized the rigorous justification of the methods for approximating solutions. Modeling and analysis of physical phenomena in applied in applied sciences often generated non - linear mathematical problems. Non- linearity may be an inner feature of the model. The theory and application of integral equations is an important subject within applied mathematics. Integral and differential equations were used as mathematical models for many and physical situation accrued as reformations of other mathematical problems. Since many physical problems were modelled by integral and differential equations, the numerical solutions of such equations had been highly studied by many authors in recent years (see [Taiwo, 1991; Taiwo. and Ishola 2009; Danful and Xufeng, 2007;) . Numerous works have been focusing on the development of more advance and efficient methods of integral equations such as implicitly linear collocation method, product integration method, Hermite type collocation method, homotopy perturbation method, Chebyshev and Taylor collocation, Haar wavelet, Tau Walsh series methods (Karamete and Sezer, 2002; Regan, 1995; Pour-Mahmoud et al, 2005; Rashed, 2004; Zhao and Corless, 2006; Wang, 2006; Rahimi- Ardabili. and Shahmorad, 2007) to mention a few. Our new approach is to improve on the results and computational cost of the above mentioned methods.

Correspondence author: *E-mail:* oataiwo2002@yahoo.com

For the purpose of our discussion, we consider the integro -- differential equation of the form:

$$y'(x) = p(x)y(x) + g(x) + \lambda \int_a^b k(x,t)\, y(t)\, dt \qquad (1)$$

Under the initial condition ,

$$y(a) = \alpha\ , \qquad (2)$$

where the upper limit of the integral is constant or variable, $\lambda,\ \alpha,\ a$ *are constants*, $g(x),\ p(x)$ *and* $k(x,t)$ are given functions, whereas $y(x)$ is to be determined.

Standard Collocation Method

In order to solve equation (1) together with (2). we assume an approximate solution of the form:

$$y(x) \approx y_N(x) = \sum_{r=0}^{N} a_r T_r(x) \qquad (3)$$

Here, $T_r(x)$ is the Chebyshev polynomial defined as:

$$T_r(x) = Cos\left(r\,Cos^{-1}\left(\frac{2x-b-a}{b-a}\right)\right) \qquad (4)$$

and satisfies the recurrence relation

$$T_{r+1}(x) = 2\left[\frac{2x-b-a}{b-a}\right]T_r(x) - T_{r-1}(x), \qquad a \le x \le b \qquad (5)$$

Thus, substituting (3) into (1), we obtain

$$\sum_{r=0}^{N} a_r T_r'(x) = p(x)\sum_{r=0}^{N} a_r T_r(x) + g(x) + \lambda \int_a^b \left[k(x,t)\sum_{r=0}^{N} a_r T_r(x)\right] dt \qquad (6)$$

After evaluating the integral part of equation (6), the resulting equation is then collocated by the three collocation points mentioned earlier. That is,

(i) *Equally spaced interior collocation points defined by*

$$x_i = a + \left(\frac{b-a}{N}\right)i, \qquad i = 1(1)N \qquad (7)\ ,$$

which produce N algebraic linear systems of equation in N+1 unknown constants $a_r\ (r = 0,1,2,\ \ldots N)$. One extra equation is obtained using equation (2). Thus, altogether we have (N+1) algebraic linear system of equations in (N+1) unknown constants. The resulting (N+1) algebraic linear system of equations are then solved using MATLAB 7 to obtain the unknown constants which are then substituted back into equation (3).

(ii) Chebyshev Gauss Lobatto Collocation points defined by

$$x_j = Cos\left(\frac{\pi j}{N}\right), \qquad j = 1(1)N \qquad (8)$$

which produce N algebraic linear systems of equation in N+1 unknown constants $a_r\ (r = 0,1,2,\ \ldots N)$. One extra equation is obtained using equation (2). Thus, altogether we have (N+1) algebraic linear system of equations in (N+1) unknown constants. The resulting (N+1) algebraic linear system of equations are then solved using MATLAB 7 to obtain the unknown constants which are then substituted back into equation (3).

(iii) ***Chebyshev Gauss – Radau Collocation points defined by***

$$x_k = Cos\left(\frac{\pi k}{N+1}\right), \qquad k = 1(1)N \qquad (9)$$

which produce N algebraic linear systems of equation in N+1 unknown constants a_r $(r = 0,1,2, \ldots N)$. One extra equation is obtained using equation (2). Thus, altogether we have (N+1) algebraic linear system of equations in (N+1) unknown constants. The resulting (N+1) algebraic linear system of equations are then solved using MATLAB 7 to obtain the unknown constants which are then substituted into equation (3).

Perturbed collocation method

In this section, we discuss the Perturbed Collocation Method. This method is aimed at improving the accuracy and efficiency of the standard collocation method discussed in section 1. above. To discuss this method, we assume the form of approximate solution given in equation (3) and then substitute into a slightly perturbed equation (1), we obtain,

$$\sum_{r=0}^{N} a_r T_r'(x) = p(x)\sum_{r=0}^{N} a_r T_r(x) + g(x) + \lambda\int_a^b \left[k(x,t)\sum_{r=0}^{N} a_r T_r(x)\right]dt + H_N(x) \quad (10)$$

where,

$H_N(x) = \tau_1 T_N(x) + \tau_2 T_{N-1}(x)$ *and* $T_N(x)$ *is the Chebyshev polynomial of* $deg\,ree\ N$ *defined, by equation* (1.3), τ_1 *and* τ_2 *are free tau parameters to be* $det\,er\,min\,ed$.

After evaluating the integral part of equation (10), the resulting equation is then collocated by the three collocation points used for standard collocation method. We note here that more algebraic linear system of equations will be solved because of the two free tau parameters introduced and also the collocation points need to be adjusted in order to cater for the unknown parameters to be determined. We also remark here that the method involved more computational work when compare with the standard method discussed in section 2.

1. Numerical Examples.

Example 1.

First we consider the integro – differential equation:

$$y'(x) = 3e^{3x} - \frac{1}{3}(2e^3 + 1)x + \int_0^1 3xty(t)dt,$$

with initial condition $y(0) = 1$

For which the exact solution is $y(x) = e^{3x}$

Example 2:

Consider the first order integro – differential equation:

$$y'(x) = y(x) - Cos(2\pi x) - 2\pi Sin(2\pi x) - \frac{1}{2}Sin(4\pi x) + \int_0^1 Sin(4\pi x + 2\pi t)y(t)dt$$

with initial condition $y(0) = 1$,

for which the exact solution is $y(x) = Cos(2\pi x)$

Example 3:

Consider the first order integro – differential equation:

$$y'(x) = y(x) - \frac{1}{2}x + \frac{1}{1+x}\ln(1+x) + \frac{1}{(\ln 2)^2}\int_0^1 \frac{x}{1+t}\,y(t)\,dt$$

with the initial condition $y(0) = 0$,
for which the exact solution is $y(x) = \ln(1+x)$.

To show the efficiency of the present methods for the problems in comparison with the exact solution, we report absolute error which is defined as:

$$Error = \max_{0\le x\le 1}|y_N - y(x)|$$

Remark

Some numerical results of these examples are shown in tables 1 – 9. Results indicate that the convergence rate is very fast and as N increases, it is worth noting that the Perturbed Collocation method with Gauss – Radau points is the fast convergence to the solutions.

Table 1: Numerical results of example 1 for case N=4.

x	Standard Collocation Method			Perturbed Collocation Method		
	Equally Spaced Points	Gauss Lobatto Points	Gauss Radau Points	Equally Spaced Points, (PESP)	Gauss Lobatto Points(PGLP)	Gauss Radau Points(PGRP
0	0	0	0	0	0	0
0.2	9.561×10^{-3}	8.334×10^{-3}	6.334×10^{-3}	8.312×10^{-7}	8.302×10^{-7}	6.291×10^{-7}
0.4	3.413×10^{-2}	3.213×10^{-2}	9.892×10^{-2}	6.915×10^{-6}	6.811×10^{-7}	6.721×10^{-7}
0.6	5.718×10^{-2}	4.161×10^{-2}	2.143×10^{-2}	4.467×10^{-6}	4.401×10^{-6}	4.391×10^{-6}
0.8	6.001×10^{-2}	5.001×10^{-2}	3.222×10^{-2}	4.114×10^{-6}	4.099×10^{-6}	$4{,}043\times10^{-6}$
1.0	6.814×10^{-2}	5.631×10^{-2}	4.007×10^{-2}	2.398×10^{-6}	2.314×10^{-6}	2.299×10^{-6}

Table 2: Numerical results of example 1 for case N=6.

x	Standard Collocation Method			Perturbed Collocation Method		
	Equally Spaced Points	Gauss Lobatto Points	Gauss Radau Points	Equally Spaced Points(PESP)	Gauss Lobatto Points(PGLP)	Gauss Radau Points(PGRP
0	0	0	0	0	0	0
0.2	4.326×10^{-3}	2.698×10^{-3}	1.999×10^{-3}	9.899×10^{-9}	9.727×10^{-9}	9.711×10^{-9}
0.4	4.431×10^{-3}	2.432×10^{-3}	2.034×10^{-3}	4.628×10^{-8}	4.533×10^{-8}	4.439×10^{-8}
0.6	5.632×10^{-3}	3.648×10^{-3}	2.481×10^{-3}	3.611×10^{-8}	3.516×10^{-8}	3.428×10^{-8}
0.8	8.931×10^{-3}	6.049×10^{-3}	3.411×10^{-3}	3.106×10^{-8}	3.063×10^{-8}	3.048×10^{-8}
1.0	9.661×10^{-3}	7.134×10^{-3}	4.672×10^{-3}	2.662×10^{-8}	1.996×10^{-8}	1.197×10^{-8}

Table 3: Numerical results of example 1 for case N=10

X	Standard Collocation Method			Perturbed Collocation Method		
	Equally Spaced Points	Gauss Lobatto Points	Gauss Radau Points	Equally Spaced Points(PESP)	Gauss Lobatto Points(PGLP)	Gauss Radau Points(PGRP)
0	0	0	0	0	0	0
0.2	6.482×10^{-5}	6.342×10^{-5}	6.163×10^{-5}	9.899×10^{-9}	9.727×10^{-9}	9.711×10^{-9}
0.4	5.187×10^{-5}	6.019×10^{-5}	5.619×10^{-5}	4.628×10^{-8}	4.533×10^{-8}	4.439×10^{-8}
0.6	5.019×10^{-5}	5.092×10^{-5}	4.992×10^{-5}	3.611×10^{-8}	3.516×10^{-8}	3.428×10^{-8}
0.8	4.918×10^{-5}	4.678×10^{-5}	4.593×10^{-5}	3.106×10^{-8}	3.063×10^{-8}	3.048×10^{-8}
1.0	4.231×10^{-5}	4.238×10^{-5}	4.112×10^{-5}	2.662×10^{-8}	1.996×10^{-8}	1.197×10^{-8}

Table 4: Numerical results of example 2 for case N=4

X	Standard Collocation Method			Perturbed Collocation Method		
	Equally Spaced Points	Gauss Lobatto Points	Gauss Radau Points	Equally Spaced Points(PESP)	Gauss Lobatto Points(PGLP)	Gauss Radau Points(PGRP)
0	0	0	0	0	0	0
0.2	2.571×10^{-4}	2.412×10^{-4}	2.411×10^{-4}	6.233×10^{-6}	6.221×10^{-6}	6.209×10^{-6}
0.4	4.792×10^{-4}	3.629×10^{-4}	2.969×10^{-4}	6.044×10^{-6}	6.016×10^{-6}	6.009×10^{-6}
0.6	7.612×10^{-4}	6.992×10^{-4}	5.668×10^{-4}	5.893×10^{-6}	5.723×10^{-6}	5.657×10^{-6}
0.8	9.678×10^{-3}	8.245×10^{-3}	7.899×10^{-4}	4.659×10^{-6}	4.418×10^{-6}	4.329×10^{-6}
1.0	1.310×10^{-3}	1.118×10^{-3}	1.116×10^{-3}	2.493×10^{-6}	2.239×10^{-6}	2.112×10^{-6}

Table 5: Numerical results of example 2 for case N=6

	Standard Collocation Method			*Perturbed Collocation Method*		
	Equally Spaced Points	*Gauss Lobatto Points*	*Gauss Radau Points*	*Equally Spaced Points(PESP)*	*Gauss Lobatto Points(PGLP)*	*Gauss Radau Points(PGRP)*
	0	0	0	0	0	0
2	6.642×10^{-5}	6.287×10^{-5}	6.248×10^{-5}	9.698×10^{-9}	9.432×10^{-9}	9.413×10^{-9}
4	5.670×10^{-5}	5.587×10^{-5}	5.421×10^{-5}	3.116×10^{-8}	2.115×10^{-8}	2.113×10^{-9}
6	5.318×10^{-5}	5.211×10^{-5}	5.113×10^{-5}	3.597×10^{-8}	3.403×10^{-8}	3.312×10^{-8}
8	4.420×10^{-5}	4.258×10^{-5}	4.031×10^{-5}	2.993×10^{-8}	2.675×10^{-8}	2.667×10^{-8}
0	1.678×10^{-5}	2.067×10^{-5}	1.983×10^{-5}	2.664×10^{-8}	2.447×10^{-8}	2.389×10^{-8}

Table 6: Numerical results of example 2 for case N=10

x	Standard Collocation Method			Perturbed Collocation Method		
	Equally Spaced Points	Gauss Lobatto Points	Gauss Radau Points	Equally Spaced Points (PESP)	Gauss Lobatto Points(PGLP)	Gauss Radau Points (PGRP).
0	0	0	0	0	0	0
0.2	7.342×10^{-6}	6.992×10^{-6}	6.812×10^{-6}	4.914×10^{-12}	4.779×10^{-12}	4.666×10^{-12}
0.4	6.998×10^{-6}	6.813×10^{-6}	6.569×10^{-6}	7.889×10^{-12}	7.856×10^{-12}	7.564×10^{-12}
0.6	4.985×10^{-6}	4.718×10^{-6}	3.999×10^{-6}	9.132×10^{-12}	9.067×10^{-12}	8.967×10^{-12}
0.8	2.719×10^{-6}	2.499×10^{-6}	2.086×10^{-6}	9.936×10^{-12}	9.883×10^{-12}	9.432×10^{-12}
1.0	1.667×10^{-6}	1.443×10^{-6}	1.136×10^{-6}	3.116×10^{-11}	2.660×10^{-11}	2.163×10^{-11}

Table 7 Numerical results of example 3 for case N=4

x	Standard Collocation Method			Perturbed Collocation Method		
	Equally Spaced Points	Gauss Lobatto Points	Gauss Radau Points	Equally Spaced Points (PESP)	Gauss Lobatto Points(PGLP)	Gauss Radau Points (PGRP).
0	0	0	0	0	0	0
0.2	9.438×10^{-5}	5.248×10^{-5}	5.214×10^{-5}	9.995×10^{-6}	8.772×10^{-6}	8.719×10^{-6}
0.4	3.734×10^{-4}	1.067×10^{-4}	1.044×10^{-4}	8.941×10^{-6}	8.923×10^{-6}	8.892×10^{-6}
0.6	8.721×10^{-4}	1.039×10^{-3}	1.027×10^{-3}	6.813×10^{-5}	6.632×10^{-6}	6.602×10^{-6}
0.8	1.432×10^{-3}	1.021×10^{-3}	1.019×10^{-3}	3.117×10^{-5}	2.559×10^{-6}	2.141×10^{-6}
1.0	2.331×10^{-3}	1.013×10^{-3}	1.007×10^{-3}	1.692×10^{-5}	2.013×10^{-6}	2.009×10^{-6}

Table 8 Numerical results of example 3 for case N=6

x	Standard Collocation Method			Perturbed Collocation Method		
	Equally Spaced Points	Gauss Lobatto Points	Gauss Radau Points	Equally Spaced Points (PESP)	Gauss Lobatto Points(PGLP)	Gauss Radau Points (PGRP).
0	0	0	0	0	0	0
0.2	7.813×10^{-5}	7.694×10^{-5}	7.389×10^{-5}	8.613×10^{-7}	8.555×10^{-7}	8.431×10^{-7}
0.4	4.639×10^{-5}	4.610×10^{-5}	4.545×10^{-5}	7.349×10^{-7}	7.321×10^{-7}	7.222×10^{-7}
0.6	2.180×10^{-5}	2.143×10^{-5}	2.122×10^{-5}	7.631×10^{-7}	7.249×10^{-7}	7.218×10^{-7}
0.8	9.063×10^{-5}	1.453×10^{-5}	1.025×10^{-5}	5.699×10^{-7}	5.562×10^{-7}	5.382×10^{-7}
1.0	7.236×10^{-4}	1.112×10^{-5}	1.002×10^{-5}	4.840×10^{-7}	4.118×10^{-7}	3.899×10^{-7}

Table 9 Numerical results of example 3 for case N=10

x	Standard Collocation Method			Perturbed Collocation Method		
	Equally Spaced Points	Gauss Lobatto Points	Gauss Radau Points	Equally Spaced Points (PESP)	Gauss Lobatto Points(PGLP)	Gauss Radau Points (PGRP).
0	0	0	0	0	0	0
0.2	8.673×10^{-6}	8.513×10^{-6}	8.421×10^{-6}	9.689×10^{-10}	9.237×10^{-10}	9.210×10^{-10}
0.4	5.984×10^{-6}	5.811×10^{-6}	5.593×10^{-6}	7.887×10^{-10}	7.665×10^{-10}	7.658×10^{-10}
0.6	4.772×10^{-6}	4.412×10^{-6}	4.389×10^{-6}	6.991×10^{-10}	6.842×10^{-10}	6.791×10^{-10}
0.8	4.249×10^{-6}	4.129×10^{-6}	4.093×10^{-6}	6.587×10^{-10}	6.336×10^{-10}	6.229×10^{-10}
1.0	3.896×10^{-6}	3.721×10^{-6}	3.118×10^{-6}	1.862×10^{-10}	1.218×10^{-10}	1.109×10^{-10}

Conclusion

The Collocation methods was used for evaluating integro – differential equations. Three examples were examined as demonstration. It was concluded that Perturbed Collocation method with the three collocation points is very powerful and efficient method in evaluating a wide class of integro – differential equations and also worth noting that Perturbed collocation method with Gauss - Radau points proved to be the best in terms of accuracy achieved.

References

Danfu, H. and S. Xufeng, 2007. Numerical solution of integro differential equations by CAS wavelet operational matrix of integration. *Applied Math. Comput.*, 194; 460-466.DOI: 10.1016/j.amc . 2007.04.048.

Karamete, A and M . Sezer, 2002. A Taylor collocation method for the solution of linear integro – differential equations. *Int. J. Comput. Math*, 79, 987-1000. DOI: 10.1080/00207160212116.

Pour-Mahmoud, J., M . Y Rahimi- Ardabili and S. Shahmorad (2005). Numerical solution of the system of Fredholm integro-differential equations by the tau method: *Applied .Math. Comput.*, 168, 465-478. DOI: 10,1016/j.amc 2004.09.026

Rahimi- Ardabili, M. Y. and Shahmorad, S . 2007. Iterative numerical solution of nonlinear integro-differential equations by the tau method. *Applied . Math . Comput.*. 193, 514-522. DOI: 10,1016/j.amc.2007.04.002.

Rashed, M. T. (2004). Numerical solution of functional differential, integral and integro –differential equations. *Applied Numeric. Math.*, 156, 485-492. DOI: 10,1016/j.amc. 2003.08.021

Regan, D. (1995). Existence results for non-linear integral equations. *J. Math. Anal . Appl.* 192, 3. 705 – 726

Taiwo, O . A (1991). Collocation methods for singularly perturbed ordinary differential equations. Ph.D thesis (unpublished), University of Ilorin, Nigeria.

Taiwo, O . A. and Ishola, C . Y (2009). Collocation Approximation methods for the numerical solution of Integral equations. *Intern. J. of pure and Applied Sciences*. 2(4). 29 – 37.

Wang, W (2006). An algorithm for solving the higher- order nonlinear Volterra- Fredholm integro – differential equationswith mechanization. *Applied Math . Comput.*. 172, 1-23.DOI: 10,1016/j.amc.2005.01.116.

Zhao, J . and R . M. Corless (2006). Compact finite difference method for integro – differential equations. *Applied .Math, Comput.*. 177, 271-28 DOI: 10,1016/j.amc.2005.11.07

Appendix.

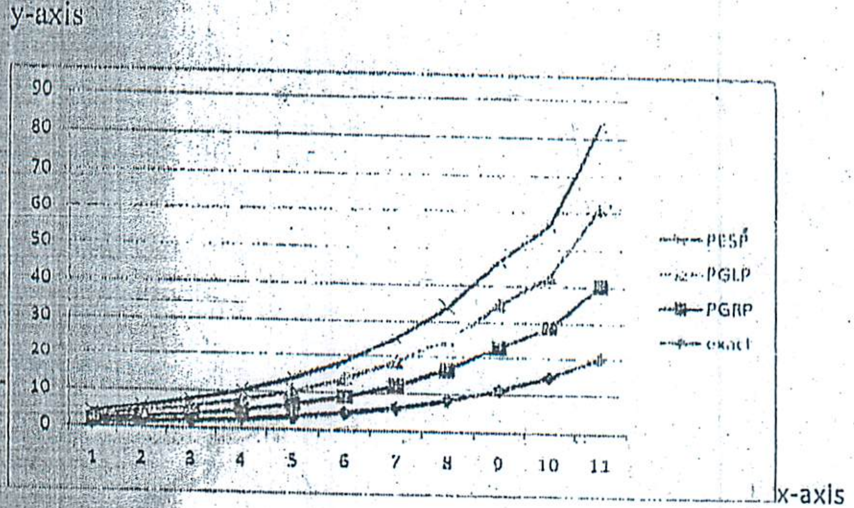

Figure 1:Graphical representation of example 1 for the exact solution and the collocation method.

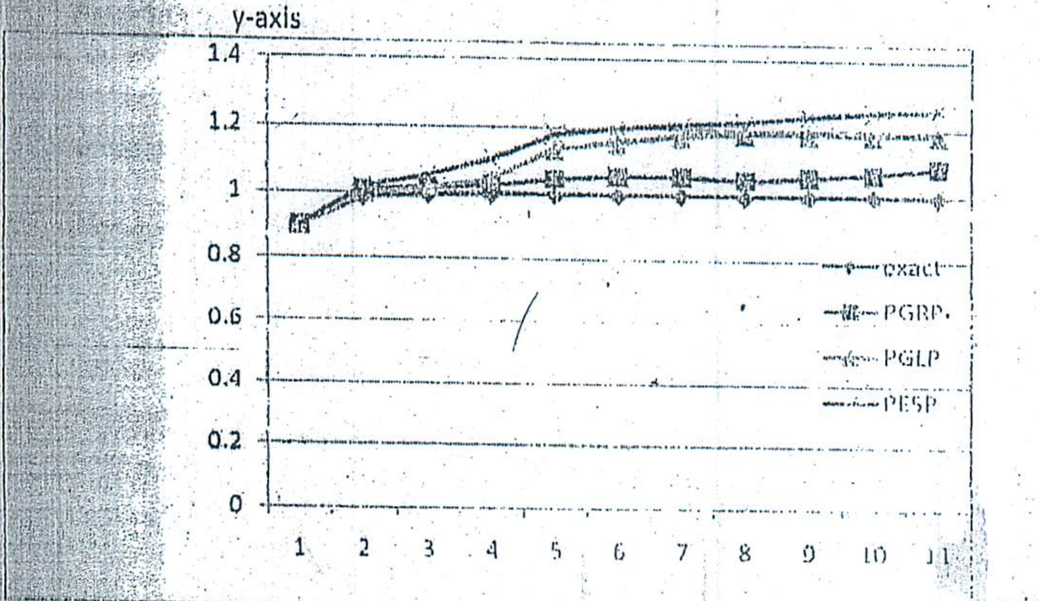

Figure 2:Graphical representation of example 2 for the exact solution and the collocation method.

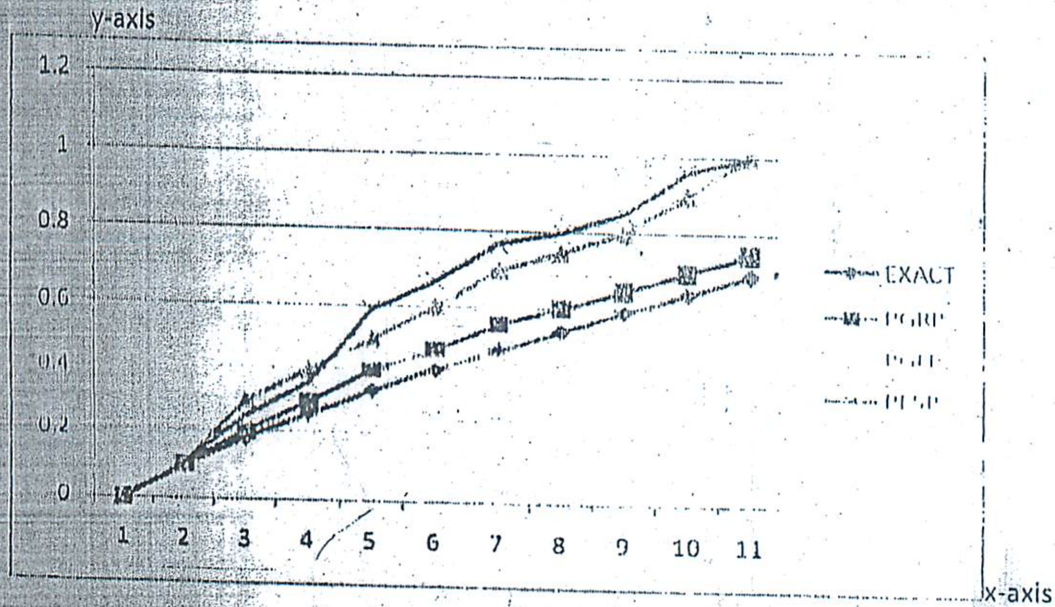

Figure 3:Graphical representation of example 3 for the exact solution and the collocation method.