*Bulletin of the Iranian Mathematical Society Vol.* 32 *No. 1 (2006), pp 33-44.*

# TAU NUMERICAL SOLUTION OF VOLTERRA INTEGRO-DIFFERENTIAL EQUATIONS WITH ARBITRARY POLYNOMIAL BASES

J. POUR-MAHMOUD*, M. Y. RAHIMI AND S. SHAHMORAD

Communicated by Mohammad Asadzadeh

Abstract. The purpose of this paper is to investigate the Tau method with arbitrary polynomial basis which is developed to find numerical solutions of the Volterra integro-differential equations (VIDEs). The differential and integral parts appearing in these equations are replaced by Tau operational representation to convert VIDEs to a system of linear algebraic equations. Some numerical results are given to demonstrate the superior performance of the Tau method, particularly, with the Chebyshev bases.

## 1. **Introduction**

Let us consider the general form of linear Volterra integro-differential equation:

$$Dy(x) - \lambda \int_a^x k(x,t)y(t)dt = f(x), \quad x \in [a,b] \tag{1.1}$$

with $n_d$ independent conditions

MSC(2000): 65R20
Keywords: Tau method, Integral and Integro-differential equations
Received: 13 June 2006

*Corresponding author

$$\sum_{s=1}^{n_d}[c_{js}^{(1)}y^{(s-1)}(a)+c_{js}^{(2)}y^{(s-1)}(x_0)]=d_j, \quad j=1,2,\cdots,n_d \tag{1.2}$$

where $f(x)$ and $k(x,t)$ are given continuous functions and $\lambda$, $a$, $c_{js}^{(1)}$, $c_{js}^{(1)}$, $x_0 \in [a,b]$ are given constants and $n_d$ is the order of the differential operator $D$ with polynomial coefficients $p_i(x)$:

$$D=\sum_{i=0}^{n_d}p_i(x)\frac{d^i}{dx^i}, \quad p_i(x)=\sum_{j=0}^{\alpha_i}p_{ij}x^j \tag{1.3}$$

while $\alpha_i$ is the degree of $p_i(x)$.

Hosseini and Shahmorad [1, 2, 3] developed the Tau method to find numerical solution of the Fredholm, Volterra and Fredholm-Volterra integro-differential equations with the standard bases $\underline{X}=[1,x,\cdots]^T$ and Fredholm integro-differential equations (FIDEs), in particular, with arbitrary polynomial bases.

The subject of this paper is to present developments of the operational Tau method (see [5] and [6]) with arbitrary polynomial bases for the numerical solution of Volterra integro-differential equations (VIDEs).

Details of the structure of the employed arbitrary bases in the operational approach to the Tau method is explained in section 2. In section 3, an efficient Tau error estimator is introduced. In section 4, preliminary steps towards application of the Chebyshev bases are taken. Finally, in section 5, some numerical results are provided to demonstrate the efficiency of using Chebyshev bases compared with the results obtained by other methods.

## 2. Matrix representation

### 2.1. Differential part and supplementary conditions.

Let $\underline{V}=[v_0(x),v_1(x),\cdots]^T$ be an orthogonal polynomial basis vector given by $\underline{V}=V\underline{X}$, where $V$ is a non-singular lower triangular matrix and deg $v_i(x)\le i$ , $i=0,1,2,\cdots$ .

We recall the following relations from [4]:

(i) Differential part:

$$Dy(x)=\underline{a}\Pi_v\underline{V}, \tag{2.1}$$

where

$$\underline{a} = [a_0, a_1, \cdots], \quad \Pi_v = V \Pi V^{-1}, \quad \Pi = \sum_{i=0}^{n_d} \eta^i p_i(\mu),$$

and

$$\mu = \begin{bmatrix} 0 & 1 & 0 & 0 & \\ & 0 & 1 & 0 & \\ & & 0 & 1 & \vdots \\ & & & 0 & \\ & & \cdots & & \ddots \end{bmatrix}, \qquad \eta = \begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ 0 & 2 & 0 & & \vdots \\ 0 & 0 & 3 & 0 & \\ & & \cdots & & \ddots \end{bmatrix}.$$

(ii) Supplementary conditions:

$$\underline{a} B = \underline{d}, \tag{2.2}$$

where $\underline{d} = [d_1, d_2, \cdots, d_{n_d}]$ is the vector consisting of the right hand sides of the conditions and the entries of the matrix $B$ are obtained by

$$b_{ij} = \sum_{s=1}^{n_d} [c_{js}^{(1)} v_{i-1}^{(s-1)}(a) + c_{js}^{(2)} v_{i-1}^{(s-1)}(x_0)], \quad i, j = 1, 2, \cdots, n_d, \tag{2.3}$$

and

$$b_{ij} = \sum_{s=1}^{n_d} [c_{js}^{(1)} v_{i-1}^{(s-1)}(a) + c_{js}^{(2)} v_{i-1}^{(s-1)}(x_0)], \\ i = n_d + 1, n_d + 2, \cdots, \quad j = 1, 2, \cdots, n_d. \tag{2.4}$$

2.2. **Integral part.** Consider the expansions of $k(x,t)$ and $y(x)$

$$k(x,t) = \sum_{i=0}^{n} \sum_{j=0}^{n} k_{ij} v_i(x) v_j(t) \quad , \quad y(x) = \sum_{i=0}^{\infty} a_i v_i(x) = \underline{a}\, \underline{V},$$

where $k_{ij}$ are constant coefficients and $v_i(x), \quad i = 0, 1, \cdots$, the entries of $\underline{V}$. So the integral part of (1.1) can be written as

$$\int_a^x k(x,t) y(t)\, dt = \sum_{s=0}^{\infty} \sum_{i=0}^{n} \sum_{j=0}^{n} k_{ij} a_s v_i(x) \int_a^x v_j(t) v_s(t)\, dt.$$

We set

$$v_{ijs}(x) = v_i(x) \textstyle\int_a^x v_j(t) v_s(t) dt, \qquad \begin{array}{l} i = 0, 1, 2, \cdots, n, \\ j = 0, 1, 2, \cdots, n, \\ s = 0, 1, 2, \cdots, \end{array}$$

and write it in terms of the basis functions:

$$v_{ijs}(x) = \sum_{m=0}^{3n+1} c_{mijs} v_m(x), \tag{2.5}$$

where $c_{mijs}$ are constant coefficients that must be determined. We multiply both sides of (2.5) by

$$w(x)v_r(x) \ , \quad r = 0, 1, \cdots, 3n+1,$$

where $w(x)$ is the weight function. Integrating on [a, b], we get

$$c_{mijs} = \frac{1}{\alpha_m} \int_a^b w(x) v_m(x) v_{ijs}(x) dx, \qquad \begin{array}{l} m = 0, 1, 2, \cdots, 3n+1 \\ i = 0, 1, 2, \cdots, n \\ j = 0, 1, 2, \cdots, n \\ s = 0, 1, 2, \cdots \ , \end{array} \tag{2.6}$$

with $\alpha_m = \int_a^b w(x) v_m^2(x) dx$, $m = 0, 1, \cdots, 3n+1$, so that

$$\int_a^x k(x,t) y(t) dt = \sum_{s=0}^{\infty} \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{m=0}^{3n+1} k_{ij} a_s c_{mijs} v_s(x),$$

which can be written as:

$$\int_a^x k(x,t) y(t) dt = \underline{a} \ \underline{K} \ \underline{V}, \tag{2.7}$$

with the vector of unknown coefficients $\underline{a} = [a_0, a_1, a_2, \cdots]$, the basis vector $\underline{V}$ and the constant matrix $\underline{K}$ with entries:

$$K_{ms} = \sum_{i=0}^{n} \sum_{j=0}^{n} k_{ij} c_{mijs}, \quad m, n = 0, 1, 2, \cdots \ .$$

**Remark 2.1.** The integral part converts to matrix form using equation (2.7).

2.3. **Converting IDE to matrix.** Let $f(x) = \sum_{i=0}^{n} f_i v_i(x) = \underline{f} \ \underline{V}$ with the coefficient vector $\underline{f} = (f_0, f_1, \cdots, f_n, 0, 0, \cdots)$. Considering the equations (2.1), (2.2) and (2.7) we have

$$\begin{array}{rl} \underline{a}\Pi_v \underline{V} - \lambda \underline{a} \ \underline{K} \ \underline{V} & = \underline{f} \ \underline{V}, \\ \underline{a}B & = \underline{d}, \end{array}$$

instead of (1.1), (1.2). So

$$\underline{a}(\Pi_v - \lambda \underline{K}) = \underline{f},\ \underline{a}B = \underline{d}, \tag{2.8}$$

because $\underline{V}$ is a basis vector. Now let

$$\begin{aligned} \widehat{K} &= \Pi_v - \lambda \underline{K}, \\ G &= [B_1, \cdots, B_{n_d}, \widehat{K}_1, \widehat{K}_2, \cdots], \\ g &= [d_1, \cdots, d_{n_d}, f_0, f_1, \cdots]. \end{aligned}$$

Furthermore, let $B$ be the matrix representation of the supplementary conditions with $j^{th}$ column $B_j$, and let $\widehat{K}_1$ ,$\widehat{K}_2$ , $\cdots$ be the columns of the matrix $\widehat{K}$. So (2.8) can be replaced by the equation

$$\underline{a}G = g. \tag{2.9}$$

**Definition 2.2.** The polynomial $y_n(x) = \underline{a}_n \underline{V}$ will be called an approximate solution of ( 1.1 ) and ( 1.2 ), if the vector $\underline{a}_n = [a_0, a_1, \cdots, a_n]$ is the solution of the system of linear algebraic equations

$$\underline{a}_n G_n = g_n, \tag{2.10}$$

where $G_n$ is the matrix defined by restriction of $G$ to its first $n+1$ rows and columns.

**Remark 2.3.** For $n_d = 0$ and $p_0(x) = 1$, the equation ( 1.1 ) is transformed into a Volterra integral equation of second kind and for $\lambda = 0$, it is transformed into a differential equation.

## 3. Error estimation

In this section an error estimator for the approximate solution of (1.1) and (1.2) is obtained. Let us call $e_n(x) = y(x) - y_n(x)$ as the error function of the approximate solution $y_n(x)$ to $y(x)$, where $y(x)$ is the exact solution of (1.1) and (1.2). Hence $y_n(x)$ satisfies the following equation:

$$Dy_n(x) - \lambda \int_a^x k(x,t) y_n(t) dt = f(x) + H_n(x), \quad x \in [a,b], \tag{3.1}$$

with $n_d$ independent conditions

$$\sum_{s=1}^{n_d}[c_{js}^{(1)}y_n^{(s-1)}(a)+c_{js}^{(2)}y_n^{(s-1)}(x_0)]=d_j, \quad j=1,2,\cdots,n_d. \tag{3.2}$$

The perturbation term $H_n(x)$ can be obtained by substituting the computed solution $y_n(x)$ into the equation

$$H_n(x)=Dy_n(x)-\lambda\int_a^x k(x,t)y_n(t)dt-f(x). \tag{3.3}$$

We proceed to find an approximation $e_{n,N}(x)$ to the error function $e_n(x)$ in the same way as we did before for the solution of equations (1.1) and (1.2). Here N denotes the Tau degree of $e_n(x)$. Subtracting (3.1) and (3.2) from (1.1) and (1.2) respectively, the error function $e_n(x)$ satisfies the equation

$$De_n(x)-\lambda\int_a^x k(x,t)e_n(t)dt=-H_n(x), \quad x\in[a,b], \tag{3.4}$$

with $n_d$ homogeneous conditions

$$\sum_{s=1}^{n_d}[c_{js}^{(1)}e_n^{(s-1)}(a)+c_{js}^{(2)}e_n^{(s-1)}(x_0)]=0, \quad j=1,2,\cdots,n_d. \tag{3.5}$$

It should be noted that in order to construct the approximation $e_{n,N}(x)$ for $e_n(x)$, we only need to re-compute the right hand side of the system (2.10). In fact the structure of the coefficient matrix $G_n$ remains the same.

## 4. Application on the Chebyshev basis

In section 2, we considered $\underline{V}=[v_0(x),v_1(x),\cdots]^T$ as a basis vector with polynomial entries where deg $(v_i(x))\le i, \quad for \quad i=0,1,2,\cdots$. It was used for converting (1.1) and (1.2) into a system of linear equations. The shifted Chebyshev polynomials are interesting polynomial bases with a matrix $V$ of the same structure. We pursue the application of the method for the case of Chebyshev polynomials.

The shifted Chebyshev polynomials are defined as

$$T_0^*(x)=1, \quad T_1^*(x)=\frac{2x-(b+a)}{b-a}, \quad x\in[a,b]$$

and, for $i \geq 1$, as

$$T^*_{i+1}(x) = 2(\frac{2x-(b+a)}{b-a})T^*_i(x) - T^*_{i-1}(x), \quad x \in [a,b].$$

In this case the functions $y_n(x)$, $f(x)$ and $k(x,t)$ are written as

$$y_n(x) = \sum_{i=0}^{n} a_i\, T^*_i(x), \quad f(x) = \sum_{i=0}^{n}{}'' f_i\, T^*_i(x),$$

$$k(x,t) = \sum_{i=0}^{n}{}'' \sum_{j=0}^{n}{}'' k_{ij}\, T^*_i(x)\, T^*_j(t),$$

where the symbol $('')$ over $\sum$ indicates that the first and the last terms must be divided by 2. Here $a_0, a_1, \cdots, a_n$ are obtained from (2.10) and $f_i$ , $k_{ij}$ are given by the following relations:

$$f_i = (\frac{2}{n}) \sum_{s=0}^{n}{}'' f(x_s) \cos(\frac{is\pi}{n}), \quad i = 0,1,2,\cdots,n,$$

$$k_{ij} = (\frac{4}{n^2}) \sum_{r=0}^{n}{}'' \sum_{s=0}^{n}{}'' k(x_s,x_r) \cos(\frac{is\pi}{n}) \cos(\frac{jr\pi}{n}), \quad i,j = 0,1,2,\cdots,n,$$

with

$$x_s = \frac{1}{2}\left[(b-a)\cos\left(\frac{s\pi}{n}\right) + (b+a)\right], \quad s = 0,1,\cdots,n,$$

and $c_{mijs}$ in (2.5) are easily computed for $m = 0,1,\cdots,3n+1$, $i = 0,1,\cdots,n$, $j = 0,1,\cdots,n$, $s = 0,1,\cdots$ as follows:

$$c_{mijs} = \left(\frac{b-a}{8}\right)\left[\frac{1}{2}(z_1+z_2+z_3+z_4) - (\gamma_1+\gamma_2+\gamma_3+\gamma_4)\beta_{m,9}\right],$$

with

$$z_1 = \begin{cases} \frac{1}{j+s+1}(\beta_{m,1}+\beta_{m,2}) & if \quad j+s+1 \neq 0 \\ 0 & otherwise, \end{cases}$$

$$z_2 = \begin{cases} \frac{-1}{j+s-1}(\beta_{m,3}+\beta_{m,4}) & if \quad j+s-1 \neq 0 \\ 0 & otherwise, \end{cases}$$

$$z_3 = \begin{cases} \frac{1}{j-s+1}(\beta_{m,5}+\beta_{m,6}) & if \quad j-s+1 \neq 0 \\ 0 & otherwise, \end{cases}$$

$$z_4 = \begin{cases} \frac{-1}{j-s-1}(\beta_{m,7}+\beta_{m,8}) & if \quad j-s-1 \neq 0 \\ 0 & otherwise, \end{cases}$$

$$\gamma_1 = \begin{cases} \frac{(-1)^{i+j+1}}{j+s+1} & if \quad j+s+1 \neq 0 \\ 0 & otherwise, \end{cases}$$

$$\gamma_2 = \begin{cases} \frac{(-1)^{j+s}}{j+s-1} & if \quad j+s-1 \neq 0 \\ 0 & otherwise, \end{cases}$$

$$\gamma_3 = \begin{cases} \frac{(-1)^{j-s+1}}{j-s+1} & if \quad j-s+1 \neq 0 \\ 0 & otherwise, \end{cases}$$

$$\gamma_4 = \begin{cases} \frac{(-1)^{j-s}}{j-s-1} & if \quad j-s-1 \neq 0 \\ 0 & otherwise, \end{cases}$$

where for $m = 0, 1, \cdots, 3n+1$ we have

$$\beta_{m,1} = \begin{cases} 1 & if \quad m = |i+j+s+1| \\ 0 & otherwise, \end{cases}$$

$$\beta_{m,2} = \begin{cases} 1 & if \quad m = |-i+j+s+1| \\ 0 & otherwise, \end{cases}$$

$$\beta_{m,3} = \begin{cases} 1 & if \quad m = |i+j+s-1| \\ 0 & otherwise, \end{cases}$$

$$\beta_{m,4} = \begin{cases} 1 & if \quad m = |-i+j+s-1| \\ 0 & otherwise, \end{cases}$$

$$\beta_{m,5} = \begin{cases} 1 & if \quad m = |i+j-s+1| \\ 0 & \quad otherwise, \end{cases}$$

$$\beta_{m,6} = \begin{cases} 1 & if \quad m = |-i+j-s+1| \\ 0 & \quad otherwise, \end{cases}$$

$$\beta_{m,7} = \begin{cases} 1 & if \quad m = |i+j-s-1| \\ 0 & \quad otherwise, \end{cases}$$

$$\beta_{m,8} = \begin{cases} 1 & if \quad m = |-i+j-s-1| \\ 0 & \quad otherwise, \end{cases}$$

$$\beta_{m,9} = \begin{cases} 1 & if \quad m = i \\ 0 & \quad otherwise. \end{cases}$$

## 5. **Numerical examples**

In this section we consider some examples demonstrating the accuracy of the method and effectiveness of the Chebyshev basis compared with the standard basis $\underline{X}$.

**Example 5.1.**(Constructed)

$$\begin{aligned} &y''(x) + (x-2)y'(x) + 4x^2y(x) + 4\int_0^x (x-1)^2 ty(t)dt = 2(x-1)^2, \\ &y(0) = -2, \\ &y'(0) = 0. \end{aligned}$$

The exact solution is $y(x) = \sin(x^2) - 2\cos(x^2)$. For numerical results and comparison with the exact solution see Table 1.

**Example 5.2.** [ See [6, 3], Example (b) ]

$$\begin{aligned} &y'(x) = 1 + 2x - y(x) + \int_0^x x(1+2x)e^{t(x-t)}y(t)dt \ , \quad 0 \le x \le 1, \\ &y(0) = 1. \end{aligned}$$

The exact solution is $y(x) = e^{x^2}$. For numerical results and comparison with exact solution see Table 2.

In the following two tables the terms $(Exact)$, $(Tau)$, $(Tau.Err)$, $(Shahmo.Err)$, $(Markro.Err)$ are provided at the selected points of the given interval for the exact solution, Tau approximate solution, their absolute error $|y(s) - y_n(s)|$ , estimation error by the Tau method in Chebysheve basis, the estimation error of the Tau method obtained by Shahmorad [4] in the standard basis $\underline{X}$ and the absolute error of the Makroglou method [6], respectively.

Table 1

| $x$ | $Exact$ | $Tau$ | $Tau.Err$ | $Est.Err$ |
|---|---|---|---|---|
| $n = 5$ | | | | |
| 0.00 | $-2.00000000$ | $-2.00000000$ | $2.00e-09$ | $4.00e-12$ |
| .20 | $-1.95841088$ | $-1.95909953$ | $6.89e-04$ | $6.88e-04$ |
| .40 | $-1.81513636$ | $-1.82041079$ | $5.27e-03$ | $5.27e-03$ |
| .60 | $-1.51951941$ | $-1.52774678$ | $8.23e-03$ | $8.21e-03$ |
| .80 | $-1.00699608$ | $-1.01541428$ | $8.42e-03$ | $7.96e-03$ |
| 1.00 | $-.23913363$ | $-.25000787$ | $1.09e-02$ | $5.34e-03$ |
| $n = 10$ | | | | |
| 0.00 | $-2.00000000$ | $-2.00000000$ | $2.00e-09$ | $2.47e-14$ |
| .20 | $-1.95841088$ | $-1.95841088$ | $0.00e-01$ | $1.20e-09$ |
| .40 | $-1.81513636$ | $-1.81513636$ | $2.00e-09$ | $5.40e-09$ |
| .60 | $-1.51951941$ | $-1.51951942$ | $7.00e-09$ | $1.85e-08$ |
| .80 | $-1.00699608$ | $-1.00699607$ | $3.00e-09$ | $1.86e-06$ |
| 1.00 | $-.23913363$ | $-.23913363$ | $3.90e-09$ | $6.45e-05$ |
| $n = 15$ | | | | |
| 0.00 | $-2.00000000$ | $-2.00000000$ | $2.00e-09$ | $1.00e-17$ |
| .20 | $-1.95841088$ | $-1.95841088$ | $2.00e-09$ | $6.49e-11$ |
| .40 | $-1.81513636$ | $-1.81513636$ | $2.00e-09$ | $6.55e-11$ |
| .60 | $-1.51951941$ | $-1.51951942$ | $1.00e-09$ | $3.80e-10$ |
| .80 | $-1.00699608$ | $-1.00699607$ | $1.00e-09$ | $1.70e-09$ |
| 1.00 | $-.23913363$ | $-.23913363$ | $2.00e-10$ | $2.46e-07$ |

*Numerical results*

Table 1; Numerical results of Example 1 in the Chebyshev basis and comparison with the numerical results of the standard basis for n = 5, 10, 15.

Table 2

| $x$ | *Exact* | *Tau* | $Tau \cdot Err$ | $Est \cdot Err$ | $Shahmo.Err$ | $Makro.Err$ |
|---|---|---|---|---|---|---|
| $n = 2$ | | | | | | |
| 0 | 1.0 | 1.0 | 0.0 | 0.0 | | |
| 0.200 | 1.04081 | 0.990415 | $5.0395 \times 10^{-2}$ | $5.7551 \times 10^{-2}$ | | |
| 0.400 | 1.17351 | 1.18107 | $7.56 \times 10^{-3}$ | $8.6335 \times 10^{-3}$ | | |
| 0.600 | 1.43333 | 1.57196 | $1.3863 \times 10^{-1}$ | $1.5831 \times 10^{-1}$ | | |
| 0.800 | 1.89648 | 2.16309 | $2.6661 \times 10^{-1}$ | $3.0446 \times 10^{-1}$ | | |
| 1.0 | 2.71828 | 2.95446 | $2.3618 \times 10^{-1}$ | $2.6971 \times 10^{-1}$ | | |
| $n = 5$ | | | | | | |
| 0 | 1.0 | 1.0 | 0.0 | 0.0 | | |
| 0.200 | 1.04081 | 1.04081 | $7.2 \times 10^{-7}$ | $8.2224 \times 10^{-7}$ | | |
| 0.400 | 1.17351 | 1.17351 | $3.2 \times 10^{-8}$ | $3.6544 \times 10^{-8}$ | | |
| 0.600 | 1.43333 | 1.43337 | $4.5 \times 10^{-7}$ | $5.1390 \times 10^{-7}$ | | |
| 0.800 | 1.89648 | 1.89642 | $6 \times 10^{-7}$ | $9.8212 \times 10^{-7}$ | | |
| 1.0 | 2.71828 | 2.71840 | $1.2 \times 10^{-5}$ | $1.3704 \times 10^{-5}$ | | |
| $n = 10$ | | | | | | |
| 0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | |
| 0.200 | 1.04081 | 1.04081 | $1.001 \times 10^{-11}$ | $1.114 \times 10^{-10}$ | $1.153 \times 10^{-9}$ | $3.63 \times 10^{-8}$ |
| 0.400 | 1.17351 | 1.17351 | $1.001 \times 10^{-10}$ | $1.112 \times 10^{-10}$ | $2.331 \times 10^{-8}$ | $1.60 \times 10^{-7}$ |
| 0.600 | 1.43333 | 1.43333 | $1.001 \times 10^{-8}$ | $1.112 \times 10^{-8}$ | $2.223 \times 10^{-6}$ | $4.45 \times 10^{-7}$ |
| 0.800 | 1.89648 | 1.89648 | $1.100 \times 10^{-7}$ | $1.113 \times 10^{-6}$ | $1.244 \times 10^{-4}$ | $1.11 \times 10^{-6}$ |
| 1.0 | 2.71828 | 2.71828 | $1.2 \times 10^{-6}$ | $1.284 \times 10^{-6}$ | $1.838 \times 10^{-3}$ | $2.75 \times 10^{-6}$ |
| $n = 15$ | | | | | | |
| 0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | |
| 0.200 | 1.04081 | 1.04081 | $7.0000 \times 10^{-11}$ | $7.994 \times 10^{-10}$ | $1.489 \times 10^{-9}$ | |
| 0.400 | 1.17351 | 1.17351 | $2.100 \times 10^{-11}$ | $2.238 \times 10^{-10}$ | $2.591 \times 10^{-9}$ | |
| 0.600 | 1.43333 | 1.43333 | $6.100 \times 10^{-10}$ | $6.962 \times 10^{-10}$ | $9.087 \times 10^{-9}$ | |
| 0.800 | 1.89648 | 1.89648 | $1.100 \times 10^{-10}$ | $1.142 \times 10^{-9}$ | $7.984 \times 10^{-7}$ | |
| 1.0 | 2.71828 | 2.71828 | $2.570 \times 10^{-8}$ | $2.93494 \times 10^{-7}$ | $2.880 \times 10^{-5}$ | |
| $n = 20$ | | | | | | |
| 0 | 1.0 | 1.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.200 | 1.04081 | 1.04081 | $1.540 \times 10^{-12}$ | $1.668 \times 10^{-11}$ | $1.123 \times 10^{-9}$ | $2.28 \times 10^{-9}$ |
| 0.400 | 1.17351 | 1.17351 | $1.571 \times 10^{-12}$ | $1.528 \times 10^{-11}$ | $1.439 \times 10^{-9}$ | $9.99 \times 10^{-9}$ |
| 0.600 | 1.43332 | 1.43332 | $1.318 \times 10^{-11}$ | $1.363 \times 10^{-11}$ | $2.373 \times 10^{-9}$ | $2.78 \times 10^{-8}$ |
| 0.800 | 1.89648 | 1.89648 | $1.2126 \times 10^{-11}$ | $1.247 \times 10^{-10}$ | $1.402 \times 10^{-9}$ | $6.91 \times 10^{-8}$ |
| 1.0 | 2.71828 | 2.71828 | $1.595 \times 10^{-10}$ | $1.958 \times 10^{-9}$ | $2.660 \times 10^{-8}$ | $1.71 \times 10^{-7}$ |

Table 2; Numerical results of Example 2 in the Chebyshev basis compared with the estimation error obtained by Hosseini and Shahmorad in the standard basis for n= 10, 15, 20 and the absolute error of the method applied by Makroglou for n= 10, 20.

## 6. **Conclusions**

Most integro-differential equations are usually difficult to solve analytically. In many cases, it is required to obtain the approximate solutions. For this purpose, the Tau method presented in this paper can be applied.

By comparing the estimation errors in the tables (1) and (2) we conclude that the results of operational Tau method in the Chebyshev basis is not only more accurate than the Makroglou method but it is also superiour to the Tau method applied in standard basis.

## References

[1] S.M. Hosseini and S. Shahmorad, Numerical solution of a class of integro-differential equations by the Tau method with an error estimation, *Appl. Math. Comput.* **136** (2003), 559-570.

[2] S.M. Hosseini and S. Shahmorad, Tau numerical solution of Fredholm integro-differential equations with arbitrary polynomial bases, *J. Appl. Math. Modelling* **27** (2003), 145-154.

[3] S.M. Hosseini and S. Shahmorad, A matrix formulation of the Tau method for Fredholm and Volterra linear integro - differential equations, *Korean J. Comput. Appl. Math.* **9** (2) (2002), 497-507.

[4] A. Makroglou, Convergence of a block-by-block method for non-linear Volterra integro-differential equations. *Math.Comp.* **35** (1980), 783-796.

[5] E.l. Ortiz, On the numerical solution of non-linear and functional differential equations whit the Tau method. In : Numerical treatment of differential equations in applications, Springer-Verlag, Berlin (1978), 127 - 139.

[6] E.l. Ortiz and H. Samara : An operational approach to the Tau method for the numerical solution of non-linear differential equations, *Computing* **27** (1981), 15 - 25.

**J. Pour-Mahmoud**
Faculty of Basic Science
Azarbijan University
Tabriz, Iran
e-mail: jafarlisa55@yahoo.ca

**M. Y. Rahimi and S. Shahmorad**
Faculty of Mathematical Science
Tabriz University
Tabriz, Iran
e-mail: m.rahimi@tabrizu.ac.ir
e-mail: shahmorad@tabrizu.ac.ir