

**THE OPERATIONAL TAU-ADOMIAN METHOD AND PADÉ
APPROXIMANTS FOR SOLVING GENERALIZED NON-LINEAR
VOLTERRA INTEGRO-DIFFERENTIAL EQUATIONS**

ALI KHANI, AYDIN OSTOVAR

ABSTRACT. In this paper we use the operational Tau and Adomian methods with Padé approximants for solving the general Non-linear Volterra Integro-Differential Equations (NVIDE). We will present our method based on the matrix form of (NVIDE). The corresponding unknown coefficients of our method have been determined by using computational aspects of matrices. To this end, the Padé approximants have been used to accurately determine the numerical solution. Finally, accuracy of the method has been verified by presenting some numerical computations.

2000 *Mathematics Subject Classification*: 65M70, 47Gxx, 45Gxx.

Keywords: Operational Tau-Adomian methods, Nonlinear Volterra Integro-Differential Equations, Matrix Forms, Padé approximants.

1. INTRODUCTION

In 1981, Ortiz and Samara [22] proposed an operational technique for finding a numerical solution of non-linear ordinary differential equations with some supplementary conditions based on the Tau method [9]. During the last last years considerable works have been done both in development of the above mentioned technique, its theoretical analysis and numerical applications. Various techniques have been described in a series of papers [11,12,14,15,21] for the case of linear ordinary differential eigenvalue problems. In [13,16,18-20] numerical solution of partial differential equations and their related eigenvalue problems have discussed. The object of this paper is to present a simpler operational approach by using the Adomian decomposition method for the general form of non-linear Volterra integro-differential equations of the second kind with initial conditions. This method leads to an algorithm with remarkable simplicity, while retaining the accuracy of results. The obtained series solution is converted into Padé approximants to study the behavior of the solution.

2. NON-LINEAR VOLTERRA INTEGRO-DIFFERENTIAL EQUATIONS

Consider the non-linear Volterra integro-differential equation

$$y^{(m)}(x) + G(x, y, y', \dots, y^{(m-1)}) - \int_0^x F(x, t, y, y', \dots, y^{(m)}) dt = f(x), \quad x \in [0, a] \quad (1)$$

with the initial conditions

$$y^{(j)}(0) = d_j, \quad j = 0, 1, \dots, m - 1. \quad (2)$$

Here we assume that $f(x)$ is polynomial, otherwise it can be approximated by polynomial to any degree of accuracy (by Taylor series or any other suitable method). Moreover, we suppose that $y_n(x)$ to be a polynomial approximation of degree n for $y(x)$. Then one can write

$$\begin{aligned} f(x) &= \sum_{j=0}^n f_j x^j = \underline{\mathbf{f}} \underline{\mathbf{X}} \\ y_n(x) &= \sum_{j=0}^n a_j x^j = \underline{\mathbf{a}}_n \underline{\mathbf{X}} \end{aligned} \quad (3)$$

where $\underline{\mathbf{f}} = [f_0, f_1, \dots, f_n, 0, 0, \dots]$, $\underline{\mathbf{a}}_n = [a_0, a_1, \dots, a_n, 0, 0, \dots]$ and $\underline{\mathbf{X}} = [1, x, x^2, \dots]^T$ are the coefficients vectors of right-hand side of equation (1), unknown coefficients vector and the basis vector respectively. Without loss of generality we have taken all polynomials of degree n , because if $f(x)$ and $y_n(x)$ are respectively of different degrees n_f and n_y then we can set $n = \max\{n_f, n_y\}$.

3. CONVERTING NVE TO A SYSTEM OF ALGEBRAIC EQUATIONS

The effect of differentiation on the coefficients $\underline{\mathbf{a}}_n = [a_0, \dots, a_n, 0, \dots]$ of a polynomial $y_n(x) = \underline{\mathbf{a}}_n \underline{\mathbf{X}}$ is the same as that of post-multiplication of $\underline{\mathbf{a}}_n$ by the matrix η by

$$\eta = \begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ 0 & 2 & 0 & & \vdots \\ 0 & 0 & 3 & 0 & \\ & & \dots & & \ddots \end{bmatrix}.$$

Lemma 1. *If $y_n(x)$ be a polynomial of the form*

$$y_n(x) = \sum_{i=0}^n a_i x^i = \sum_{i=0}^{\infty} a_i x^i.$$

Then

$$\frac{d^r}{dx^r} y_n(x) = \underline{\mathbf{a}}_n \eta^r \underline{\mathbf{X}}, \quad r = 1, 2, 3, \dots$$

where $\underline{\mathbf{a}}_n = [a_0, a_1, \dots, a_n, 0, 0, \dots]$.

The proof follows immediately by induction. \square

By using the Adomian decomposition method (See [13]), one can simplify the non-linear terms of (1) as follows.

By setting $\hat{G}(x) = G(x, y(x), y'(x), \dots, y^{(m-1)}(x))$, we have

$$\begin{aligned} \hat{G}(x) &= G(x, \sum_{j=0}^{\infty} a_j x^j, \sum_{j=0}^{\infty} (j+1) a_{j+1} x^j, \dots, \sum_{j=0}^{\infty} \frac{(j+m-1)!}{j!} a_{j+m-1} x^j) \\ &= \sum_{i=0}^{\infty} A_i^G x^i \\ &= \underline{\mathbf{A}}^G \underline{\mathbf{X}} \end{aligned} \tag{4}$$

where $\underline{\mathbf{A}}^G = [A_0^G, A_1^G, \dots]$ with

$$A_i^G = \frac{1}{i!} \left\{ \frac{d^i}{dx^i} G(x, \sum_{j=0}^{\infty} a_j x^j, \dots, \sum_{j=0}^{\infty} \frac{(j+m-1)!}{j!} a_{j+m-1} x^j) \right\}_{x=0} = \frac{\hat{G}^{(i)}(0)}{i!}$$

which is depend on $a_0, a_1, \dots, a_{i+m-1}$ for $i = 0, 1, \dots$

By using this method for the non-linear term under integral sign, we have

$$\begin{aligned} \hat{F}(x, t) &= F(x, t, y, y', \dots, y^{(m)}) \\ &= \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} A_{ij}^F x^i t^j \end{aligned} \tag{5}$$

where

$$\begin{aligned} A_{ij}^F &= \frac{1}{i!j!} \left\{ \frac{\partial^{i+j}}{\partial x^i \partial t^j} F(x, t, y, y', \dots, y^{(m)}) \right\}_{(x,y)=(0,0)} \\ &= \frac{\frac{\partial^{i+j} \hat{F}}{\partial x^i \partial t^j}(0, 0)}{i!j!} \end{aligned}$$

which is depend on a_0, a_1, \dots, a_{r+m} for $i, j = 0, 1, 2, \dots$, and $r = \max\{i, j\}$.

Now if we replace $y(x)$ by $y_n(x) = \sum_{i=0}^n a_i x^i$ in (5), then we have

$$\begin{aligned} \int_0^x F(x, t, y, y', \dots, y^{(m)}) dt &= \int_0^x \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} A_{ij}^F x^i t^j dt \\ &= \sum_{i=0}^{\infty} \sum_{j=0}^{\infty} A_{ij}^F \frac{x^{i+j+1}}{j+1} \\ &= \sum_{i=1}^{\infty} \hat{A}_i^F x^i \\ &= \underline{\hat{\mathbf{A}}}^F \underline{\mathbf{X}}, \end{aligned} \tag{6}$$

where $\underline{\hat{\mathbf{A}}}^F = [\hat{A}_0^F, \hat{A}_1^F, \dots]$ with $\hat{A}_0^F = 0$ and $\hat{A}_i^F = \sum_{j=1}^i \frac{A_{i-j, j-1}^F}{j}$ for $i = 1, 2, \dots$.

Therefore, the matrix form of (1) can be written as:

$$\underline{\mathbf{a}}_n \eta^m \underline{\mathbf{X}} + \underline{\mathbf{A}}^G \underline{\mathbf{X}} - \underline{\hat{\mathbf{A}}}^F \underline{\mathbf{X}} = \underline{\mathbf{f}} \underline{\mathbf{X}}$$

which yields

$$\underline{\mathbf{a}}_n \eta^m + \underline{\mathbf{A}}^G - \underline{\hat{\mathbf{A}}}^F = \underline{\mathbf{f}}, \tag{7}$$

since $\underline{\mathbf{X}}$ is a base vector. Now, the unknown coefficients can be determined by (2) and (7). Note that we use (2) to write

$$a_j = \frac{d_j}{j!} \quad j = 0, 1, \dots, m-1,$$

which determines the unknowns for the index $j = 0, \dots, m-1$.

Other coefficients are determined by solving (7), as:

$$a_{m+j} = \frac{f_j + \hat{A}_j^F - A_j^G}{\frac{(m+j)!}{j!}}, \quad j = 0, 1, \dots, n-m.$$

The obtained numerical solution by above mentioned procedure can be improved by the use of the Padé approximants.

4. PADÉ APPROXIMANTS

A Padé approximant is the ratio of two polynomials constructed from the coefficients of the Taylor series expansion of a function $u(x)$. The $[L/M]$ Padé approximants to a function $y(x)$ are given by [16].

$$\left[\begin{matrix} L \\ M \end{matrix} \right] = \frac{P_L(x)}{Q_M(x)}, \quad (8)$$

where $P_L(x)$ is polynomial of degree at most L and $Q_M(x)$ is a polynomial of degree at most M . The formal power series

$$y(x) = \sum_{i=1}^{\infty} a_i x^i, \quad (9)$$

$$y(x) - \frac{P_L(x)}{Q_M(x)} = O(x^{L+M+1}), \quad (10)$$

determine the coefficients of $P_L(x)$ and $Q_M(x)$ by the equation.

Since we can clearly multiply the numerator and denominator by a constant and leave $[L/M]$ unchanged, we imposed the normalization condition

$$Q_M(x) = 1.0. \quad (11)$$

Finally, we require that $P_L(x)$ and $Q_M(x)$ have non common factors. If we write the coefficient of $P_L(x)$ and $Q_M(x)$ as

$$\begin{aligned} P_L(x) &= p_0 + p_1 x + \cdots + p_L x^L, \\ Q_M(x) &= q_0 + q_1 x + \cdots + q_M x^M, \end{aligned} \quad (12)$$

Then by (11) and (12), we may multiply (8) by $Q_M(x)$, which linearizes the coefficient equations. We can write out (10) in more details as

$$\left\{ \begin{array}{l} a_{L+1} + a_L q_1 + \cdots + a_{L-M} q_M = 0, \\ a_{L+2} + a_{L+1} q_1 + \cdots + a_{L-M+2} q_M = 0, \\ \vdots \\ a_{L+M} + a_{L+M-1} q_1 + \cdots + a_L q_M = 0, \end{array} \right. \quad (13)$$

$$\left\{ \begin{array}{l} a_0 = p_0, \\ a_1 + a_0 q_1 = p_1, \\ \vdots \\ a_L + a_{L-1} q_1 + \cdots + a_0 q_L = p_L, \end{array} \right. \quad (14)$$

To solve these equations, we start with Eq.(13), which is a set of linear equations for all the unknown q 's. Once the q 's are known, then Eq.(14) gives an explicit formula for the unknown p 's, which complete the solution. If Eqs.(13) and (14) are nonsingular, then we can solve them directly and obtain Eq.(14), where Eq.(14) holds, and if the lower index on a sum exceeds the upper, the sum is replaced by zero:

$$\left[\frac{L}{M} \right] = \frac{\det \begin{bmatrix} a_{L-M+1} & a_{L-M+2} & \cdots & a_{L+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_L & a_{L+1} & \cdots & a_{L+M} \\ \sum_{j=M}^L a_{j-M} x^j & \sum_{j=M-1}^L a_{j-M+1} x^j & \cdots & \sum_{j=0}^L a_j x^j \end{bmatrix}}{\det \begin{bmatrix} a_{L-M+1} & a_{L-M+2} & \cdots & a_{L+1} \\ \vdots & \vdots & \ddots & \vdots \\ a_L & a_{L+1} & \cdots & a_{L+M} \\ x^M & x^{M-1} & \cdots & 1 \end{bmatrix}} \quad (15)$$

5. ESTIMATION OF ERROR FUNCTION

In this section, an error function is obtained for the approximate solution of Eqs.(1) and (2). Let $e_n(x) = y(x) - y_n(x)$ be called the error function of Tau approximation $y_n(x)$ to $y(x)$ where $y(x)$ is the exact solution. Hence $y_n(x)$ satisfies the following problem:

$$\begin{aligned} y_n^{(m)}(x) + G(x, y_n, y'_n, \dots, y_n^{(m-1)}) &= \int_0^x F(x, t, y_n, y'_n, \dots, y_n^{(m)}) dt \\ &= f(x) + H_n(x), \quad x \in [0, a]. \end{aligned} \quad (16)$$

with

$$y_n^{(j)}(0) = d_j, \quad j = 0, 1, \dots, m-1. \quad (17)$$

The function $H_n(x)$ is the perturbation term associated with $y_n(x)$. Hence

$$H_n(x) = y_n^{(m)}(x) + G(x, y_n, y'_n, \dots, y_n^{(m-1)}) - \int_0^x F(x, t, y_n, y'_n, \dots, y_n^{(m)}) dt - f(x)$$

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 problems in Eqs.(1) and (2). Subtracting

(16) and (17) from (1) and (2) respectively and taking a term of expansions $G'(x)$ and $F'(x, t)$ around $y_n(t)$, the error function $e_n(x)$ satisfies the problem

$$e_n^{(m)}(x) + \sum_{i=0}^{m-1} e_n^{(i)} \frac{\partial G}{\partial y^{(i)}}(x, y_n, y'_n, \dots, y_n^{(m-1)}) - \int_0^x \sum_{i=0}^m \frac{\partial F}{\partial y^{(i)}} F(x, t, y_n, y'_n, \dots, y_n^{(m)}) dt = -H_n(x), \quad x \in [0, a].$$

with

$$e_n^{(j)}(0) = 0, \quad j = 0, 1, \dots, m - 1.$$

6. NUMERICAL EXAMPLES

6.1. Population problem

The study of Volterra integral equations originated with the work of Volterra on population dynamics. The equation actually used by Volterra, has the form

$$\frac{dN(t)}{dt} = N(t) \left\{ \alpha - \beta N(t) - \int_0^t k(t-s) N(s) ds \right\},$$

where the term $-\beta N^2(t)$ introduced to account for the competition between individuals in the population and tends to inhibit the growth of the population [10], and $k(t-s)$ is the survival function [1].

6.2. Problem of polymer rheology

The equation

$$\mu u'(t) = u^3(t)g(t) + \int_0^t k(t-s) \left\{ \frac{u^3(t)}{u^2(s)} - u(s) \right\} ds$$

models the elongation of filament of a certain polyethylene which is stretched on the time interval $-\infty < t \leq 0$, then released and allowed to undergo elastic recovery for $t > 0$ [10].

Both problems can be solved by the method of this paper directly.

7. NUMERICAL RESULTS

The following examples are given to clarify accuracy of the presented method and also shows the importance of the after-treatment method used to improve the accuracy of the approximate solution in Tau-Padé method. Example[3,4] are selected

from different references, so their numerical results obtained here can be compared with the other numerical methods. The computations associated with the examples were performed using Maple 13 on a Personal computer.

Example 1.

$$\begin{cases} y''(x) - \text{Ln}(y(x)y'(x)) + \int_0^x \left\{ \frac{2y'(t)}{y(x)} (\sin(t)e^{-t}y(t) + \cos(t)) - y(t) \right\} dt \\ \qquad \qquad \qquad = 2\sin(x) - 2x + 1, \\ y(0) = 1, \quad y'(0) = 1, \quad 0 \leq x \leq 1. \end{cases}$$

The exact solution is given by $y(x) = e^x$. For the numerical results with $n = 10, 15$ see Table 1.

Example 2.

$$\begin{cases} y^{(4)}(x) + (1+x)e^{\text{Ln}(y''(x))} + \int_0^x 2y'(t)\text{Ln}(y''(t))dt = e^{-x} - x^2 + 1, \quad 0 \leq x \leq 1 \\ y(0) = 1, \quad y'(0) = 0, \quad y''(0) = 1, \quad y'''(0) = -1. \end{cases}$$

The exact solution is given by $y(x) = x + e^{-x}$. For the numerical results with $n = 10, 15$ see Table 2.

Example 3. [3,4,17]

$$\begin{cases} y'(x) + \int_0^x \cos(x-t)y^2(t)dt = \sin(2x), \\ y(0) = 1, \quad 0 \leq x \leq 1. \end{cases}$$

The exact solution is given by $y(x) = \cos(x)$. Table 3 shows the numerical results and comparison with the other numerical solutions.

Example 4. [2,6,8]

$$\begin{cases} y'(x) = 1 + \int_0^x y(t)y'(t)dt, \\ y(0) = 0, \quad 0 \leq x \leq 1. \end{cases}$$

The exact solution is given by $y(x) = \sqrt{2} \tan(\frac{x}{\sqrt{2}})$. For the numerical results with $n = 8$ see Table 4.

Remark 1. Note that in the following tables, the notations *Exact*, *App*, *Err.Tau*, *Est.Err.* and *Err.Tau-Pade*[m, n], have been used for exact solution, approximate solution obtained by our method and absolute error and estimate error of approximate solution and absolute error of approximate solution improved by use of the Padé approximants respectively.

x	<i>Exact</i>	<i>App</i>	<i>Err.Tau</i>	<i>Est.Err.</i>	<i>Err.Tau – Pade</i> [5, 5]
$n = 10$					
0.0	1.000000	1.000000	0	0	0
0.2	1.221403	1.221403	$5.21752e - 16$	$5.13067e - 16$	$2.48910e - 18$
0.4	1.491825	1.491825	$1.08869e - 12$	$1.05076e - 12$	$6.24388e - 15$
0.6	1.822119	1.822119	$9.56518e - 11$	$9.08333e - 11$	$6.62761e - 13$
0.8	2.225541	2.225541	$2.30479e - 09$	$2.15196e - 09$	$1.92928e - 11$
1.0	2.718282	2.718282	$2.73127e - 08$	$2.50521e - 08$	$2.76650e - 10$
x	<i>Exact</i>	<i>App</i>	<i>Err.Tau</i>	<i>Est.Err.</i>	<i>Err.Tau – Pade</i> [8, 7]
$n = 15$					
0.0	1.000000	1.000000	0	0	0
0.2	1.221403	1.221403	$3.16954e - 25$	$3.13228e - 25$	$5.87483e - 29$
0.4	1.491825	1.491825	$2.10217e - 20$	$2.05277e - 20$	$4.65275e - 24$
0.6	1.822119	1.822119	$1.39757e - 17$	$1.34834e - 17$	$3.69789e - 21$
0.8	2.225541	2.225541	$1.41155e - 15$	$1.34530e - 15$	$4.47002e - 19$
1.0	2.718282	2.718282	$5.07711e - 14$	$4.77948e - 14$	$1.92645e - 17$

Table 1. Numerical results of Example 1.

x	<i>Exact</i>	<i>App</i>	<i>Err.Tau</i>	<i>Est.Err.</i>	<i>Err.Tau – Pade</i> [5, 5]
$n = 10$					
0.0	1.000000	1.000000	0	0	0
0.2	1.018731	1.018731	$5.04646e - 16$	$5.13067e - 16$	$3.03697e - 18$
0.4	1.070320	1.070320	$1.01678e - 12$	$1.05076e - 12$	$5.31185e - 15$
0.6	1.148812	1.148812	$8.65450e - 11$	$9.08883e - 11$	$3.93157e - 13$
0.8	1.249329	1.249329	$2.01685e - 09$	$2.15196e - 09$	$7.98071e - 12$
1.0	1.367879	1.367879	$2.31143e - 08$	$2.50521e - 08$	$7.98044e - 11$
x	<i>Exact</i>	<i>App</i>	<i>Err.Tau</i>	<i>Est.Err.</i>	<i>Err.Tau – Pade</i> [8, 7]
$n = 15$					
0.0	1.000000	1.000000	0	0	0
0.2	1.018731	1.018731	$3.09583e - 25$	$3.13228e - 25$	$4.03810e - 29$
0.4	1.070320	1.070320	$2.00552e - 20$	$2.05277e - 20$	$2.19830e - 24$
0.6	1.148812	1.148812	$1.30229e - 17$	$1.34834e - 17$	$1.20086e - 21$
0.8	1.249329	1.249329	$1.28469e - 15$	$1.34530e - 15$	$9.97743e - 20$
1.0	1.367879	1.367879	$4.51317e - 14$	$4.77948e - 14$	$2.95549e - 18$

Table 2. Numerical results of Example 2.

x	<i>Exact</i>	<i>Tau</i> ($n = 15$)	<i>Tau - Pade</i> [8, 7]	<i>BPF</i>	<i>Adomian</i>
0.0	1.000000	1.000000	1.000000	1.000000	1.000000
0.1	0.995004	0.995004	0.995004	0.995141	0.994951
0.2	0.980067	0.980067	0.980067	0.975784	0.980303
0.3	0.955336	0.955336	0.955336	0.960386	0.955685
0.4	0.921061	0.921061	0.921061	0.918443	0.921165
0.5	0.877583	0.877583	0.877583	0.862193	0.877048
0.6	0.825336	0.825336	0.825336	0.828963	0.825596
0.7	0.764842	0.764842	0.764842	0.752929	0.755333
0.8	0.696707	0.696707	0.696707	0.710418	0.667739
0.9	0.621610	0.621610	0.621610	0.617232	0.547241
1.0	0.540302	0.540302	0.540302	0.566917	0.364798

Table 3. Numerical results of Example 3.

x	<i>Exact</i>	<i>Tau</i> ($n = 8$)	<i>Err.Tau</i>	<i>Err.T. - P.</i> [4, 4]	<i>Err.WGM</i>	<i>Err.HPM</i>
0	0	0	0	0	0	0
0.0625	0.062541	0.062541	$1.99e - 14$	$9.18e - 18$	$5.93e - 05$	0
0.1250	0.125327	0.125327	$1.02e - 11$	$4.73e - 15$	$2.65e - 05$	0
0.1875	0.188606	0.188606	$3.94e - 10$	$1.83e - 13$	$6.41e - 06$	0
0.2500	0.252637	0.252637	$5.28e - 09$	$2.47e - 12$	$6.29e - 05$	0
0.3125	0.317688	0.317688	$3.96e - 08$	$1.87e - 11$	$1.24e - 05$	0
0.3750	0.384043	0.384043	$2.06e - 07$	$9.84e - 11$	$5.65e - 05$	$1.00e - 05$
0.4375	0.452013	0.452012	$8.35e - 07$	$4.03e - 10$	$1.25e - 05$	$2.00e - 05$
0.5000	0.521931	0.521928	$2.81e - 06$	$1.38e - 09$	$6.59e - 05$	$5.00e - 05$
0.5625	0.594169	0.594160	$8.23e - 06$	$4.10e - 09$	$3.14e - 05$	$1.20e - 04$
0.6250	0.669142	0.669120	$2.16e - 05$	$1.10e - 08$	$5.81e - 05$	$2.40e - 04$
0.6875	0.747320	0.747268	$5.19e - 05$	$2.69e - 08$	$1.98e - 05$	$4.70e - 04$
0.7500	0.829239	0.829123	$1.16e - 04$	$6.15e - 08$	$6.10e - 05$	$8.60e - 04$
0.8125	0.915520	0.915276	$2.43e - 04$	$1.33e - 07$	$8.03e - 05$	$1.51e - 03$
0.8750	1.006886	1.006400	$4.87e - 04$	$2.73e - 07$	$1.38e - 05$	$2.56e - 03$
0.9375	1.104193	1.103263	$9.30e - 04$	$5.39e - 07$	$6.82e - 06$	$4.17e - 03$
1.0000	1.208460	1.206746	$1.71e - 03$	$1.03e - 06$	$3.98e - 05$	$6.61e - 03$

Table 4. Numerical results of Example 4.

8. CONCLUSIONS

In this paper, we have solved a special class of NVIDEs which is important in practical problems, see the given practical problems in section 6. For solving this type of problems, we have designed remarkably simple method which has high accuracy in comparison with other existing methods and clarified the accuracy through solving numerical examples (see Tables[3,4]). Some of the advantages of this method are as follows:

1. It solves NVIDEs without linearizing the nonlinear terms;
2. It gives an error estimator as a polynomial and gives more accurate solution by increasing n .

REFERENCES

- [1] J. Abdul Jerri, *Introduction to Integral Equations with Applications*, John Wiley and Sons, New York, (1999).
- [2] A. Avudainayagam, C. Vani, *Wavelet-Galerkin method for integro-differential equations*, Appl. Math. Comput., 32 (2000), 247-254.
- [3] E. Babolian , Z. Masouri , S. Hatamzadeh-Varmazyarb, *Numerical solution of nonlin-ear Volterra-Fredholm integro-differential equations via direct method using triangular functions*, Comput. Math. Appl., 58 (2009), 239-247.
- [4] E. Babolian , Z. Masouri , S. Hatamzadeh-Varmazyarb, *New direct method to solve nonlinear Volterra-Fredholm integral and integro differential equation using operational matrix with Block-pulse functions*, Progress In Electromagnetics Research B., 8 (2008), 59-76.
- [5] T. Badredine, K. Abbaoui, Y. Cherruault, *Convergence of Adomians method applied to integral equations*, Kybernetes., 28 (1999), 557-564.
- [6] D. Bahuguna, A. Ujlayan , D.N. Pandey, *A comparative study of numerical methods for solving an integro-differential equation*, Comput. Math. Appl., 57 (2009), 1485-1493.
- [7] G.A. Baker, *Essentials of Pade Approximants*, Academic press, London., (1975).
- [8] M. Ghasemi, M. Tavassoli Kajani, E. Babolian, *Application of Hes homotopy perturbation method to nonlinear integro-differential equations*, Appl. Math. Comput., 188 (2007), 538-548.
- [9] C. Lanczos, *Trigonometric interpolation of empirical and analytical functions*, J. Math. Phys., 17 (1938), 123-129.
- [10] P. Linz, *Analytical and Numerical Methods for Volterra Equations*, SIAM, USA., (1985).
- [11] K.M. Liu, E.L. Ortiz, *Numerical solution of ordinary and partial function-differential eigenvalue problems with the Tau method*, Computing., 41 (1989), 205-217.
- [12] K.M. Liu, E.L. Ortiz, *Tau method approximation of differential eigenvalue problems where the spectral parameter enters nonlinearly*, J. Comput. Phys., 72 (1987), 299-310.

- [13] K.M. Liu, E.L. Ortiz, *Numerical solution of eigenvalue problems for partial differential equations with the Tau-lines method*, J. Comp. Math. Appl. B, 12 (1986), 1153-1168.
- [14] K.M. Liu and E.L. Ortiz, *Approximation of eigenvalues defined by ordinary differential equations with the Tau method*, Matrix Pencils, Springer, Berlin., (1983), 90-102.
- [15] K.M. Liu, E.L. Ortiz, *Eigenvalue problems for singularly perturbed differential equations*, in J.J.H. Miller (Ed.), Proceeding of the BAIL II conference, Boole press, Dublin., (1982), 324-29.
- [16] K.M. Liu, E.L. Ortiz, K.S. Pun, *Numerical solution of steklovs partial differential equation eigenvalue problem*, in J.J.H. Miller (Ed.), Computational and asymptotic methods for boundary and interior layers III, Boole Press, Dublin., (1984), 244-249.
- [17] K. Maleknejad, M. Hadizadeh, *The numerical analysis of Adomians decomposition method for nonlinear Volterra integral and integro-differential equations*, International Journal of Engineering Science., 8 (1997), 33-48.
- [18] E.L. Ortiz, K.S. Pun, *A bi-dimensional Tau-elements method for the numerical solution of nonlinear partial differential equations with an application to Burgersequation*, J.Comp. Math. Appl. B, 12 (1986), 1225-1240.
- [19] E.L. Ortiz and K.S. Pun, *Numerical solution of nonlinear partial differential equations with Tau method*, J. Comp. Appl. Math., 12 (1985), 511-516.
- [20] E.L. Ortiz, H. Samara, *Numerical solution of partial differential equations with variable coefficients with an operational approach to the Tau method*, J.Comp. Math. Appl., 10 (1984), 5-13.
- [21] E.L. Ortiz, H. Samara, *Numerical solution of differential eigenvalue problems with an operational approach to the Tau method*, Computing., 31 (1983), 95-103.
- [22] E.L. Ortiz, H. Samara, *An operational approach to the Tau method for the numerical solution of non-linear differential equations*, Computing., 27 (1981), 15-25.

Ali Khani, Aydin Ostovar
Department of Applied Mathematics,
Azarbaijan Shahid Madani University,
Tabriz, Iran
email: khani@azaruniv.edu, aydin.ostovar@yahoo.com