Bulletin of the Iranian Mathematical Society Vol. 34 *No.* 2 (*2008*), *pp 141-152.*

NUMERICAL SOLUTION OF SPECIAL CLASS OF SYSTEMS OF NON-LINEAR VOLTERRA INTEGRO-DIFFERENTIAL EQUATIONS BY A SIMPLE HIGH ACCURACY METHOD

A. KHANI*, M. MOHSENI MOGHADAM AND S. SHAHMORAD

Communicated by Mohammad Asadzadeh

ABSTRACT. We study a simple new method to find a numerical solution for the special class of Systems of Non-linear Volterra integro-differential Equations (SNVE). To this end, we will present our method based on matrix form of SNVE. The corresponding unknown coefficients of our method is determined using computational aspects of matrices. Finally, accuracy of the method is verified by presentation of some numerical computations.

1. **Introduction**

The SNVE arise from mathematical modeling of many scientific phenomena. Non-linear phenomena such as population and polymer rheology [2,3], that appear in many applications in scientific fields, can be modeled by non-linear Volterra integro-differential equations. Analytically, solving these problems are difficult, and so we want to present a simple high accuracy method for the numerical solution of an special class of these non-linear problems. Recently, the operational Tau

MSC(2000): Primary 65R20
Keywords: Volterra Integro-Differential Equations, Matrix Forms, Numerical Solutions
Received: 15 October 2007, Accepted: 19 April 2008
∗Corresponding author

method and Adomian decomposition method were developed to solve the non-linear Volterra integro-differential equations [4,5]. These methods lead to solving systems of non-linear algebraic equations. Our method here is a simple operational approach using the Adomian decomposition method, which leads to a system of linear algebraic equations and is solved simply. This method also leads to an algorithm with a remarkable simplicity. It must be mentioned that the Adomian decomposition method play an important role here, since using this method we convert the non-linear terms in SNVE to the polynomials in terms of independent variables and then use the operational Tau method to convert the SNVE to the system of algebraic equations.

2. System of non-linear Volterra integro-differential equations

Consider the following non-linear system of Volterra integro-differential equations,

$$\sum_{j=1}^{r} g_{ij}(x)y_j^{(m_j)}(x) + G_i(x, y_1, y_1', \ldots, y_1^{(m_1-1)}, \ldots, y_r, y_r', \ldots, y_r^{(m_r-1)})$$
$$- \int_0^x F_i(x, t, y_1, y_1', \ldots, y_1^{(m_1)}, \ldots, y_r, y_r', \ldots, y_r^{(m_r)})dt = f_i(x),$$
$$x \in [0, a], \quad i = 1, \ldots, r, \tag{2.1}$$

whit the initial conditions,

$$y_i^{(j)}(0) = d_{ij}, \quad j = 0, 1, \ldots, m_i - 1, \; i = 1, 2, \ldots, r. \tag{2.2}$$

Here, we assume that $g_{ij}(x)$ and $f_i(x)$ for $i, j = 1, 2, \ldots, r$ are polynomials; otherwise, they can be approximated by polynomials to any degree of accuracy (by Taylor series or any other suitable approximation). We denote by $G_i(x, y_1, y_1', \ldots, y_1^{(m_1-1)}, \ldots, y_r, y_r', \ldots, y_r^{(m_r-1)})$ and $F_i(x, t, y_1, y_1', \ldots, y_1^{(m_1)}, \ldots, y_r, y_r', \ldots, y_r^{(m_r)})$, for $i = 1, 2, \ldots, r$, the non-linear terms of SNVE. We suppose that $y_{in}(x)$ is a polynomial approximation of degree n for $y_i(x)$. Then, one can write,

$$g_{ij}(x) = \sum_{k=0}^{n} g_{ijk} x^k = \underline{\mathbf{g}}_{ij} \underline{\underline{\mathbf{X}}}$$
$$f_i(x) = \sum_{j=0}^{n} f_{ij} x^j = \underline{\mathbf{f}}_i \underline{\underline{\mathbf{X}}}$$
$$y_{in}(x) = \sum_{j=0}^{n} a_{ij} x^j = \underline{\mathbf{a}}_i \underline{\underline{\mathbf{X}}}, \tag{2.3}$$

where $\underline{\mathbf{g}}_{ij} = [g_{ij0}, g_{ij1}, \ldots, g_{ijn}, 0, \ldots]$, $\underline{\mathbf{f}}_i = [f_{i0}, f_{i1}, \ldots, f_{in}, 0, \ldots]$, $\underline{\mathbf{a}}_i = [a_{i0}, a_{i1}, \ldots, a_{in}, 0, \ldots]$, and $\underline{\underline{\mathbf{X}}} = [1, x, x^2, \ldots]^T$. Without loss of generality, we have taken all polynomials of the same degree n, because if $g_{ij}(x)$, $f_i(x)$ and $y_{in}(x)$ are respectively of different degrees $n_{g_{ij}}$,n_{f_i} and n_{y_i}, then we can set $n = \max\{n_{g_{ij}}, n_{f_i}, n_{y_i}\}$.

3. Converting SNVE to a system of algebraic equations

The effect of differentiation or shifting on the coefficients $\underline{\mathbf{p}}_n = [p_0, p_1, \cdots, p_n, 0, 0, \cdots]$ of a polynomial$p_n(x) = \underline{\mathbf{p}}_n \underline{\underline{\mathbf{X}}}$ is the same as that of post-multiplication of $\underline{\mathbf{p}}_n$ by either the matrix η or the matrix μ defined by:

$$\mu = \begin{bmatrix} 0 & 1 & 0 & 0 & \\ & 0 & 1 & 0 & \\ & & 0 & 1 & \vdots \\ & & & 0 & \\ & & \cdots & & \ddots \end{bmatrix} \quad \text{and} \quad \eta = \begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ 0 & 2 & 0 & & \vdots \\ 0 & 0 & 3 & 0 & \\ & & \cdots & & \ddots \end{bmatrix}.$$

Lemma 3.1. *Let $p_n(x)$ be a polynomial of the form,*

$$p_n(x) = \sum_{i=0}^{n} p_i x^i = \underline{\mathbf{p}}_n \underline{\underline{\mathbf{X}}}.$$

Then,

i) $\frac{d^k}{dx^k} p_n(x) = \underline{\mathbf{p}}_n \eta^k \underline{\underline{\mathbf{X}}}$, $k = 0, 1, 2, \cdots$.

ii) $x^k p_n(x) = \underline{\mathbf{p}}_n \mu^k \underline{\underline{\mathbf{X}}}$, $k = 0, 1, 2, \cdots$.

Proof. The proof follows immediately by induction. □

Lemma 3.2. *Let $p(x) = \underline{\mathbf{p}}\,\underline{\underline{\mathbf{X}}}$ and $q(x) = \underline{\mathbf{q}}\,\underline{\underline{\mathbf{X}}}$ with $\underline{\mathbf{p}} = [p_0, p_1, \ldots]$ and $\underline{\mathbf{q}} = [q_0, q_1, \ldots]$. Then, $p(x)q(x) = \underline{\mathbf{p}} q(\mu) \underline{\underline{\mathbf{X}}}$.*

Proof. See [6].

Using lemmas 3.1 and 3.2, one can write,

$$\begin{aligned} g_{ij}(x)y_j^{(m_j)}(x) &= g_{ij}(x)\underline{\mathbf{a}}_j\eta^{m_j}\underline{\underline{\mathbf{X}}} \\ &= \underline{\mathbf{a}}_j\eta^{m_j}g_{ij}(\mu)\underline{\underline{\mathbf{X}}}, \qquad i,j=1,2,\ldots,r, \end{aligned} \tag{3.1}$$

where,

$$g_{ij}(\mu) = \begin{bmatrix} g_{ij0} & g_{ij1} & g_{ij2} & g_{ij3} & \cdots & g_{ij,n} & 0 & 0 & \cdots \\ 0 & g_{ij0} & g_{ij1} & g_{ij2} & \cdots & g_{ij,n-1} & g_{ij,n} & 0 & \cdots \\ 0 & 0 & g_{ij0} & g_{ji1} & \cdots & g_{ij,n-2} & g_{ij,n-1} & g_{ij,n} & \cdots \\ & & & & \vdots & & & & \ddots \\ & & & & & & & & \end{bmatrix},$$

and

$$\eta^{m_j} = \begin{bmatrix} 0 & 0 & 0 & 0 & \cdots \\ & \vdots & & \vdots & \\ 0 & 0 & 0 & 0 & \cdots \\ \frac{m_j!}{0!} & 0 & 0 & 0 & \cdots \\ 0 & \frac{(m_j+1)!}{1!} & 0 & 0 & \cdots \\ 0 & 0 & \frac{(m_j+2)!}{2!} & 0 & \cdots \\ 0 & 0 & 0 & \frac{(m_j+3)!}{3!} & \cdots \\ & \vdots & & \vdots & \ddots \\ & & & & \end{bmatrix}.$$

thus,

$$\eta^{m_j} g_{ij}(\mu) =$$

$$\begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\ & & & \vdots & & & \vdots \\ 0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\ m_j! g_{ij0} & m_j! g_{ij1} & m_j! g_{ij2} & \cdots & m_j! g_{ijn} & 0 & \cdots \\ 0 & \frac{(m_j+1)!}{1!} g_{ij0} & \frac{(m_j+1)!}{1!} g_{ij1} & \cdots & \frac{(m_j+1)!}{1!} g_{ij,n-1} & \frac{(m_j+1)!}{1!} g_{ijn} & \cdots \\ 0 & 0 & \frac{(m_j+2)!}{2!} g_{ij0} & \cdots & \frac{(m_j+2)!}{2!} g_{ij,n-2} & \frac{(m_j+2)!}{2!} g_{ij,n-1} & \cdots \\ & & & \vdots & & & \ddots \end{bmatrix}.$$

Using the Adomian decomposition method (See [1]), one can simplify the non-linear terms of (2.1) as follows.

Setting

$$\hat{G}_i(x) = G_i(x, y_1(x), y_1'(x), \ldots, y_1^{(m_1-1)}(x), \ldots, y_r(x), y_r'(x), \ldots, y_r^{(m_r-1)}(x)),$$

we have,

$$\begin{aligned} \hat{G}_i(x) &= G_i(x, \sum_{j=0}^{\infty} a_{1j} x^j, \sum_{j=0}^{\infty} j a_{1j} x^j, \ldots, \sum_{j=0}^{\infty} \frac{(j+m_1-1)!}{j!} a_{1,j+m_1-1} x^j, \ldots, \\ &\quad \sum_{j=0}^{\infty} a_{rj} x^j, \sum_{j=0}^{\infty} j a_{rj} x^j, \ldots, \sum_{j=0}^{\infty} \frac{(j+m_r-1)!}{j!} a_{r,j+m_r-1} x^j) \\ &= \textstyle\sum_{j=0}^{\infty} A_j^{G_i} x^j \\ &= \underline{\mathbf{A}}^{G_i} \underline{\underline{\mathbf{X}}}, \qquad i = 1, 2, \ldots, r, \end{aligned} \tag{3.2}$$

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

$$\begin{aligned} A_k^{G_i} = \frac{1}{k!}\{\frac{d^k}{dx^k} G_i(x, \sum_{j=0}^{\infty} a_{1j} x^j, \sum_{j=0}^{\infty} j a_{1j} x^j, \ldots, \sum_{j=0}^{\infty} \frac{(j+m_1-1)!}{j!} a_{1,j+m_1-1} x^j, \ldots, \\ \sum_{j=0}^{\infty} a_{rj} x^j, \sum_{j=0}^{\infty} j a_{rj} x^j, \sum_{j=0}^{\infty} \frac{(j+m_r-1)!}{j!} a_{r,j+m_r-1} x^j)\}_{x=0} \end{aligned}$$

$$= \frac{\hat{G}_i^{(k)}(0)}{k!}, \quad i = 1, 2, \ldots, r, \ \ k = 0, 1, 2, \ldots,$$

which depend on $a_{10}, a_{11}, \ldots, a_{1,k+m_1-1}, \ldots, a_{r0}, a_{r1}, \ldots, a_{r,k+m_r-1}$, for $k = 0, 1, \cdots$.

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

$$\begin{aligned} \hat{F}_i(x,t) &= F_i(x, t, y_1, y_1^{'}, \ldots, y_1^{(m_1)}, \ldots, y_r, y_r^{'}, \ldots, y_r^{(m_r)}) \\ &= \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} A_{jk}^{F_i} x^j t^k, \quad i = 1, 2, \ldots, r, \end{aligned} \tag{3.3}$$

where,

$$\begin{aligned} A_{jk}^{F_i} &= \tfrac{1}{j!k!}\{\tfrac{\partial^{j+k}}{\partial x^j \partial t^k} F(x, t, y_1, y_1^{'}, \ldots, y_1^{(m_1)}, \ldots, \\ & \quad y_r, y_r^{'}, \ldots, y_r^{(m_r)})\}_{(x,t)=(0,0)} \\ &= \frac{\frac{\partial^{j+k} \hat{F}_i}{\partial x^j \partial t^k}(0,0)}{j!k!}, \quad i = 1, 2, \ldots, r, \ \ j, k = 0, 1, 2, \ldots, \end{aligned}$$

which depend on $a_{10}, a_{11}, \ldots, a_{1,s+m_1}, \ldots, a_{r0}, a_{r1}, \ldots, a_{r,s+m_r}$, for $j, k = 0, 1, 2, \ldots$, and $s = \max\{j, k\}$.

Now, if we replace $y_i(x)$ by $y_{in}(x) = \sum_{j=0}^{n} a_{ij} x^j$ in (3.3), then we have,

$$\begin{aligned} \int_0^x F_i(x, t, y_1, y_1^{'}, \ \ldots \ , y_1^{(m_1)}, \ldots, y_r, y_r^{'}, \ldots, y_r^{(m_r)}) dt = \\ &= \int_0^x \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} A_{jk}^{F_i} x^j t^k dt \\ &= \sum_{j=0}^{\infty} \sum_{k=0}^{\infty} A_{jk}^{F_i} \tfrac{x^{j+k+1}}{k+1} \\ &= \sum_{j=1}^{\infty} \hat{A}_j^{F_i} x^j \\ &= \underline{\hat{\mathbf{A}}}^{F_i} \underline{\underline{\mathbf{X}}}, \end{aligned} \tag{3.4}$$

where, $\underline{\hat{\mathbf{A}}}^{F_i} = [\hat{A}_0^{F_i}, \hat{A}_1^{F_i}, \ldots]$ with $\hat{A}_0^{F_i} = 0$ and $\hat{A}_j^{F_i} = \sum_{k=1}^{j} \frac{A_{j-k,k-1}^{F_i}}{k}$, for $i = 1, \ldots, r$ and $j = 1, 2, \cdots$.

Therefore, by (3.1), (3.2) and (3.4) the matrix form of (2.1) can be written as:

$$\sum_{j=1}^{r} \underline{\mathbf{a}}_j \eta^{m_j} g_{ij}(\mu) \underline{\underline{\mathbf{X}}} + \underline{\mathbf{A}}^{G_i} \underline{\underline{\mathbf{X}}} - \underline{\hat{\mathbf{A}}}^{F_i} \underline{\underline{\mathbf{X}}} = \underline{\mathbf{f}}_i \, \underline{\underline{\mathbf{X}}}, \quad i = 1, 2, \ldots, r,$$

which yields:

$$\sum_{j=1}^{r} \underline{\mathbf{a}}_j \eta^{m_j} g_{ij}(\mu) + \underline{\mathbf{A}}^{G_i} - \underline{\hat{\mathbf{A}}}^{F_i}, = \underline{\mathbf{f}}_i, \quad i = 1, 2, \ldots, r, \tag{3.5}$$

since $\underline{\underline{\mathbf{X}}}$ is a base vector. Now, the unknown coefficients can be determined by (2.2) and (3.5). Note that we use (2.2) to write,

$$a_{ij} = \frac{d_{ij}}{j!}, \quad i = 1, 2, \ldots, r, \; j = 0, 1, \cdots, m_i - 1,$$

and determine other coefficients by forward substituting from the following systems,

$$\begin{cases} m_1! g_{110} a_{1,m_1} + m_2! g_{120} a_{2,m_2} + \cdots + m_r! g_{1r0} a_{r,m_r} = h_{10} \\ m_1! g_{210} a_{1,m_1} + m_2! g_{220} a_{2,m_2} + \cdots + m_r! g_{2r0} a_{r,m_r} = h_{20} \\ \qquad \vdots \qquad\qquad\qquad \vdots \\ m_1! g_{r10} a_{1,m_1} + m_2! g_{r20} a_{2,m_2} + \cdots + m_r! g_{rr0} a_{r,m_r} = h_{r0} \end{cases}$$

where,

$$h_{i0} = f_{i0} + \hat{A}_0^{F_i} - A_0^{G_i} = f_{i0} - A_0^{G_i}, \quad i = 1, \ldots, r,$$

since $\hat{A}_0^{F_i} = 0$, for $i = 1, \ldots, r$.

We continue this substituting process to determine the remaining unknowns,

$$\begin{cases} \frac{(m_1+j)!}{j!} g_{110} a_{1,m_1+j} + \frac{(m_2+j)!}{j!} g_{120} a_{2,m_2+j} + \cdots + \frac{(m_r+j)!}{j!} g_{1r0} a_{r,m_r+j} = h_{1j} \\ \frac{(m_1+j)!}{j!} g_{210} a_{1,m_1+j} + \frac{(m_2+j)!}{j!} g_{220} a_{2,m_2+j} + \cdots + \frac{(m_r+j)!}{j!} g_{2r0} a_{r,m_r+j} = h_{2j} \\ \qquad \vdots \\ \frac{(m_1+j)!}{j!} g_{r10} a_{1,m_1+j} + \frac{(m_2+j)!}{j!} g_{r20} a_{2,m_2+j} + \cdots + \frac{(m_r+j)!}{j!} g_{rr0} a_{r,m_r+j} = h_{rj} \end{cases}$$

where,

$$h_{ij} = f_{ij} + \hat{A}_j^{F_i} - A_j^{G_i} - \sum_{k=1}^{r} \sum_{s=0}^{j-1} \frac{(m_k + s)!}{s!} g_{ik,j+1-s} a_{k,m_k+s},$$

for $i = 1, \ldots, r, \quad j = 1, 2, \ldots, \tilde{m}$, where $\tilde{m} = n - \min\{m_1, m_2, \ldots, m_r\}$. The last system terminates determining the unknowns.

Remark 3.3. Note that, this method failed to solve (2.1)-(2.2), whenever all coefficients of $g_{ij}(x)$, for $i, j = 1, 2, \dots, r$, were zero at $x = 0$.

4. **Numerical examples**

The following examples are given to clarify accuracy of the proposed method. Note that all results were obtained using programs in the Maple 8 software environment.

Example 1:

$$\begin{cases} (x+x^2)y_1(x) + 2y_2(x) + \int_0^x \big(y_1(t) - y_2(t)\big)dt = 2x - \frac{1}{2}x^2 + \frac{4}{3}x^3 + x^4, \\ y_1(x) - xyu_2(x) - \int_0^x \big(2xy_1(t) + 2y_2(t)\big)dt = -x^2 - \frac{2}{3}x^4, \quad 0 \leq x \leq 1. \end{cases}$$

The exact solution is given by $y_1(x) = x^2, y_2(x) = x$. For the numerical results with $n = 2$, see Table 1.

Example 2:

$$\begin{cases} (cos^2 x - 2)y_1(x) + (x-1)y_2(x) - \\ \qquad \int_0^x \{e^{t-x}\big(y_1(t) - y_2(t)\big) + 8sin(t-x)y_1(t)y_2^2(t)\}dt = -e^{-x}, \\ e^{2x}y_1^(x) - 4xcos(x)y_2(x) - \\ \qquad \int_0^x \{e^{t+x}\big(y_1(t) + y_2(t)\big) + 4(t+x)y_1(t)y_2(t)\}dt = -3x - \frac{1}{2}sin(2x), \\ 0 \leq x \leq \frac{\pi}{2}. \end{cases}$$

The exact solution is given by $y_1(x) = sin(x), y_2(x) = cos(x)$. For the numerical results with $n = 5, 10, 15$, see Table 2.

Example 3:

$$\begin{cases} -e^x y_1'(x) + xy_2(x) - \int_0^x e^{x+t} y_1(t)y_2(t)dt = xe^{-x}, \\ y_1'(x) + y_2(x) + \int_0^x (y_1(t) + y_2(t))dt = 1, \quad y_1(0) = 1, \quad 0 \leq x \leq \frac{\pi}{2}. \end{cases}$$

The exact solution is given by $y_1(x) = cos(x), y_2(x) = e^{-x}$. For the numerical results with $n = 5, 10, 15$, see Table 3.

Example 4:

$$\begin{cases} 2y_1^{''}(x)-(x+1)y_2^{'}(x)-y_2(x)e^{y_1(x)}+ \\ \qquad \int_0^x\{cos(2t)e^{\frac{y_1(t)y_2(t)}{y_2(x)}}-y_1(x)\}dt=-(sin(x)+cos(x)), \\ y_1^{''}(x)-y_2^{'}(x)-y_1^{'}(x)y_2(x)- \\ \qquad \int_0^x\frac{3y_1(t)y_2^2(t)}{y_2(x)}dt=-\frac{1}{cos(x)}, \\ y_1(0)=0,\ y_1^{'}(0)=1,\ y_2(0)=1,\ 0\le x\le\frac{\pi}{2}. \end{cases}$$

The exact solution is given by $y_1(x)=sin(x),\ y_2(x)=cos(x)$. For the numerical results with $n=5,10,15$, see Table 4.

Example 5:

$$\begin{cases} xy_1^{''}(x)+(1+3x^2)y_2^{'''}(x)+xy_1^2(x)-y_1^{'}(x)y_2(x)-y_2^{''}(x)- \\ \qquad \int_0^x\{7y_1^2(t)+\frac{y_2^{''}(t)}{y_2(x)}\}dt=6x^2-1+e^{-x}, \\ y_1^{''}(x)+y_2^{'''}(x)-y_2(x)-\int_0^x\frac{y_1^{''}(t)y_2(t)}{y_2^{'}(x)}dt=6-6e^{-x}, \\ y_1(0)=y_1^{'}(0)=0,\quad y_2(0)=y_2^{'}(0)=y_2^{''}(0)=1,\quad 0\le x\le 1. \end{cases}$$

The exact solution is given by $y_1(x)=x^3,\ y_2(x)=e^x$. For the numerical results with $n=5,10,15$, see Table 5.

Remark 4.1. Note that in the following tables, the notations Exacty_i, App.y_i and Abs.Err.y_i, for $i=1,2$, have been used for exact solution, approximate solution obtained by our method and absolute error of approximate solution, respectively.

Table 1: Example 1

n	x	$Exacty_1$	$App.y_1$	$Abs.Err.y_1$	$Exacty_2$	$App.y_2$	$Abs.Err.y_2$
2	0.00	0.000000	0.000000	0	0.000000	0.000000	0
	0.20	0.040000	0.040000	0	0.200000	0.200000	0
	0.40	0.160000	0.160000	0	0.400000	0.400000	0
	0.60	0.360000	0.360000	0	0.600000	0.600000	0
	0.80	0.640000	0.640000	0	0.800000	0.800000	0
	1.00	1.000000	1.000000	0	1.000000	1.000000	0

Table 2: Example 2

n	x	$Exacty_1$	$App.y_1$	$Abs.Err.y_1$	$Exacty_2$	$App.y_2$	$Abs.Err.y_2$
5	0.00	0.000000	0.000000	0	1.000000	1.000000	0
	0.31	0.309017	0.309017	$5.98444e-08$	0.951057	0.951058	$1.33291e-06$
	0.63	0.587785	0.587793	$7.62868e-06$	0.809017	0.809102	$8.48570e-05$
	0.94	0.809017	0.809146	$1.29455e-04$	0.587785	0.588743	$9.58118e-04$
	1.26	0.951057	0.952017	$9.60607e-04$	0.309017	0.314335	$5.31768e-03$
	1.57	1.000000	1.004525	$4.52486e-03$	0.000000	0.019969	$1.99690e-02$
10	0.00	0.000000	0.000000	0	1.000000	1.000000	0
	0.31	0.309017	0.309017	$7.36577e-14$	0.951057	0.951057	$1.92853e-15$
	0.63	0.587785	0.587785	$1.50565e-10$	0.809017	0.809017	$7.88642e-12$
	0.94	0.809017	0.809017	$1.29825e-08$	0.587785	0.587785	$1.02047e-09$
	1.26	0.951057	0.951057	$3.06032e-07$	0.309017	0.309017	$3.20938e-08$
	1.57	1.000000	1.000004	$3.54258e-06$	0.000000	0.000000	$4.64766e-07$
15	0.00	0.000000	0.000000	0	1.000000	1.000000	0
	0.31	0.309017	0.309017	$7.94976e-24$	0.951057	0.951057	$4.30168e-22$
	0.63	0.587785	0.587785	$1.04109e-18$	0.809017	0.809017	$2.81643e-17$
	0.94	0.809017	0.809017	$1.02427e-15$	0.587785	0.587785	$1.84697e-14$
	1.26	0.951057	0.951057	$1.35987e-13$	0.309017	0.309017	$1.83866e-12$
	1.57	1.000000	1.000000	$6.02342e-12$	0.000000	0.000000	$6.51336e-11$

Table 3: Example 3

n	x	$Exacty_1$	$App.y_1$	$Abs.Err.y_1$	$Exacty_2$	$App.y_2$	$Abs.Err.y_2$
5	0.00	1.000000	1.000000	0	1.000000	1.000000	0
	0.31	0.951057	0.951058	$1.33291e-06$	0.730403	0.730401	$1.27761e-06$
	0.63	0.809017	0.809102	$8.48570e-05$	0.533488	0.533410	$7.83491e-05$
	0.94	0.587785	0.588743	$9.58118e-04$	0.389661	0.388805	$8.56311e-04$
	1.26	0.309017	0.314335	$5.31768e-03$	0.284610	0.279987	$4.62252e-03$
	1.57	0.000000	0.019969	$1.99690e-02$	0.207880	0.190917	$1.69626e-02$
10	0.00	1.000000	1.000000	0	1.000000	1.000000	0
	0.31	0.951057	0.951057	$1.92853e-15$	0.730403	0.730403	$7.18203e-14$
	0.63	0.809017	0.809017	$7.88642e-12$	0.533488	0.533488	$1.43408e-10$
	0.94	0.587785	0.587785	$1.02047e-09$	0.389661	0.389661	$1.21007e-08$
	0.26	0.309017	0.309017	$3.20938e-08$	0.284610	0.284610	$2.79636e-07$
	1.57	0.000000	0.000000	$4.64766e-07$	0.207880	0.207883	$3.17890e-06$
15	0.00	1.000000	1.000000	0	1.000000	1.000000	0
	0.31	0.951057	0.951057	$4.30168e-22$	0.730403	0.730403	$4.22491e-22$
	0.63	0.809017	0.809017	$2.81643e-17$	0.533488	0.533488	$2.71935e-17$
	0.94	0.587785	0.587785	$1.84697e-14$	0.389661	0.389661	$1.75476e-14$
	1.26	0.309017	0.309017	$1.83866e-12$	0.284610	0.284610	$1.72048e-12$
	1.57	0.000000	0.000000	$6.51336e-11$	0.207880	0.207880	$6.00816e-11$

Table 4: Example 4

n	x	$Exacty_1$	$App.y_1$	$Abs.Err.y_1$	$Exacty_2$	$App.y_2$	$Abs.Err.y_2$
5	0.00	0.000000	0.000000	0	1.000000	1.000000	0
	0.31	0.309017	0.309017	$5.98444e-08$	0.951057	0.951058	$1.33291e-06$
	0.63	0.587785	0.587793	$7.62868e-06$	0.809017	0.809102	$8.48570e-05$
	0.94	0.809017	0.809146	$1.29455e-04$	0.587785	0.588743	$9.58118e-04$
	1.26	0.951057	0.952017	$9.60607e-04$	0.309017	0.314335	$5.31768e-03$
	1.57	1.000000	1.004525	$4.52486e-03$	0.000000	0.019969	$1.99690e-02$
10	0.00	0.000000	0.000000	0	1.000000	1.000000	0
	0.31	0.309017	0.309017	$7.36577e-14$	0.951057	0.951057	$1.92853e-15$
	0.63	0.587785	0.587785	$1.50565e-10$	0.809017	0.809017	$7.88642e-12$
	0.94	0.809017	0.809017	$1.29825e-08$	0.587785	0.587785	$1.02047e-09$
	1.26	0.951057	0.951057	$3.06032e-07$	0.309017	0.309017	$3.20938e-08$
	1.57	1.000000	1.000004	$3.54258e-06$	0.000000	0.000000	$4.64766e-07$
15	0.00	0.000000	0.000000	0	1.000000	1.000000	0
	0.31	0.309017	0.309017	$7.94976e-24$	0.951057	0.951057	$4.30168e-22$
	0.63	0.587785	0.587785	$1.04109e-18$	0.809017	0.809017	$2.81643e-17$
	0.94	0.809017	0.809017	$1.02427e-15$	0.587785	0.587785	$1.84697e-14$
	1.26	0.951057	0.951057	$1.35987e-13$	0.309017	0.309017	$1.83866e-12$
	1.57	1.000000	1.000000	$6.02342e-12$	0.000000	0.000000	$6.51336e-11$

Table 5: Example 5

n	x	$Exacty_1$	$App.y_1$	$Abs.Err.y_1$	$Exacty_2$	$App.y_2$	$Abs.Err.y_2$
5	0.00	0.000000	0.000000	0	1.000000	1.000000	0
	0.20	0.008000	0.008000	0	1.221403	1.221403	$9.14935e-08$
	0.40	0.064000	0.064000	0	1.491825	1.491819	$6.03097e-06$
	0.60	0.216000	0.216000	0	1.822119	1.822048	$7.08004e-05$
	0.80	0.512000	0.512000	0	2.225541	2.225131	$4.10262e-04$
	1.00	1.000000	1.000000	0	2.718282	2.716667	$1.61516e-03$
10	0.00	0.000000	0.000000	0	1.000000	1.000000	0
	0.20	0.008000	0.008000	0	1.221403	1.221403	$5.21752e-16$
	0.40	0.064000	0.064000	0	1.491825	1.491825	$1.08690e-12$
	0.60	0.216000	0.216000	0	1.822119	1.822119	$9.56518e-11$
	0.80	0.512000	0.512000	0	2.225541	2.225541	$2.30479e-09$
	1.00	1.000000	1.000000	0	2.718282	2.718282	$2.73127e-08$
15	0.00	0.000000	0.000000	0	1.000000	1.000000	0
	0.20	0.008000	0.008000	0	1.221403	1.221403	$3.16954e-25$
	0.40	0.064000	0.064000	0	1.491825	1.491825	$2.10217e-20$
	0.60	0.216000	0.216000	0	1.822119	1.822119	$1.39757e-17$
	0.80	0.512000	0.512000	0	2.225541	2.225541	$1.41155e-15$
	1.00	1.000000	1.000000	0	2.718282	2.718282	$5.07711e-14$

5. Conclusion

We considered a practically important special class of systems of non-linear Voltera integro-differential equations. We designed a remarkably simple method which obtained a high accuracy in solving these problems. We verified this by solving several numerical examples (see tables 1, 2, 3, 4 and 5).

Acknowledgment

The authors thank the referees who comments leading to improvements. However, the scientific responsibility rests on the authors.

References

[1] T. Badredine, K. Abbaoui and Y. Cherruault, Convergence of Adomian's method applied to integral equations, Kybernetes, **28**(5) (1999), 557-564.

[2] P. Linz, *Analytical and Numerical Methods for Volterra Equations*, SIAM, Philadelphia, PA, (1985).

[3] J. Abdul Jerri, *Introduction to Integral Equations with Applications*, John Wiley and Sons, New York, (1999).

[4] G. Ebadi, M. Y. Rahimi-Ardabili and S. Shahmorad, Numerical solution of the non-linear Volterra integro-differential equations by the Tau method, *Appl. Math. Comput.* **188** (2007), 1580-1586.

[5] E. Babolian and J. Biazar, Solving concrete examples by Adomian method, *Appl. Math. Comput.* **135** (2003), 161-167.

[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.

Ali Khani
Faculty of Science, Department of Mathematics, Azarbaijan University of Tarbiat Moallem, Tabriz, IRAN
Email: khani@azaruniv.edu

Mahmoud Mohseni Moghadam
Mahani Mathematical Research Center, University of Kerman, Kerman, IRAN
Email: mohseni@mail.uk.ac.ir

Sedaghat Shahmorad
Department of Applied Mathematics, University of Tabriz, Tabriz, IRAN
Email: shahmorad@tabrizu.ac.ir