Volume 31, N. 1, pp. 191–203, 2012

ISSN 0101-8205 / ISSN 1807-0302 (Online)
www.scielo.br/cam

A block by block method with Romberg quadrature for the system of Urysohn type Volterra integral equations

R. KATANI and S. SHAHMORAD*

Faculty of Mathematical Sciences, University of Tabriz, Tabriz, Iran

E-mail: katani@tabrizu.ac.ir / shahmorad@tabrizu.ac.ir

Abstract. In this paper, we propose an efficient numerical method for solving systems of linear and nonlinear integral equations of the first and second kinds, which avoids the need for special starting values. The method has also the advantages of simplicity of application and at least six order of convergence. A convergence analysis is given and accuracy of the method is clarified by numerical examples.

Mathematical subject classification: 65R20.

Key words: Urysohn type Volterra integral equations, Romberg quadrature rule, block by block method.

1 Introduction

A number of problems in physics, engineering, biology, applied mathematics and many area of analysis, as well as other branches of science are described by system of integral equations.

Numerical solution of integral equations have attracted attention of many researchers. Considered methods are including: the methods that include applications of spline functions [2]; Runge-Kutta method [7]; Chebyshev polynomials method [1]; expansion method [9]; variational method [6]; HPM

#CAM-357/11. Received: 20/IV/11. Accepted: 13/II/12.
*Corresponding author.

(Homotopy Perturbation Method), ADM (Adomian Decomposition Method) [11] and RBFN (Radial Basis Function Network) method [3].

In this paper, we propose a block by block method with Romberg quadrature rule which is a development of the presented method in [4].

The concept of a block by block method for integral equations seems to be described for the first time by Young [10]. A similar technique for differential equations was given by Milne [8]. A block method is essentially an extrapolation procedure and has the advantage of being self starting. As we shall see, it produces a block of values at a time and it is effective for the long intervals. The block by block method with Romberg quadrature rule has the following extra advantages:

1. For the given step size h, the order of convergence is at least h^6 while it is h^4 by using Simpson rule [4].

2. By increasing number of blocks, the order of convergence increases such that it would be at least h^8 and h^{10} respectively for 8 and 16 blocks.

3. At the first step of Romberg rule one can use Simpson rule instead of trapezoidal rule, then the order of convergence for 4 blocks will be at least h^8.

The rest of the paper is organized as follows. In Section 2, we will present the method in a simple case (system of two equations) and in Section 3, we describe the general case. In Section 4, we prove a convergence result. Finally, we illustrate the performance of the method by comparing the numerical results of the HPM, RBFN method and the block by block method in Section 5 (see Table 1).

2 Description of the method

Consider a system of Volterra integral equations (VIEs) of the form

$$\mathbf{f}(x) = \mathbf{g}(x) + \int_0^x \mathbf{K}(x, s, \mathbf{f}(s))ds, \qquad 0 \leq s \leq x \leq X \tag{2.1}$$

where

$$\mathbf{f}(x) = \left[f_1(x), f_2(x), \ldots, f_n(x)\right]^T, \quad \mathbf{g}(x) = \left[g_1(x), g_2(x), \ldots, g_n(x)\right]^T,$$

and

$$\mathbf{K}(x, s, \mathbf{f}(s)) = \begin{bmatrix} k_1(x, s, f_1(s), \ldots, f_n(s)) \\ k_2(x, s, f_1(s), \ldots, f_n(s)) \\ \vdots \\ k_n(x, s, f_1(s), \ldots, f_n(s)) \end{bmatrix}.$$

We assume that the system (2.1) is uniquely solvable. Necessary and sufficient conditions for the existence and uniqueness of solution for (2.1) can be found in [5], therefore we assume

(i) $\mathbf{g}(x)$ is continuous (i.e. each component is continuous),

(ii) $\mathbf{K}(x, s, \mathbf{u})$ is a continuous function for

$$0 \le s \le x \le X \quad \text{and} \quad 0 \le \|\mathbf{u}\| < \infty,$$

(iii) the kernel satisfies the Lipschtiz condition

$$\|\mathbf{K}(x, s, \mathbf{y}) - \mathbf{K}(x, s, \mathbf{z})\| \le L\|\mathbf{y} - \mathbf{z}\|$$

where the norm is defined as $\|\mathbf{f}\| = \max_{0 \le i \le n} |f_i(t)|$.

For simplicity, let $n = 2$ and the number of blocks to be 4. For $8, 16, \ldots$ blocks the process will be similar. Also, let $0 = x_0 < x_1 < \cdots < x_N = X$ be a partition of $[0, X]$ with $x_i = x_0 + ih$ and $h = X/N$, (note that N must be multiple of the number of blocks). Then by putting $x = x_{4m+p}$ in (2.1), we have

$$\begin{aligned} f_1(x_{4m+p}) &= g_1(x_{4m+p}) + \int_0^{x_{4m}} k_1(x_{4m+p}, s, f_1(s), f_2(s))ds \\ &\quad + \int_{x_{4m}}^{x_{4m+p}} k_1(x_{4m+p}, s, f_1(s), f_2(s))ds \end{aligned} \tag{2.2}$$

$$\begin{aligned} f_2(x_{4m+p}) &= g_2(x_{4m+p}) + \int_0^{x_{4m}} k_2(x_{4m+p}, s, f_1(s), f_2(s))ds \\ &\quad + \int_{x_{4m}}^{x_{4m+p}} k_2(x_{4m+p}, s, f_1(s), f_2(s))ds \end{aligned} \tag{2.3}$$

for $m = 0, 1, 2, \ldots, N/4 - 1$ and $p = 1, 2, 3, 4$. Let $F_{i,j} \approx f_i(x_j)$ for $i = 1, 2, \ldots, n$ and $j = 0, 1, 2, \ldots, N$.

If $F_{i,0},\ F_{i,1},\ \ldots,\ F_{i,4m}$ $(i = 1, 2)$ are known, then the first integral in (2.2) and (2.3) can be approximated by standard quadrature rules. The second integral is estimated by Romberg quadrature rule at the points x_{4m}, x_{4m+1}, x_{4m+2}, x_{4m+3} and x_{4m+4}, thus a system of eight simultaneous equations is obtained that is solved for a block of eight values of F. We use the trapezoidal rule for $\int_{x_v}^{x_u} k_i(x_{4m+p}, s, \mathbf{f}(s))ds$ $(i = 1, 2)$ with $x_u = uh,\ x_v = vh(u, v = 0, \ldots, N)$ and define

$$
\begin{aligned}
T_i^{(0)}(u,v) &:= \frac{x_{u-v}}{2}\left[k_i\left(x_{4m+p}, x_v, \mathbf{F}_v\right) + k_i\left(x_{4m+p}, x_u, \mathbf{F}_u\right)\right],\\
T_i^{(1)}(u,v) &:= \frac{1}{2}T_i^{(0)}(u,v) + \frac{x_{u-v}}{2}k_i\left(x_{4m+p}, x_{\frac{u+v}{2}}, \mathbf{F}_{\frac{u+v}{2}}\right),\\
T_i^{(2)}(u,v) &:= \frac{1}{2}T_i^{(1)}(u,v) + \frac{x_{u-v}}{4}\left[k_i\left(x_{4m+p}, x_{\frac{u+3v}{4}}, \mathbf{F}_{\frac{u+3v}{4}}\right)\right.\\
&\qquad \left. + k_i\left(x_{4m+p}, x_{\frac{3u+v}{4}}, \mathbf{F}_{\frac{3u+v}{4}}\right)\right], \qquad (2.4)\\
T_i^{(3)}(u,v) &:= \frac{1}{2}T_i^{(2)}(u,v) + \frac{x_{u-v}}{8}\left[k_i\left(x_{4m+p}, x_{\frac{u+7v}{8}}, \mathbf{F}_{\frac{u+7v}{8}}\right)\right.\\
&\qquad + k_i\left(x_{4m+p}, x_{\frac{3u+5v}{8}}, \mathbf{F}_{\frac{3u+5v}{8}}\right) + k_i\left(x_{4m+p}, x_{\frac{5u+3v}{8}}, \mathbf{F}_{\frac{5u+3v}{8}}\right)\\
&\qquad \left. + k_i\left(x_{4m+p}, x_{\frac{7u+v}{8}}, \mathbf{F}_{\frac{7u+v}{8}}\right)\right]
\end{aligned}
$$

where

$$\mathbf{F}_\alpha = (F_{1,\alpha}, F_{2,\alpha}) \approx (f_1(x_\alpha), f_2(x_\alpha)) \quad \text{for} \quad \alpha = v, \frac{u+7v}{8}, \ldots, u.$$

By using the Romberg rule we define

$$
\begin{aligned}
A_i(u,v) &:= \frac{64}{45}T_i^{(2)}(u,v) - \frac{20}{45}T_i^{(1)}(u,v) + \frac{1}{45}T_i^{(0)}(u,v),\\
B_i(u,v) &:= \frac{4096}{2835}T_i^{(3)}(u,v) - \frac{1344}{2835}T_i^{(2)}(u,v)\\
&\qquad + \frac{84}{2835}T_i^{(1)}(u,v) - \frac{1}{2835}T_i^{(0)}(u,v)
\end{aligned}
$$

then the second integral in (2.2) and (2.3) can be written as

$$\int_{x_{4m}}^{x_{4m+p}} k_1(x_{4m+p}, s, \mathbf{f}(s)) \approx A_1(4m+p, 4m),$$

$$\int_{x_{4m}}^{x_{4m+p}} k_2(x_{4m+p}, s, \mathbf{f}(s)) \approx A_2(4m+p, 4m)$$

for $p = 1, 2, 3, 4$. If $\frac{jp}{4} (j = 1, 2, 3)$ are not integers, then $F_{i,4m+\frac{jp}{4}} (i = 1, 2, \ldots, n)$ will be unknown in (2.4), but they can be approximated by Lagrange interpolation at the points x_{4m}, x_{4m+1}, x_{4m+2}, x_{4m+3} and x_{4m+4}. Thus we obtain

$$\begin{aligned}
\mathbf{F}_{4m+\frac{3}{2}} &\approx \tfrac{-5}{128}\mathbf{F}_{4m} + \tfrac{15}{32}\mathbf{F}_{4m+1} + \tfrac{45}{64}\mathbf{F}_{4m+2} - \tfrac{5}{32}\mathbf{F}_{4m+3} + \tfrac{3}{128}\mathbf{F}_{4m+4}, \\
\mathbf{F}_{4m+\frac{3}{4}} &\approx \tfrac{195}{2048}\mathbf{F}_{4m} + \tfrac{585}{512}\mathbf{F}_{4m+1} - \tfrac{351}{1024}\mathbf{F}_{4m+2} + \tfrac{65}{512}\mathbf{F}_{4m+3} - \tfrac{45}{2048}\mathbf{F}_{4m+4}, \\
\mathbf{F}_{4m+\frac{9}{4}} &\approx \tfrac{35}{2048}\mathbf{F}_{4m} - \tfrac{63}{512}\mathbf{F}_{4m+1} + \tfrac{945}{1024}\mathbf{F}_{4m+2} + \tfrac{105}{512}\mathbf{F}_{4m+3} - \tfrac{45}{2048}\mathbf{F}_{4m+4}, \qquad (2.5) \\
\mathbf{F}_{4m+\frac{1}{2}} &\approx \tfrac{35}{128}\mathbf{F}_{4m} + \tfrac{35}{32}\mathbf{F}_{4m+1} - \tfrac{35}{64}\mathbf{F}_{4m+2} + \tfrac{7}{32}\mathbf{F}_{4m+3} - \tfrac{5}{128}\mathbf{F}_{4m+4}, \\
\mathbf{F}_{4m+\frac{1}{4}} &\approx \tfrac{1155}{2048}\mathbf{F}_{4m} + \tfrac{385}{512}\mathbf{F}_{4m+1} - \tfrac{495}{1024}\mathbf{F}_{4m+2} + \tfrac{105}{512}\mathbf{F}_{4m+3} - \tfrac{77}{2048}\mathbf{F}_{4m+4}.
\end{aligned}$$

When we use this method with very small step size, we must approximate the first integrals in a large interval, so we use the Romberg rule with three steps. If m is even, then

$$\int_{0}^{x_{4m}} k_1(x_{4m+p}, s, \mathbf{f}(s))ds \approx B_1(4m, 0),$$

$$\int_{0}^{x_{4m}} k_2(x_{4m+p}, s, \mathbf{f}(s))ds \approx B_2(4m, 0)$$

and we obtain from (2.2) and (2.3)

$$\begin{aligned}
F_{1,4m+p} &= g_1(x_{4m+p}) + B_1(4m, 0) + A_1(4m+p, 4m), \\
F_{2,4m+p} &= g_2(x_{4m+p}) + B_2(4m, 0) + A_2(4m+p, 4m)
\end{aligned} \qquad (2.6)$$

otherwise

$$\begin{aligned}
\int_0^{x_{4m}} k_1(x_{4m+p}, s, \mathbf{f}(s)) &= \int_0^{x_4} k_1(x_{4m+p}, s, \mathbf{f}(s))ds \\
&\quad + \int_{x_4}^{x_{4m}} k_1(x_{4m+p}, s, \mathbf{f}(s))ds \\
&\approx A_1(4,0) + B_1(4m,4), \\
\int_0^{x_{4m}} k_2(x_{4m+p}, s, \mathbf{f}(s)) &= \int_0^{x_4} k_2(x_{4m+p}, s, \mathbf{f}(s))ds \\
&\quad + \int_{x_4}^{x_{4m}} k_2(x_{4m+p}, s, \mathbf{f}(s))ds \\
&\approx A_2(4,0) + B_2(4m,4)
\end{aligned}$$

and we obtain

$$\begin{aligned}
F_{1,4m+p} &= g_1(x_{4m+p}) + A_1(4,0) + B_1(4m,4) + A_1(4m+p, 4m), \\
F_{2,4m+p} &= g_2(x_{4m+p}) + A_2(4,0) + B_2(4m,4) + A_2(4m+p, 4m).
\end{aligned} \tag{2.7}$$

Therefore for $p = 1, 2, 3, 4$, (2.6) (or (2.7)) form a system of equations with the unknowns $\mathbf{F}_{4m+1}$, $\mathbf{F}_{4m+2}$, $\mathbf{F}_{4m+3}$ and $\mathbf{F}_{4m+4}$ which will be linear and non-linear respectively for linear and nonlinear integral equations. For the linear case, it is solved via a direct method but for the nonlinear case, the system may be solved by using an iterative method or by using a suitable software package such as Maple.

3 The general process

Consider the system of VIEs

$$\mathbf{f}(x) = \mathbf{g}(x) + \int_0^x \mathbf{K}(x, s, \mathbf{f}(s))ds, \quad 0 \le s \le x \le X \tag{3.1}$$

where **f**, **g** and **K** are n-tuples vectors.

Let

$$0 = x_0 < x_1 < \cdots < x_N = X$$

be a partition of $[0, X]$ with the step size h. Similar to previous section, if m be even, then we will have

$$\begin{aligned}
F_{1,4m+p} &= g_1(x_{4m+p}) + B_1(4m, 0) + A_1(4m + p, 4m),\\
F_{2,4m+p} &= g_2(x_{4m+p}) + B_2(4m, 0) + A_2(4m + p, 4m),\\
&\vdots\\
F_{n,4m+p} &= g_n(x_{4m+p}) + B_n(4m, 0) + A_n(4m + p, 4m)
\end{aligned}$$

for $m = 0, 1, \ldots, N/4 - 1$ and $p = 1, \ldots, 4$.
Otherwise (m be odd), we obtain

$$\begin{aligned}
F_{1,4m+p} &= g_1(x_{4m+p}) + A_1(4, 0) + A_1(4m + p, 4m) + B_1(4m, 4),\\
F_{2,4m+p} &= g_2(x_{4m+p}) + A_2(4, 0) + A_2(4m + p, 4m) + B_2(4m, 4),\\
&\vdots\\
F_{n,4m+p} &= g_n(x_{4m+p}) + A_n(4, 0) + A_n(4m + p, 4m) + B_n(4m, 4)
\end{aligned}$$

or equivalently

$$\begin{bmatrix}
F_{1,4m+1} - A_1(4m + 1, 4m)\\
\vdots\\
F_{1,4m+4} - A_1(4m + 4, 4m)\\
F_{2,4m+1} - A_2(4m + 1, 4m)\\
\vdots\\
F_{n,4m+4} - A_n(4m + 4, 4m)
\end{bmatrix}
=
\begin{bmatrix}
g_1(x_{4m+1}) + A_1(4, 0) + B_1(4m, 4)\\
\vdots\\
g_1(x_{4m+4}) + A_1(4, 0) + B_1(4m, 4)\\
g_2(x_{4m+1}) + A_2(4, 0) + B_2(4m, 4)\\
\vdots\\
g_n(x_{4m+4}) + A_n(4, 0) + B_n(4m, 4)
\end{bmatrix}
\tag{3.2}$$

for $m = 0, 1, \ldots, N/4 - 1$. Consequently, at each step we get a system of $4n$ equations with unknowns $F_{1,4m+1}, \ldots, F_{1,4m+4}$, $F_{2,4m+1}, \ldots, F_{2,4m+4}, \ldots$, $F_{n,4m+1}, \ldots, F_{n,4m+4}$.

4 Convergence analysis

Theorem 4.1. *The approximation method given by the system (3.2), is convergent and its order of convergence is at least* 6.

Proof. For simplicity, we prove the theorem for $n = 2$, the general case is proved similarly. Define $\varepsilon_{i,j} := |F_{i,j} - f_i(x_j)|$, then

$$\begin{aligned}
\varepsilon_{1,4m+1} &= \left|F_{1,4m+1} - f_1(x_{4m+1})\right| \\
&= \left| h\sum_{i=0}^{4m} w_i k_1\left(x_{4m+1}, x_i, \mathbf{F}_i\right) + \frac{7}{90}x_1 k_1\left(x_{4m+1}, x_{4m}, \mathbf{F}_{4m}\right)\right. \\
&\quad + \frac{7}{90}x_1 k_1\left(x_{4m+1}, x_{4m+1}, \mathbf{F}_{4m+1}\right) + \frac{6}{45}x_1 k_1\left(x_{4m+1}, x_{4m+1/2}, \mathbf{F}_{4m+1/2}\right) \\
&\quad + \frac{16}{45}x_1 k_1\left(x_{4m+1}, x_{4m+1/4}, \mathbf{F}_{4m+1/4}\right) + \frac{16}{45}x_1 k_1\left(x_{4m+1}, x_{4m+\frac{3}{4}}, \mathbf{F}_{4m+\frac{3}{4}}\right) \\
&\quad \left. - \int_0^{x_{4m+1}} k_1(x_{4m+1}, s, \mathbf{f}(s))ds \right|.
\end{aligned}$$

By using (2.5), adding and diminishing the terms

$$h\sum_{i=0}^{4m} w_i k_1\left(x_{4m+1}, x_i, \mathbf{f}(x_i)\right),\ \frac{7}{90}x_1 k_1\left(x_{4m+1}, x_{4m}, \mathbf{f}(x_{4m})\right), \ldots,$$
$$\frac{16}{45}x_1 k_1\left(x_{4m+1}, x_{4m+\frac{3}{4}}, \frac{195}{2048}\mathbf{f}(x_{4m}) + \cdots - \frac{45}{2048}\mathbf{f}(x_{4m+4})\right)$$

and using the Lipschitz condition for k_1, we obtain

$$\begin{aligned}
\varepsilon_{1,4m+1} \;\leq\; & hc'\sum_{i=0}^{4m}(\varepsilon_{1,i}, \varepsilon_{2,i}) + hc_1(\varepsilon_{1,4m+1}, \varepsilon_{2,4m+1}) + hc_2(\varepsilon_{1,4m+2}, \varepsilon_{2,4m+2}) \\
& + hc_3(\varepsilon_{1,4m+3}, \varepsilon_{2,4m+3}) + hc_4(\varepsilon_{1,4m+4}, \varepsilon_{2,4m+4}) + R
\end{aligned}$$

where c' and $c_1, \ldots, c_4$ are constants and R is the error of the numerical integration.

Without diminishing universality, we assume that

$$\max_{j=4m+1,\ldots,4m+4} \varepsilon_{l,j} = \varepsilon_{l,4m+1},\ \max_{l=1,2} \varepsilon_{l,j} = \varepsilon_{1,j}$$

then

$$\varepsilon_{1,4m+1} \leq hc'\sum_{i=0}^{4m}\varepsilon_{1,i} + hc\varepsilon_{1,4m+1} + R,$$

where $c = c_1 + c_2 + c_3 + c_4$. Hence from Gronwall inequality [5], we have

$$\varepsilon_{1,4m+1} \leq \frac{R}{1-hc}e^{\frac{c'}{1-hc}x_n}.$$

It follow that if n approach to infinity then $\varepsilon_{1,4m+1}$ will close to the 0 and for the functions k and f with at least sixth order derivatives, we have $R = O(h^6)$ and so $\varepsilon_{1,4m+1} = O(h^6)$ and the proof is completed.

5 Numerical results

We consider the following examples to illustrate the theoretical results of Theorem 4.1 and compare numerical results of the method with the results of HPM and RBFN method.

Example 1 **([3]).** Consider the nonlinear system

$$\sin(x) - x + \int_0^x \left(f_1^2(s) + f_2^2(s)\right) ds = f_1(x),$$

$$\cos(x) - 1/2\sin^2(x) + \int_0^x f_1(s) f_2(s) ds = f_2(x),$$

with the exact solutions $f_1(x) = \sin(x)$ and $f_2(x) = \cos(x)$.

Example 2. Consider the linear system

$$e^x + \frac{1}{6} - \frac{1}{6}e^{2x} - \frac{x^4}{4} + \int_0^x \left(\frac{1}{3}e^s f_1(s) + s^2 f_2(s)\right) ds = f_1(x),$$

$$x - \frac{1}{2}x^3 + \int_0^x \left(x^2 e^{-s} f_1(s) - x f_2(s)\right) ds = f_2(x)$$

with the exact solutions $f_1(x) = e^x$ and $f_2(x) = x$.

Example 3. Consider a triple system of VIEs as

$$5e^x - 4 - 4xe^x + \int_0^x 4 f_1(s) f_2(s) ds = f_1(x),$$

$$x - 1 - xe^x + e^x - \frac{x^2}{2} + \int_0^x (f_1(s) + f_2(s))\, ds = f_2(x),$$

$$x^2 + \frac{1}{15}x^5 - \frac{1}{3}\int_0^x (s f_1(s) f_3(s))\, ds = f_3(x)$$

with the exact solutions $f_1(x) = e^x$, $f_2(x) = x$ and $f_3(x) = x^2$.

Example 4 ([3]). As a final example, we consider a non-linear system of the first kind VIEs of the form

$$\int_0^x \left(1 - x^2 + s^2\right)\left(f_1(s) + f_2^3(s)\right) ds = \frac{1}{6}x^6 + \frac{1}{5}x^5 + \frac{1}{4}(1-x^2)x^4 + \frac{1}{3}(1-x^2)x^3,$$

$$\int_0^x (5 + x - s)\left(f_1(s) - f_2(s)\right) ds = -\frac{1}{4}x^4 + \frac{1}{3}(6+x)x^3 + \frac{1}{2}(-5-x)x^2$$

with the exact solutions $f_1(x) = x^2$ and $f_2(x) = x$.

The results in Tables 1-4 show the absolute errors for the examples 1-4. All results computed by programming in Maple 11. In Table 1, we show the superiority of the block by block method by comparing its results (for $h = 0.05$) with the results of RBFN-MshA (a modified version of Shi's algorithm) [3] and HPM [11], where the results of RBFN-MshA obtained with 6 hidden nodes and the results of HPM obtained with 4 iterations. Moreover,

1. The time of computation in the block by block method is less than that in the HPM whenever programming of both method is done using Maple package. Also, according to the structure of HPM, increasing number of iterations do not affect on the precision.
2. The values of the RBF widths affect significantly on the accuracy of results and determination of them is still a challenging problem whereas the block by block method dose not need any starting values.
3. At each step of the RBFN method, the weights are updated by using an optimization method, but the block by block method is independent of using any other method. Hence the RBFN method is more complicated than the block by block method.

Tables 2 and 4 show that the block by block method is an efficient method for the large values of x, whereas other methods are useless.

6 Conclusion

In this paper, we have shown that the block by block method can achieve at least 6 order of convergence. Numerical results given in Tables 1-4 confirm this convergence order and show the high accuracy of the method. The idea can be applied to other types of integral equations and with other suitable quadrature rules.

Acknowledgements. The authors would like to thank so much the referee for carefully reading and his/her useful comments and suggestions on this paper.

x	RBFN-MSHA		HPM		Block by Block	
	e_{F_1}	e_{F_2}	e_{F_1}	e_{F_2}	e_{F_1}	e_{F_2}
0.1	$8.33e{-7}$	$9.47e{-7}$	$3.365e{-04}$	$4.864e{-03}$	$1.334e{-11}$	$1.741e{-10}$
0.2	$2.85e{-7}$	$4.37e{-7}$	$2.736e{-03}$	$1.962e{-02}$	$9.542e{-11}$	$5.105e{-11}$
0.3	$4.35e{-7}$	$7.52e{-7}$	$8.950e{-03}$	$4.394e{-02}$	$1.179e{-10}$	$2.223e{-10}$
0.4	$1.58e{-7}$	$1.41e{-7}$	$2.053e{-02}$	$7.684e{-02}$	$2.191e{-10}$	$2.116e{-10}$
0.5	$5.03e{-7}$	$8.75e{-7}$	$4.057e{-02}$	$1.174e{-01}$	$2.487e{-10}$	$1.912e{-10}$
0.6	$2.67e{-7}$	$5.93e{-7}$	$6.768e{-02}$	$1.636e{-01}$	$3.593e{-10}$	$2.953e{-10}$
0.7	$6.69e{-8}$	$6.27e{-8}$	$1.052e{-01}$	$2.150e{-01}$	$4.132e{-10}$	$2.559e{-10}$
0.8	$2.15e{-7}$	$6.08e{-7}$	$1.507e{-01}$	$2.683e{-01}$	$5.598e{-10}$	$4.687e{-10}$
0.9	$1.16e{-6}$	$3.25e{-6}$	$2.047e{-01}$	$3.224e{-01}$	$2.505e{-10}$	$2.355e{-09}$
1.0	$1.23e{-6}$	$3.70e{-6}$	$2.630e{-01}$	$3.739e{-01}$	$2.233e{-11}$	$2.323e{-09}$
time	—		$29.593''$		$1.217''$	

Table 1 – Numerical results of example 1.

x	F_1	e_{F_1}	F_2	e_{F_2}
0.125	1.13314842037	$3.270e{-08}$	0.12499999865	$1.348e{-09}$
0.250	1.28402539852	$1.817e{-08}$	0.24999999640	$3.595e{-09}$
0.375	1.45499139217	$2.245e{-08}$	0.37499998844	$1.156e{-08}$
0.500	1.64872141201	$1.413e{-07}$	0.49999999990	$1.012e{-10}$
0.625	1.86824601702	$5.959e{-08}$	0.62499997030	$2.970e{-08}$
0.750	2.11700012510	$1.085e{-07}$	0.74999997917	$2.083e{-08}$
0.875	2.39887540154	$1.076e{-07}$	0.87499995920	$4.080e{-08}$
1.000	2.71828240247	$5.740e{-07}$	1.00000003811	$3.811e{-08}$
1.125	3.08021670501	$1.439e{-07}$	1.12499994750	$5.250e{-08}$
1.250	3.49034291266	$4.480e{-08}$	1.24999998035	$1.965e{-08}$
1.375	3.95507663518	$8.774e{-08}$	1.37499992899	$7.101e{-08}$
1.500	4.48169022877	$1.158e{-06}$	1.50000011184	$1.118e{-07}$
1.625	5.07841871546	$3.217e{-07}$	1.62499991705	$8.295e{-08}$
1.750	5.75460262435	$5.165e{-08}$	1.74999999033	$9.668e{-09}$
1.875	6.52081894972	$1.706e{-07}$	1.87499989782	$1.022e{-07}$
2.000	7.38905946606	$3.367e{-06}$	2.00000025992	$2.599e{-07}$

Table 2 – Numerical results of example 2, $h = 0.125$.

x	F_1	e_{F_1}	F_2	e_{F_2}	F_3	e_{F_3}
0.1	1.1051709	$8.703e{-12}$	0.1000000	$2.188e{-12}$	0.0100000	$5.627e{-18}$
0.2	1.2214027	$1.740e{-10}$	0.2000000	$4.356e{-11}$	0.0400000	$6.047e{-16}$
0.3	1.3498588	$2.133e{-11}$	0.3000000	$4.528e{-12}$	0.0900000	$1.294e{-14}$
0.4	1.4918246	$3.402e{-10}$	0.4000000	$8.707e{-11}$	0.1600000	$4.860e{-14}$
0.5	1.6487212	$7.443e{-10}$	0.4999999	$1.737e{-10}$	0.2500000	$6.149e{-13}$
0.6	1.8221187	$4.563e{-10}$	0.5999999	$8.254e{-11}$	0.3600000	$2.050e{-12}$
0.7	2.0137527	$2.289e{-09}$	0.6999999	$4.966e{-10}$	0.4900000	$6.459e{-12}$
0.8	2.2255409	$2.745e{-09}$	0.7999999	$5.296e{-10}$	0.6400000	$1.652e{-11}$
0.9	2.4596031	$4.483e{-09}$	0.8999999	$8.952e{-10}$	0.8100000	$2.924e{-11}$
1.0	2.7182818	$6.707e{-09}$	0.9999999	$1.233e{-09}$	1.0000000	$6.738e{-11}$

Table 3 – Numerical results of example 3, $h = 0.05$.

x	F_1	e_{F_1}	F_2	e_{F_2}
0.2	0.040	0	0.200	0
0.4	0.160	0	0.400	0
0.6	0.360	0	0.600	0
0.8	0.640	$1.000e{-30}$	0.800	$1.000e{-30}$
1.0	1.000	0	1.000	$1.000e{-30}$
1.2	1.440	0	1.200	0
1.4	1.960	$1.000e{-29}$	1.400	$1.000e{-29}$
1.6	2.560	$2.000e{-29}$	1.600	$1.000e{-29}$
1.8	3.240	$5.000e{-29}$	1.800	0
2	4.000	$7.000e{-29}$	2.000	0
2.2	4.840	$1.400e{-28}$	2.200	$5.000e{-29}$
2.4	5.760	$6.600e{-28}$	2.400	$3.000e{-29}$
2.6	6.760	$1.500e{-28}$	2.600	$1.800e{-28}$
2.8	7.840	$6.300e{-28}$	2.800	$4.000e{-28}$
3	9.000	$1.380e{-27}$	3.000	$1.310e{-27}$
3.2	10.240	$3.800e{-27}$	3.200	$4.070e{-27}$
3.4	11.560	$1.280e{-26}$	3.400	$1.401e{-26}$
3.6	12.960	$4.410e{-26}$	3.600	$4.364e{-26}$
3.8	14.440	$1.935e{-25}$	3.800	$1.943e{-25}$
4	16.000	$7.883e{-25}$	4.000	$7.857e{-25}$

Table 4 – Numerical results of example 4, $h = 0.2$.

REFERENCES

[1] A. Akyüz-Dascloĝlu, *Chebyshev polynomail solutions of systems of linear integral equations.* Appl. Math. Comput., **151** (2004), 221–232.

[2] M.E. Eltom, *Application of spline functions to system of Volterra integral equation of the first and second kinds.* IMA, J. Appl. Math., **17** (1976), 295–310.

[3] A. Golbabai, M. Mammadov and S. Seifollahi, *Solving a system of nonlinear integral equations by an RBF network.* Comput. Math. Appl., **57** (2009), 1651–1658.

[4] R. Katani and S. Shahmorad, *Block by block method for the systems of nonlinear Volterra integral equations.* Appl. Math. Modelling, **34** (2010), 400–406.

[5] P. Linz, *Analytical and numerical methods for Volterra equations.* SIAM Philadelphia, (1985).

[6] H.M. Liu, *Variational approach to nonlinear electrochemical system.* Chaos, Solitons and Fractals, **23**(2) (2005), 573–576.

[7] K. Maleknejad and M. Shahrezaee, *Using Runge-Kutta method for numerical solution of the system of Volterra integral equation.* Appl. Math. Comput., **149** (2004), 399–410.

[8] W.E. Milne, *Numerical solution of differential equations.* Wiley, (1953).

[9] M. Rabbani, K. Maleknejad and N. Aghazadeh, *Numerical computational solution of the Volterra integral equations system of the second kind by using an expansion method.* Appl. Math. Comput., **187** (2007), 1143–1146.

[10] A. Young, *The application of approximate product-integration to the numerical solution of integral equations.* Proc. Roy. Soc. London. Ser. A, **224** (1954), 561–573, MR 16,179.

[11] E. Yusufoğlu (Agadjanov), *A homotopy perturbation algorithm to solve a system of Fredholm-Volterra type integral equations.* Math. and Comput. Modelling, **47**, Issues 11-12 (2008), 1099–1107.