Collocation Approximation Methods For The Numerical Solutions Of General Nth Order Nonlinear Integro-Differential Equations By Canonical Polynomial

[1]Taiwo O. A And [2]Raji M. T

[1]Department of Mathematics, University of Ilorin
[2]Department of Mathematics and Statistics, The Poly., Ibadan

Abstract

In this Paper, a method based on the Tau method by canonical polynomials as the basis function is developed to find the numerical solutions of general nth order nonlinear integro-differential equations. The differential parts appearing in the equation are used to construct the canonical polynomials and the nonlinear cases are linearized by the Newton's linearization scheme of order n and hence resulted to the use of iteration. Numerical examples are given to illustrate the effectiveness, convergence and the computational cost of the methods.

Introduction

Nonlinear differential equations are used in modelling many real life problems in science and engineering. Nonlinear ordinary differential equations mostly defy closed form solutions because the actual elegant theory valid for their linear counterparts often fails for them. Newton's linearization procedures leading to the use of iteration are commonly employed to facilitate provision of analytic solution.

This paper concerns the development of the Tau numerical method by canonical polynomials as the basis function (see Taiwo [8]) for the solution of nth order integro-differential equation. The Tau numerical method by Chebyshev polynomials has found extensive application in recent year (see Taiwo and Evans [10], Taiwo [9], Taiwo and Ishola [11]) to mention a few for the case of numerical solution of ordinary differential equations. Applications of the Chebyshev Polynomials as basis function and their merits in solving ordinary differential equations numerically have been discussed by many authors (see Taiwo [7], Asady and Kajani [1], Danfu and Xafeng [3], Rahimi-Ardabili and Shahmorad [4], Tavassoli et al. [12] and Behiry and Mohamed [2]). Many different approaches have also been proposed in the literature to handle integro-differential equations numerically (Wang and He [6], Zhao and Corless [13], Shahmorad et al. [5]).

This paper is aimed, therefore, to work in this direction of extending the Tau numerical method by canonical polynomial as the basis function for the solution of general nth order integro-differential equations. Finally, some results are presented to demostrate the efficiency of the new method compared with those results available in Behiry and Mohamed [2].

For the purpose of our discussion, we consider the nonlinear general nth-order ordinary integro-differential equation of the form:

$$Gy_n \equiv y^n(x) - f(x, y(x), y'(x))y'(x) + \int_a^b k(x,t)y(t)dt = g(x); \qquad a \le x \le b. \tag{1}$$

together with the linear boundary conditions

$$\begin{aligned} &A_1 y(a) + A_2 y'(a) + A_3 y''(a) + A_4 y'''(a) + \cdots + A_{n-1} y^{n-2} = \alpha \\ &B_1 y(b) + B_2 y'(b) + B_3 y''(b) + B_4 y'''(b) + \cdots + B_{n-1} y^{n-2} = \beta \\ &C_1 y(c) + C_2 y'(c) + C_3 y''(c) + C_4 y'''(c) + \cdots + C_{n-1} y^{n-2} = \gamma \\ &D_1 y(d) + D_2 y'(d) + D_3 y''(d) + D_4 y'''(d) + \cdots + D_{n-1} y^{n-2} = \chi \\ &\vdots \end{aligned} \tag{2}$$

Here, $A, B, C, D, \alpha, \beta, \gamma$ and χ are constants and $y(x)$ are unknown functions, $g(x)$ and $k(x,t)$ are any given smooth function and in this case it can be linear or nonlinear and f is generally nonlinear.

Many numerical techniques have been used successively for equations (1) & (2) and in this section, we discussed in details a straight forward yet generally applicable techniques, the Tau numerical collocation method by canonical polynomial as the basis function. The Newton's scheme from the Taylor's series expansion is represented around (x_n, t_n, y_n) in the following form

$$\begin{aligned} &G + \Delta y \frac{\partial G}{\partial y} + \Delta y' \frac{\partial G}{\partial y'} + \Delta y'' \frac{\partial G}{\partial y''} + \cdots + \Delta y^{n-1} \frac{\partial G}{\partial y^{n-1}} + \Delta y^n \frac{\partial G}{\partial y^n} \\ &+ \int_a^b \Big[k(x_n, t_n, y_n) + (x - x_n) \frac{\partial G}{\partial x}(x_n, t_n, y_n) + (t - t_n) \frac{\partial G}{\partial t}(x_n, t_n, y_n) \\ &+ (y - y_n) \frac{\partial G}{\partial y}(x_n, t_n, y_n) \Big] y(t)dt = g(x); \qquad a \le x \le b. \end{aligned} \tag{3}$$

The integral parts of equation (3), where t is an independent variable, y is the dependent variable, are integrated with respect to t to obtain

$$\begin{aligned} &G + \Delta y \frac{\partial G}{\partial y} + \Delta y' \frac{\partial G}{\partial y'} + \Delta y'' \frac{\partial G}{\partial y''} + \cdots + \Delta y^{n-1} \frac{\partial G}{\partial y^{n-1}} + \Delta y^n \frac{\partial G}{\partial y^n} + \int_{t_n}^t k(x_n, t_n, y_n) y(t) dt \\ &+ \left[(x - x_n) \frac{\partial G}{\partial x}(x_n, t_n, y_n) + (t - t_n) \frac{\partial G}{\partial t}(x_n, t_n, y_n) + (y - y_n) \frac{\partial G}{\partial y}(x_n, t_n, y_n) \right] \\ &\times \int_{t_n}^t (t - t_n) dt = g(x) \end{aligned} \tag{4}$$

Hence, from equation (1), we obtain the following

$$\left.\begin{aligned} &\frac{\partial G}{\partial y} = f_y y', \\ &\frac{\partial G}{\partial y'} = -(f_y)y' - f, \\ &\frac{\partial G}{\partial y_n''} = 1, \\ &\vdots \end{aligned}\right\}. \tag{5}$$

and $\Delta y_k^{(j)}(x) = y_{k+1}^{(j)}(x) - y_k^{(j)}(x), \quad j = 1,2,3,\ldots$

Thus, substituting equation (5) into equation (4), after simplification, we obtain

$$y_{n+1}^{n}(x) + \cdots + y_{n+1}(x) + \left(f_n + f_{yn}y'\right)y'_{n+1}(x) + f_{yn}y'y_{n+1}(x) + f_n y_n y'_n(x)$$

$$+\left[k(x_n, t_n, y_n) + (x - x_n)\frac{\partial G}{\partial y_n}(x_n, t_n, y_n)\right]\int_a^b y_{n+1}(t)dt$$

$$+\left[k(x_n, t_n, y_n) + \frac{\partial G}{\partial y_n}(x_n, t_n, y_n)\right]\int_a^b \{(t - t_n) + \cdots + (y_{n+1}^n - y_n^n)\} y_{n+1}(t)dt$$

$$= g(x). \tag{6}$$

Thus, equation (6) is the linearized form of equation (1).

In order to solve equation (6), we assumed an approximate solution of the form

$$y_{N,n+1}(x) = \sum_{n=0}^{N} a_{N,n}\Phi_{N,n}(x); \qquad a \le x \le b, \tag{7}$$

where $\Phi_{N,n}(x), \quad (n = 0,1,2,\ldots,t)$ are the canonical basis functions generated below, $a_{N,n}$ are the unknown constants to be determined, N is the degree of the approximant used and n is the number of iteration to be carried-out.

Construction of Canonical Polynomials for nth-order IDEs.

Consider the nth-order integro-differential equation

$$P_0(x)y(x) + P_1(x)y'(x) + P_2(x)y''(x) + P_3(x)y'''(x) + \cdots + P_n(x)y^n(x) + \int_a^b k(x,t)y(t)dt = g(x) \tag{8}$$

subject to the conditions

$$y(a) + y'(a) + \cdots + y^{n-1}(a) = A \tag{9}$$

$$y(b) + y'(b) + \cdots + y^{n-1}(b) = B \tag{10}$$

Where $P_1(x)$; I = 0, 1, 2, . . . n can be variable or constants coefficients.

We define the following operator

$$D \equiv P_n \frac{d^n}{dx^n} + P_{n-1}\frac{d^{n-1}}{dx^{n-1}} + \cdots + P_1 \frac{d}{dx} + P_0. \qquad (11)$$

A set of polynomials $\Phi_n(x)$, $(n = 0,1,2,3\ldots)$ is defined by

$$D\Phi_n(x) = x^n. \qquad (12)$$

Which is uniquely associated with the operated D and, which is obtained recursively as,

$$\Phi_n(x) = \frac{1}{P_0}\{x^n - P_1 n\Phi_{n-1}(x) - P_2 n(n-1)\Phi_{n-2}(x) - P_3 n(n-1)(n-2)\Phi_{n-3}(x)\cdots\}; \qquad n \geq 0. \quad (13)$$

Construction of Canonical polynomial for case n = 2

In order to generate the canonical polynomial, we consider the differential part of equation (6) i. e.

$$L \equiv P_2 \frac{d^2}{dx^2} + P_1 \frac{d}{dx} + P_0$$

$$L\Phi_i(x) = x^i$$

$$Lx^i = P_2 i(i-1)x^{i-2} + P_1 i x^{i-1} + P_0 x^i$$

$$L[L\Phi_i(x)] = P_2 i(i-1)L\Phi_{i-2}(x) + P_1 i L\Phi_{i-1}(x) + P_0 L\Phi_i(x)$$

$$x^i = P_2 i(i-1)\Phi_{i-2}(x) + P_1 i\Phi_{i-1}(x) + P_o\Phi_i(x)$$

$$\Phi_i(x) = \frac{1}{P_0}\left[x^i - P_1 i\Phi_{i-1}(x) - P_2 i(i-1)\Phi_{i-2}(x)\right], i \geq 0; P_0 \neq 0 \qquad (14)$$

Thus, equation (14) is the recurrence relation

For i = 0: $$\Phi_0(x) = \frac{1}{P_0}$$

For i =1: $$\Phi_1(x) = \frac{1}{P_0}\left(x - P_1\Phi_0(x)\right) = \frac{x}{P_0} - \frac{P_1}{P_0^2}$$

For i = 2: $$\Phi_2(x) = \frac{1}{P_0}\left[x^2 - 2P_1\Phi_1(x) - 2P_2\Phi_0(x)\right] = \frac{x^2}{P_0} - 2x\frac{P_1}{P_0^2} + 2\frac{P_1^2}{P_0^3} - 2\frac{P_2}{P_0^2}$$

For i = 3:

$$\Phi_3(x) = \frac{1}{P_0}\left[x^3 - 3P_1\Phi_2(x) - 6P_2\Phi_1(x)\right]$$

$$\Phi_3(x) = \frac{x^3}{P_0} - \frac{3x^2P_1}{P_0^2} + \frac{6xP_1^2}{P_0^3} - \frac{6P_1^3}{P_0^4} - \frac{6xP_2}{P_0^2}$$

For i = 4:

$$\Phi_4(x) = \frac{1}{P_0}\left[x^4 - 4P_1\Phi_3(x) - 12P_2\Phi_2(x)\right]$$

$$= \left[\frac{x^4}{P_0} - \frac{4x^3P_1}{P_0^2} + \frac{12x^2P_1^2}{P_0^3} - \frac{24xP_1^3}{P_0^4} + \frac{24P_1^4}{P_0^5} + \frac{48xP_1P_2}{P_0^3} - \frac{12x^2P_2}{P_0^2} - \frac{24P_1^2P_2}{P_0^4} + \frac{24P_2^2}{P_0^3}\right]$$

$$\vdots$$

For the case **n = 3**, we define our operator as:

$$L \equiv P_3\frac{d^3}{dx^3} + P_2\frac{d^2}{dx^2} + P_1\frac{d}{dx} + P_0$$

$$L\Phi_i(x) = x^i$$

$$Lx^i = P_3 i(i-1)(i-2)x^{i-3} + P_2 i(i-1)x^{i-2} + P_1 i x^{i-1} + P_0 x^i$$

$$L[L\Phi_i(x)] = P_3 i(i-1)(i-2)L\Phi_{i-3}(x) + P_2 i(i-1)L\Phi_{i-2}(x) + P_1 i L\Phi_{i-1}(x) + P_0 L\Phi_i(x)$$

$$x^i = P_3 i(i-1)(i-2)\Phi_{i-3}(x) + P_2 i(i-1)\Phi_{i-2}(x) + P_1 i\Phi_{i-1}(x) + P_o\Phi_i(x)$$

$$\Phi_i(x) = \frac{1}{P_0}\left[x^i - P_1 i\Phi_{i-1}(x) - P_2 i(i-1)\Phi_{i-2}(x) - P_3 i(i-1)(i-2)\Phi_{i-3}(x)\right], i \ge 0; P_0 \ne 0 \quad (15)$$

Thus, equation (15) is the recurrence relation

For i = 0:

$$\Phi_0(x) = \frac{1}{P_0}$$

For i =1:

$$\Phi_1(x) = \frac{1}{P_0}\left(x - P_1\Phi_0(x)\right) = \frac{x}{P_0} - \frac{P_1}{P_0^2}$$

For i = 2:

$$\Phi_2(x) = \frac{1}{P_0}\left[x^2 - 2P_1\Phi_1(x) - 2P_2\Phi_0(x)\right] = \frac{x^2}{P_0} - 2x\frac{P_1}{P_0^2} + 2\frac{P_1^2}{P_0^3} - 2\frac{P_2}{P_0^2}$$

For i = 3:

$$\Phi_3(x) = \frac{1}{P_0}\left[x^3 - 3P_1\Phi_2(x) - 6P_2\Phi_1(x) - 6P_3\Phi_0(x)\right]$$

$$\Phi_3(x) = \frac{x^3}{P_0} - \frac{3x^2P_1}{P_0^2} + \frac{6xP_1^2}{P_0^3} - \frac{6P_1^3}{P_0^4} - \frac{6xP_2}{P_0^2} - \frac{6P_3}{P_0^2}$$

For i = 4:

$$\Phi_4(x) = \frac{1}{P_0}\left[x^4 - 4P_1\Phi_3(x) - 12P_2\Phi_2(x) - 24P_3\Phi_1(x)\right]$$

$$= \left[\frac{x^4}{P_0} - \frac{4x^3P_1}{P_0^2} + \frac{12x^2P_1^2}{P_0^3} - \frac{24xP_1^3}{P_0^4} + \frac{24P_1^4}{P_0^5} - \frac{72P_1^2P_2}{P_0^4} + \frac{48P_1\,P_3}{P_0^3} - \frac{12x^2P_2}{P_0^2} - \frac{24xP_3}{P_0^2} + \frac{24P_2^2}{P_0^3}\right]$$

$$\vdots$$

For the case **n = 4**, we define our operator as:

$$L \equiv P_4\frac{d^4}{dx^4} + P_3\frac{d^3}{dx^3} + P_2\frac{d^2}{dx^2} + P_1\frac{d}{dx} + P_0$$

$$L\Phi_i(x) = x^i$$

$$Lx^i = P_4 i(i-1)(i-2)(i-3)x^{i-4} + P_3 i(i-1)(i-2)x^{i-3} + P_2 i(i-1)x^{i-2} + P_1 i x^{i-1} + P_0 x^i$$

$$L[L\Phi_i(x)] = P_4 i(i-1)(i-2)(i-3)L\Phi_{i-4}(x) + P_3 i(i-1)(i-2)L\Phi_{i-3}(x) + P_2 i(i-1)L\Phi_{i-2}(x) + P_1 i L\Phi_{i-1}(x) + P_0 L\Phi_i(x)$$

$$x^i = P_4 i(i-1)(i-2)(i-3)\Phi_{i-4}(x) + P_3 i(i-1)(i-2)\Phi_{i-3}(x) + P_2 i(i-1)\Phi_{i-2}(x) + P_1 i\Phi_{i-1}(x) + P_o\Phi_i(x)$$

$$\Phi_i(x) = \frac{1}{P_0}\left[x^i - P_1 i\Phi_{i-1}(x) - P_2 i(i-1)\Phi_{i-2}(x) - P_3 i(i-1)(i-2)\Phi_{i-3}(x) + P_4 i(i-1)(i-2)(i-3)\Phi_{i-4}(x)\right],$$

$$i \geq 0; P_0 \neq 0 \qquad (16)$$

Thus, equation (16) is the recurrence relation

For i = 0: $$\Phi_0(x) = \frac{1}{P_0}$$

For i =1: $$\Phi_1(x) = \frac{1}{P_0}(x - P_1\Phi_0(x)) = \frac{x}{P_0} - \frac{P_1}{P_0^2}$$

For i = 2: $$\Phi_2(x) = \frac{1}{P_0}\left[x^2 - 2P_1\Phi_1(x) - 2P_2\Phi_0(x)\right] = \frac{x^2}{P_0} - 2x\frac{P_1}{P_0^2} + 2\frac{P_1^2}{P_0^3} - 2\frac{P_2}{P_0^2}$$

For i = 3:

$$\Phi_3(x) = \frac{1}{P_0}\left[x^3 - 3P_1\Phi_2(x) - 6P_2\Phi_1(x) - 6P_3\Phi_0(x)\right]$$

$$\Phi_3(x) = \frac{x^3}{P_0} - \frac{3x^2P_1}{P_0^2} + \frac{6xP_1^2}{P_0^3} - \frac{6P_1^3}{P_0^4} - \frac{6xP_2}{P_0^2} + \frac{6P_1P_2}{P_0^3} - \frac{6P_3}{P_0^2}$$

For i = 4:

$$\Phi_4(x) = \frac{1}{P_0}\left[x^4 - 4P_1\Phi_3(x) - 12P_2\Phi_2(x) - 24P_3\Phi_1(x)\right]$$

$$= [\frac{x^4}{P_0} - \frac{4x^3P_1}{P_0^2} + \frac{12x^2P_1^2}{P_0^3} - \frac{24xP_1^3}{P_0^4} + \frac{24P_1^3}{P_0^5} - \frac{48P_1\,P_2}{P_0^4} + \frac{24xP_2}{P_0^3} + \frac{24P_3}{P_0^3} - \frac{12x^2P_2}{P_0^2} + \frac{24xP_1}{P_0^3} - \frac{24P_1^2P_2}{P_0^4}$$

$$+ \frac{24P_2^2}{P_0^3} - \frac{24xP_3}{P_0^2} + \frac{24P_1P_3}{P_0^3} - \frac{24P_4}{P_0^2}]$$

$$\vdots$$

DESCRIPTION OF METHODS

PERTURBED COLLOCATION METHOD

In this section, we discuss the collocation Tau numerical solution for the solutions of the linearized equation (6).

In this method, after the evaluation of the integrals in equation (6), equation (7) is substituted into a slightly perturbed equation (6) to give

$$y_{N,n+1}^{n}(x) + \cdots + y''_{N\,n+1}(x) + \left(f_n + f_{yn}y'(x)\right)y'_{N\,n+1}(x) + f_{yN,n}y'_n y_{N\,n+1}(x) + f_n y_{N\,n} y'_{N\,n}(x)$$

$$+ \left[\frac{\partial G}{\partial y_n}\left(x_n, t_n, y_{N\,n}\right) + (x - x_n)\frac{\partial G}{\partial y_n}\left(x_n, t_n, y_{N\,n}\right)\right]\int_a^b y_{N\,n+1}(t)dt$$

$$+ \left[\frac{\partial G}{\partial x_n}\left(x_n, t_n, y_{N\,n}\right) + \frac{\partial G}{\partial y_n}\left(x_n, t_n, y_{N\,n}\right)\right] \times \int_a^b (t - t_n) + \cdots + (y_{N\,n+1} - y_{N\,n})y_{N\,n+1}(t)dt$$

$$= g(x) + H_N(x). \tag{17}$$

together with the boundary conditions

$$\begin{aligned} &y_{N,n+1}^{n-1}(a) + \cdots + y_{N\,n+1}(a) = \alpha \\ &y_{N,n+1}^{n-1}(b) + \cdots + y_{N\,n+1}(b) = \beta \\ &\vdots \\ &y_{N,n+1}^{n-1}(f) + \cdots + y_{N\,n+1}(f) = \chi \end{aligned} \tag{18}$$

Where $H_N(x)=\sum_{i=0}^{N}\tau_i T_{N-i+1}(x)$ and $\tau_N(x)$ is the Chebyshev polynomials of degree N valid in [a, b] and is defined by,

$$T_N(x)=Cos\left[NCos^{-1}\left\{\frac{2x-b-a}{b-a}\right\}\right], \qquad a\le x\le b. \tag{19}$$

The recurrence relation of equation (19) is given as:

$$T_{n+1}(x)=2\left\{\frac{2x-b-a}{b-a}\right\}T_n(x)-T_{n-1}(x). \tag{20}$$

The Chebyshev polynomials oscillate with equal amplitude in the range under consideration and this makes the Chebyshev polynomials suitable in function approximation problem.

Thus, equation (17) is collocated at point $x=x_k$, hence, we get

$$\begin{aligned}&y^{n}_{N,n+1}(x_k)+\cdots+y''_{N\,n+1}(x_k)+\left(f_n+f_{yn}y'(x_k)y'_{N\,n+1}(x_k)+f_{yN,n}(x_k)y_{N},_{n+1}(x_k)\right.\\&+f_n y_N,_n(x_k)y'_N,_{n+1}(x_k)+\left[\frac{\partial G}{\partial y_n}(x_n,t_n,y_N,_n(x_k))+(x_k-x_n)\frac{\partial G}{\partial y_n}(x_n,t_n,y_N,_n(x_k))\right]\int_a^b y_N,_{n+1}(t)dt\\&+\left[\frac{\partial G}{\partial x_n}(x_n,t_n,y_N,_n(x_k))+\frac{\partial G}{\partial y_n}(x_n,t_n,y_N,_n(x_k))\right]\times\int_a^b(t-t_n+yN,_{n+1}-yN,_n)yN,_{n+1}(t)dt=g(x_k)+H_N(x_k)\end{aligned} \tag{21}$$

Where for some obvious practical reasons, we have chosen the collocation points to be

$$x_k=a+\frac{(b-a)k}{N+2}, k=1,2,3,\dots,N+1.$$

Thus, we have $(N+1)$ collocation equations in $(N+3)$ unknowns $(a_0,a_1,\dots,a_N,\ \tau_1$ and $\tau_2)$ constants to be determined.

Other extra equations are obtained from equation (18).

Altogether, we have a total of $(N+3)$ algebraic linear system of equations in $(N+3)$ unknown constants. The $(N+3)$ linear algebraic systems of equations are then solved by Gaussian elimination method to obtain the unknown constants $a_i(i\ge 0)$ which are then substituted back into the approximate solution given in equation (7).

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 9, November- 2012

STANDARD COLLOCATION METHOD

In this section, we discuss the collocation Tau numerical solution for the solutions of the linearized equation (6).

In this method, after the evaluation of the integrals in equation (6), equation (7) is substituted into the linearized equation (6), to obtain

$$P_n y^n_{N,n+1}(x)+\cdots+P_2 y''_{N\,n+1}(x)+P_1\left(f_n+f_{yn}y'(x)\right)y'_{N\,n+1}(x)+f_{yN,n}y'_n(x)y_{N\,n+1}(x)+f_n y_{N\,n}(x)y'_{N\,n}(x)$$

$$+\left[\frac{\partial G}{\partial y_n}(x_n,t_n,y_{N\,n})+(x-x_n)\frac{\partial G}{\partial y_n}(x_n,t_n,y_{N\,n})\right]\int_a^b y_{N\,n+1}(t)dt$$

$$+\left[\frac{\partial G}{\partial x_n}(x_n,t_n,y_{N\,n})+\frac{\partial G}{\partial y_n}(x_n,t_n,y_{N\,n})\right]\times\int_a^b (t-t_n)+\cdots+(y_{N\,n+1}-y_{N\,n})y_{N\,n+1}(t)dt$$

$$=g(x). \tag{22}$$

together with the boundary conditions

$$\begin{aligned} &y^{n-1}_{N,n+1}(a)+\cdots+y_{N\,n+1}(a)=\alpha\\ &y^{n-1}_{N,n+1}(b)+\cdots+y_{N\,n+1}(b)=\beta\\ &\vdots\\ &y^{n-1}_{N,n+1}(f)+\cdots+y_{N\,n+1}(f)=\chi \end{aligned} \tag{23}$$

Chebyshev polynomials of degree N valid in [a, b] and recurrence relation generated in equation (19) and (20) as above.

Thus, equation (22) is collocated at point $x=x_k$, hence, we get

$$P_n y^n_{N,n+1}(x_k)+\cdots+P_2 y''_{N\,n+1}(x_k)+P_1\left(f_n+f_{yn}y'(x_k)\right)y'_{N\,n+1}(x_k)+f_{yN,n}(x_k)y_{N,n+1}(x_k)$$

$$+f_n y_{N,n}(x_k)y'_{N,n+1}(x_k)+\left[\frac{\partial G}{\partial y_n}(x_n,t_n,y_{N,n}(x_k))+(x_k-x_n)\frac{\partial G}{\partial y_n}(x_n,t_n,y_{N,n}(x_k))\right]\int_a^b y_{N,n+1}(t)dt$$

$$+\left[\frac{\partial G}{\partial x_n}(x_n,t_n,y_{N,n}(x_k))+\frac{\partial G}{\partial y_n}(x_n,t_n,y_{N,n}(x_k))\right]\times\int_a^b (t-t_n+yN,_{n+1}-yN,_n)yN,_{n+1}(t)dt=g(x_k)$$

(24)

Where for some obvious practical reasons, we have chosen the collocation points to be

$$x_k = a + \frac{(b-a)k}{N}, k = 1,2,3,\ldots, N-1.$$

Thus, we have N collocation equations in $(N+1)$ unknowns $(a_0, a_1, \ldots, a_N)$ constants to be determined.

Other extra equations are obtained from equation (23).

Altogether, we have a total of $(N+1)$ algebraic linear system of equations in $(N+1)$ unknown constants. The $(N+1)$ linear algebraic systems of equations are then solved by Gaussian elimination method to obtain the unknown constants $a_i (i \geq 0)$ which are then substituted back into the approximate solution given in equation (7).

Remark: All the above procedures have been automated by the use of symbolic algebraic program MATLAB 7.9 and no manual computation is required at any stage.

Error Estimation

In this section, we perform the estimating error for the Integro-Differential Equations. Let us call $e_n(s) = y(s) - y_N(s)$ the error function of the Tau approximation $y_N(s)$ to y(s).

where y(s) is the exact solution of

$$Dy(s) + \lambda \int_a^b k(s,t)y(t)dt = f(s) \qquad s \in [a,b], \tag{25}$$

together with the condition

$$\sum_{k=1}^{n_d} [c_{jk}^1 y_n^{(k-i)}(a) + c_{jk}^2 y_n^{(k-i)}(b)] = d_j, \qquad j = 1, \ldots, n_d. \tag{26}$$

Therefore, $y_N(s)$ satisfies the problem

$$Dy_n(s) + \lambda \int_a^b k(s,t) y_n(t)dt = f(s) + H_n(s), \qquad s \in [a,b], \tag{27}$$

together with the condition

$$\sum_{k=1}^{n_d} [c_{jk}^1 y_n^{(k-i)}(a) + c_{jk}^2 y_n^{(k-i)}(b)] = d_j, \qquad j = 1, \ldots, n_d. \tag{28}$$

$H_n(s)$ is a perturbation term associated with $y_N(s)$ and can be obtained by substituting $y_N(s)$ into the equation

$$H_n(s) = Dy_n(s) + \lambda \int_a^b k(s,t) y_n(t) dt - f(s).$$

We proceed to find an approximation $e_{n,N}(s)$ to the $e_N(s)$ in the same way as we did before for the solution of problem (6). Subtracting equations (27) and (28) from (25) and (26), respectively, the error equation with the homogeneous condition is followed:

$$De_n(s) - \lambda \int_a^b k(s,t) e_n(t) dt = -H_n(s), \qquad s \in [a,b], \tag{29}$$

together with the condition

$$\sum_{k=1}^{n_d} [c_{jk}^1 y_n^{(k-i)}(a) + c_{jk}^2 y_n^{(k-i)}(b)] = d_j, \qquad j = 1, \ldots, n_d. \tag{30}$$

and solving this problem in the same way, we get the approximation $e_{n,N}(s)$. It should be noted that in order to construct the Tau approximation $e_{n,N}(s)$ to $e_{n,}(s)$, only the right -hand side of system (29) needs to be recomputed, the structure of the coefficient matrix G_n remains the same.

Numerical Examples

Numerical Experiments and Discussion

In this section, we present numerical results obtained with that obtained by Behiry and Mohammed [2], that considered these problems stated below as test problems and the problems are of orders 5, 6 and 8 nonlinear integro-differential equations. We present tables of exact solutions, results of methods used and the results obtained by Behiry and Mohammed [2] for different values of the approximants.

Example 1: Consider the nonlinear integro-differential equation.

$$x^3 y^{(5)}(x) - 2y'(x) + xy(x) = x^4 + 5x^3 - 4e^x - \frac{(e^2-1)}{4} x + 2 + \int_0^x ty(t)dt + \int_0^1 xy^2(t)dt, \; 0 \le x \le 1$$

together with the following initial conditions.

$$y(0) = 0, \; y'(0) = 1, \; y''(0) = 2, \; y'''(0) = 3 \text{ and } y^4(0) = 4.$$

The exact solution is given as $y(x) = xe^x$. For favorable comparison, we have chosen our initial guess $y_{N,k} = xe^x$. Here k is the number of iterations in the new method and N is the degree of approximant used.

Table 1a: Table of solution for example $1(k=5, N=5)$

X	Exact value	Standard Collocation method	Perturbed Collocation Method	Result Obtained by Behiry and Mohamed [2]
0.0	0.00000000	0.00000000	0.00000000	0.00000000
0.1	0.1105170918	0.1105171045	0.1105170820	0.1105170918
0.2	0.2442805516	0.2442806261	0.2442805427	0.2442805516
0.3	0.4049576423	0.4049577205	0.4049576451	0.4049576423
0.4	0.5967298791	0.5967299147	0.5967298684	0.5967298791
0.5	0.8243606354	0.8243607764	0.8243606246	0.8243606354
0.6	1.0932712800	1.0932738510	1.093272540	1.0932712800
0.7	1.409626895	1.409627468	1.409626860	1.409626895
0.8	1.780432743	1.780433065	1.780432654	1.780432743
0.9	2.213642800	2.213644521	2.213643411	2.213642800
1.0	2.718281828	2.718281831	2.718281825	2.718281828

Table 1b: Table of error for example 1

X	Standard Collocation method	Perturbed Collocation method	Result Obtained by Behiry and Mohamed [2]
0	0.0000000E+00	0.0000000E+00	0.0000000E+00
0.1	1.2700000E-08	9.8000000E-09	1.2700000E-08
0.2	7.4500000E-08	8.9000000E-09	7.4500000E-08
0.3	7.8200000E-08	2.8000000E-09	7.8200000E-08
0.4	3.5600000E-08	1.0700000E-08	3.5600000E-08
0.5	1.4100000E-07	1.0800000E-08	1.4100000E-07
0.6	2.5710000E-06	1.2600000E-06	2.5710000E-06
0.7	5.7300000E-07	3.5000000E-08	5.7300000E-07
0.8	3.2200000E-07	8.9000000E-08	3.2200000E-07
0.9	1.7210000E-06	6.1100000E-07	1.7210000E-06
1.0	3.0000002E-09	2.9999998E-09	3.0000002E-09

Example 2: Consider the nonlinear integro-differential equation

$$x^4 y^{(6)}(x) + y^{(3)}(x) + y'(x) = -x^4 \cos x + 0.5\sin 2x + 3x + 0.4$$
$$-0.1e\{[\cos(1)+\sin(1)]\times[\cos^2(1)+3e]\} - 2\int_0^x [1+y^2(t)dt + \int_0^1 e^t y^3(t)dt$$

together with the following initial conditions.

$$y(0)=1,\ \ y'(0)=0,\ \ y''(0)=-1,\ \ y'''(0)=0,\ \ y^{(4)}(0)=1 \text{ and } y^{(5)}(0)=0\,.$$

The exact solution is $y(x)=Cosx.$ the results of applying above methods with initial guess $y_{N,k} = 1 - x$ are given as follows (k denotes the number of iterations in new method and N the degree of approximant used).

Table 2a: Table of solution for example $2(k=5, N=5)$

X	Exact value	Standard Collocation method	Perturbed Collocation Method	Result Obtained by Behiry and Mohamed [2]
0.0	1.00000000	1.00000000	1.00000000	1.00000000
0.1	0.9950041653	0.9950042014	0.9950041476	0.9950041653
0.2	0.9800665778	0.9800666500	0.9800665694	0.9800665778
0.3	0.9553364891	0.9553365047	0.9553364781	0.9553364891
0.4	0.921060994	0.921061086	0.921060852	0.921060994
0.5	0.8775825619	0.8775826457	0.8775825569	0.8775825619
0.6	0.8253356149	0.8253357162	0.8253356046	0.8253356149
0.7	0.7648421873	0.7648422641	0.7648421764	0.7648421873
0.8	0.6967067093	0.6967067270	0.6967066982	0.6967067093
0.9	0.6216099683	0.6216010098	0.6216099512	0.6216099683
1.0	0.5403023059	0.5403023562	0.5403023056	0.5403023059

Table 2b: Table of error for example 2

X	Standard Collocation method	Perturbed Collocation method	Result Obtained by Behiry and Mohamed [2]
0	0.0000000E+00	0.0000000E+00	0.0000000E+00
0.1	3.6100000E-08	1.7700000E-08	3.6100000E-08
0.2	7.2200000E-08	8.3999999E-09	7.2200000E-08
0.3	1.5600000E-08	1.1000000E-08	1.5600000E-08
0.4	9.2000000E-08	1.4200000E-07	9.2000000E-08
0.5	8.3800000E-08	5.0000000E-09	8.3800000E-08
0.6	1.0130000E-07	1.0300000E-08	1.0130000E-07
0.7	7.6800000E-08	1.0900000E-08	7.6800000E-08
0.8	1.7700000E-08	1.1100000E-08	1.7700000E-08
0.9	8.9585000E-06	1.7100000E-08	8.9585000E-06
1.0	5.0300002E-08	2.9999991E-10	5.0300000E-08

Example 3. Consider the nonlinear Volterra-Fredholm integro-differential equation

$$y^{(8)}(x)-\pi^{8}y(x)=\frac{x}{2}-\int_{0}^{x}y^{2}(t)dt+\frac{\sin(2\pi x)}{2\pi}\int_{0}^{1}\big(\cos(\pi t)-y(t)\big)y(t)dt,\ \ 0\le x\le 1$$

With the initial conditions

$$y(0)=0,\quad y'(0)=\pi,\quad y''(0)=0,\quad y^{(3)}(0)=-\pi^{3},\quad y^{(4)}(0)=0,$$
$$y^{(5)}(0)=\pi^{5},\quad y^{(6)}(0)=0\ \ \textit{and}\ \ y^{(7)}(0)=-\pi^{7}.$$

The exact solution is $y(x)=\sin(\pi x)$. the results of applying above methods with initial guess $y_{N,k}=\dfrac{\sin(2\pi x)}{2\pi}$ are given as follows (k denotes the number of iterations in new method and *N* the degree of approximant used).

Table 3a: Table of solution for example $3(k=5, N=5)$

X	Exact value	Standard Collocation method	Perturbed Collocation Method	Result Obtained by Behiry and Mohamed [2]
0.0	0.00000000	0.000000000	0.000000000	0.000000000
0.1	0.3090169944	0.3090169992	0.3090169849	0.3090169944
0.2	0.5877852523	0.5877852684	0.5877852498	0.5877852523
0.3	0.8090169944	0.8090169989	0.8090169870	0.8090169944
0.4	0.9510565163	0.9510565248	0.9510565041	0.9510565163
0.5	1.0000000000	1.0000000000	1.0000000000	1.0000000000
0.6	0.9510565163	0.9510565248	0.9510565041	0.9510565163
0.7	0.8090169944	0.8090169989	0.8090169870	0.8090169944
0.8	0.5877852523	0.5877852684	0.5877852498	0.5877852523
0.9	0.3090169944	0.3090169992	0.3090169849	0.3090169944
1.0	0.0000000000	0.0000000000	0.0000000000	0.0000000000

Table 3b: Table of error for example 3

X	Standard Collocation method	Perturbed Collocation method	Result Obtained by Behiry and Mohamed [2]
0	0.0000000E+00	0.0000000E+00	0.0000000E+00
0.1	4.8000000E-09	9.5000000E-09	4.8000000E-09
0.2	1.6100000E-08	2.5000000E-09	1.6100000E-08
0.3	4.5000000E-09	7.3999999E-09	4.5000000E-09
0.4	8.5000000E-09	1.2200000E-08	8.5000000E-09
0.5	0.0000000E+00	0.0000000E+00	0.0000000E+00
0.6	8.5000000E-09	1.2200000E-08	8.5000000E-09
0.7	4.5000000E-09	7.3999999E-09	4.5000000E-09
0.8	1.6100000E-08	2.5000000E-09	1.6100000E-08
0.9	4.8000000E-09	9.5000000E-09	4.8000000E-09
1.0	0.0000000E+00	0.0000000E+00	0.0000000E+00

Conclusion

Higher-order nonlinear integro-differential equations are usually difficult to solve analytically. In many cases, it is required to obtain the approximate solutions. For this purpose, the presented methods can be proposed. A considerable advantage of the methods is achieved as different approximate solutions are obtained by different values of N. Furthermore, after calculation of the approximate solutions, the approximate solution $y_N(x)$ can be easily evaluated for arbitrary values of x at low computation effort.

To get the best approximating solution of the equation, N (the degree of the approximating polynomial) must be chosen large enough. From the tabular points shown in Table 1, it is observed that the solution found for N=10 shows close agreement for various values of x. In particular, the solution of example 3, for N=10 shows a very close approximation to the analytical solution at the points in interval $0 \le x \le 1$. An interesting feature of the Standard and Perturbed collocation methods is that we get an analytical solution in many cases, as demonstrated in examples 1, 2 and 3.

ISSN: 2278-0181
Vol. 1 Issue 9, November- 2012

Reference

1. Asady, B. and Kajani M. T. (2005); Direct method for solving integro-differential equations using hybrid Fourier and block-pulse functions, Int. J. Comput. Math. 82, 889-895

2. Behiry, S. H. and Mohamed S. I.(2012); Solving high-order nonlinear Volterra- Fredholm integro-differential equations by differential transform method, Natural Science Vol. 4,(8), 581-587

3. Han Danfu and Shang Xufeng(2007); Numerical solution of integro- differential equations by using CAS wavelet operational matrix of integration, Appl. Math. Comput, 194(2), 460-466.

4. Rahimi-Ardabili, M. Y. and Shahmorad, S.(2007); Iterative numerical solution of non-linear integro-differential equations by Tau method, Appl. Math. Comput. 193(2), 514-522.

5. Shahmorad, S. Ivaz, K. Mostahkam, B. S.(2009); Newton-Tau numerical solution of one dimensional non-linear integro-differential equations, Southeast Asian Bull. Math. 33(4), 733-740.

6. Shu-Qiang Wang and Ji-Huan He(2007); Variational iteration method for solving integro-differential equations Phys. Lett. A 367(3), 188-191.

7. Taiwo, O. A. (2004);The application of cubic spline collocation Tau methods for the solution of second order nonlinear boundary value problems, J. Nigerian Math. Soc. 23, 27-40

8. Taiwo, O. A.(2005); Comparison of collocation methods for the solution of second order non-linear boundary value problems, Int. J. Comput. Math. 82(11), 1389-1401.

9. Taiwo, O. A. (2007); Comparison of two collocation methods for the solution of integral eqautions by cubic spline, Journal of Mathematical Association of Nigeria, (ABACUS) 34(2B), 289-294.

10. Taiwo, O. A. and Evans, D. J.(1997); Collocation approximation for fourth-order boundary value problems, Int. J. Comput. Math. 63(1-2), 57-66.

11. Taiwo O. A. and Ishola C. Y.(2009); Collocation approximation methods for the numerical solution of integral equations, Int. J. Pure Appl. Sci. 2, 29-37.

12. Tavassoli Kajani, M.,Ghasemi, M. and Babolian, E.(2006); Numerical solution of linear integro-differential equation by using sine-cosine wavelets, appl. Math. Comput. 180(2), 569-574.

13. Zhao, J. and Corless R. M. (2006); Compact finite difference method for integro-differential equations, Appl. Math. Comput. 177(1), 271-288.