Barycentric - Maclaurin Interpolation Method for Solving Volterra Integral Equations of the Second Kind

Shoukralla, E. S.
Prof. of Eng. Math., Faculty of Eng. and Technology, FUE in Egypt.

Elgohary, H.
Lecturer, Faculty of Electronic Engineering, Menoufia University, Egypt.

Basma Magdy
Assistant, Faculty of Eng. and Technology, FUE in Egypt.

Abstract

In this paper, the Lagrange functions of Lagrange interpolation are expanded into Maclaurin polynomials to improve the performance of an improved formula of the Barycentric Lagrange interpolation with uniformly spaced nodes and was used for solving Volterra integral equations of the second kind. For the implementation of this technique, the given data function, the kernel and the unknown function are approximated by the given improved formula to get interpolated polynomials of the same degree. Furthermore, the interpolate unknown function is represented by four matrices and is substituted twice into both sides of the considered integral equation, while the kernel is represented by five matrices. This enforcement provided the possibility to reduce the solution of the Volterra equation into an equivalent algebraic linear system in a matrix form. For showing the efficiency of this method four examples are solved. It turns out that, the obtained approximate solutions were equal to the exact ones. Moreover, it is noticed that a smaller number of nodes are applied if the given function and the kernel were algebraic functions and the upper bound of the integration domain variable was canceled. For non–algebraic given function and the kernel, the exact solutions were obtained by increasing the number of nodes and taking the upper bound of the integration domain equal to one, which ensures the accuracy and Authenticity of the presented method.

1. Introduction

Many published articles have provided innovative methods to solve linear and nonlinear, singular and non-singular integral equations of different types and kinds. These equations usually appear when solving boundary value problems via integral equations. In fact, the solution of the original boundary problems through the integral equation method reduces the required solution to the solution of an equivalent boundary integral equation, which simplifies procedures for obtaining a solution by using the traditional methods of mathematical physics; Particularly in the case of open boundaries [1-4]. In addition, the equivalent integral equation clearly illustrates the characteristics and properties of the solution of the original problems. However, this paper is devoted to the numerical solutions of linear non-singular Volterra integral equations of the second kind. Although there are many types of research and a variety of methods [5-11] to solve similar equations, solutions using Barycentric Lagrange Interpolation are almost non-existent. However, the authors in [8] present a method for the numerical solution of the linear and nonlinear high-dimensional Fredholm integral equations of the second kind by means of two barycentric interpolation collocation methods based on the barycentric Lagrange function and the barycentric rational function. In this paper, an improved version of the Barycentric Lagrange Interpolation is created. The main incentive for this method is to reformulate the mathematical formula of the Barycentric Lagrange Interpolation to establish a new improved formula in a matrix form and using it for solving Volterra integral equations of the second kind without any need to apply collocation points to get the equivalent algebraic system. Based on the fact that finding Lagrange interpolant polynomial, requires $O(n^2)$ additions and multiplication, in addition to the instability of the numerical computation [12-14], the improved formula will be created in such a manner that each Barycentric Lagrange Interpolation function is replaced by a product of two matrices the first is the constant coefficients matrix and the second is the monomial matrix. Thus, upon this idea, it is required only $O(n)$ addition and multiplication and the numerical solutions become stable as shown in the presented solved examples.

The procedure begins by interpolating the unknown function by the given adjustment Barycentric Lagrange formula to represent it as a product of three matrices, one of them is the unknown coefficient matrix. By the same way, the given data function is interpolated, while the kernel is interpolated twice with respect to its two arguments and so it is turned into a product of three matrices. Moreover, the advantages of this method are not only to simplify the calculations but also to gain access to an equivalent system of equations without any need to apply collocation points. The achievement of this idea can be summarized in representing the interpolated unknown function through the interpolated kernel and approximate data functions, and then substituting it into both sides of the considered Volterra equation. By solving the obtained equivalent algebraic system, the unknown coefficients matrix is found and thereby we get the approximate unknown function of the considered Volterra integral equations. Four examples are solved by the proposed method, and the solutions are found to be converging to the exact solutions. In addition, the given tables and the plotted graphs demonstrate the authenticity, efficiency, and accuracy of the presented new method.

2. Barycentric Maclaurin Interpolation method

Consider Volterra integral equation of the second kind

$$u(x)=f(x)-\lambda\int_a^x k(x,t)u(t)\,dt;\mathrm{a}\le x\le b<\infty \qquad (1)$$

where the Volterra operator $\mathrm{K}u=\int_a^x k(x,t)u(t)\,dt$ acting in $L_2[a,b]$. The theory of Volterra integral equations of the second kind ensures the existence and uniqueness of the square-integrable solution for each complex $\lambda\ne\infty$ such that $|\lambda|<\frac{1}{M(b-a)}$, where the kernel $k(x,t)$ is defined on the square $\{(x,t):a\le x,\ t\le b\}$ and vanishing in the triangle $a\le x<t\le b$. Moreover, it is assumed that $\max\limits_{a,x\in[a,b]} k(x,t)=N$ and $\|k(x,t)\|_2\le M<\infty$. For $\lambda=1$, we have

$$u(x)=f(x)+\int_0^x k(x,t)u(t)\,dt;\quad 0\le x\le b<\infty \qquad (2)$$

where $f(x),k(x,t)$ are given functions, and $u(x)$ is the unknown function.

Let $u(x)\in C^{n+1}[a,\mathrm{b}]$ and $\tilde{u}(x)$ be the Lagrange interpolating polynomial of degree n that interpolates $u(x)$ at the $(n+1)$ equally spaced distinct nodes $\{x_i\}_{i=0}^{n}\subset[a,b]$. By choosing a step size $h>0$ such that $h=\frac{b}{n}$ we get the required $(n+1)$ equidistant interpolation nodes $x_j=jh;j=\overline{0,n}$. The presented method begins firstly; by adjustment the Barycentric Lagrange interpolation formula in a matrix form of three matrices, and in addition to this improvement we expand each Barycentric functions matrix into Maclaurin polynomial. Using this technique, the unknown function of Eq. (2) can be replaced by the approximate one denoted by $\tilde{u}(x)$ in the following matrix form

$$[\tilde{u}(x)]=X(x)C^TWU \qquad (3)$$

Here U is the unknown coefficients column matrix, W is a square diagonal matrix, C is a known Maclaurin coefficients matrix, and $\mathrm{X}(x)$ is a known row power matrix such that

(See equations (4), (5) and (6))

In the same manner, let F denotes the known Barycentric coefficients column matrix of the given data function $f(x)$, then it can be replaced by its approximate function $\tilde{f}(x)$ in the matrix form

$$\left[\tilde{f}(x)\right]=\mathrm{X}(x)\mathrm{C}^T\mathrm{WF} \qquad (7)$$

where

$$\mathrm{F}=\begin{bmatrix}f_0 & f_0 & \dots & f_n\end{bmatrix}^T;\ f_j=f(x_j);\ j=\overline{0,n} \qquad (8)$$

Consequently, the kernel $k(x,t)$ can be approximated twice with respect to its two variables x,t to get

$$\left[\tilde{k}(x,t)\right]=\mathrm{X}(x)\mathrm{C}^T\mathrm{KCX}^T(t) \qquad (9)$$

where the square matrix K is given by

(See equations (10))

Subediting by $\tilde{k}(x,t)$, and $\tilde{u}(t)$ into the right side of Eq. (2), we get: (See equations (11)).

$$\mathrm{U}=\begin{bmatrix}u_o & u_1 & \dots & u_n\end{bmatrix}^T,\mathrm{W}=diag[\gamma_0,\gamma_1,\dots,\gamma_n],\mathrm{C}=\left[c_{ji}\right]_{i,j=0}^{n}, \qquad (4)$$

$$\mathrm{X}(x)=\begin{bmatrix}1 & x & \dots & x^n\end{bmatrix}$$

where

$$c_{ji}=\frac{\psi_j^{(i)}(0)}{i!}\ \forall i,j=\overline{0,n};\ \psi_j(x)=\frac{\zeta_j(x)}{\tilde{\phi}(x)}\ \forall\ j=\overline{0,n} \qquad (5)$$

and

$$\tilde{\phi}(x)=\sum_{j=0}^{n}\gamma_j\xi_j(x);\quad \gamma_j=(-1)^j\binom{n}{j};j=\overline{0,n};\ \xi_j(x)=\frac{1}{x-x_j};\ j=\overline{0,n} \qquad (6)$$

$$\mathrm{K}=\left[\gamma_{ij}k_{ij}\right]_{\underline{i,j=0}}^{n};\ k_{ij}=k\left(x_i,t_j\right);\gamma_{ij}=\gamma_i\times\gamma_j;x_i=ih,\ t_j=jh;i,j=\overline{0,n} \tag{10}$$

$$\left[\tilde{u}(x)\right]=\left[f(x)\right]+\int_0^x \mathrm{X}(x)\mathrm{C}^T\mathrm{KCX}^T(t)\mathrm{X}(t)\mathrm{C}^T\mathrm{WU}dt \tag{11}$$

$$\Phi(x)=\left[\phi_{ij}(x)\right]_{i,j=0}^{n};\phi_{ij}(x)=\int_0^x t^{i+j}dt=\frac{x^{i+j+1}}{i+j+1};\ i,j=\overline{0,n} \tag{14}$$

$$\mathrm{X}(x)\mathrm{C}^T\mathrm{KC}\Phi(x)\mathrm{C}^T\mathrm{WU}-\mathrm{X}(x)\mathrm{C}^T\mathrm{KC}.\tilde{\Phi}(x)\mathrm{C}^T\mathrm{WU} =\mathrm{X}(x)\mathrm{C}^T\mathrm{KC}\Phi(x)\mathrm{C}^T\mathrm{WF} \tag{15}$$

$$\mathrm{U}=\left(\Phi(x)\mathrm{C}^T\mathrm{W}-\tilde{\Phi}(x)\mathrm{C}^T\mathrm{W}\right)^{-1}\Phi(x)\mathrm{C}^T\mathrm{WF} \tag{17}$$

Now, let

$$\tilde{\mathrm{X}}(t)=\mathrm{X}^T(t)\mathrm{X}(t)=\left[t^{i+j}\right]_{i,j=0}^{n};i,j=\overline{0,n} \tag{12}$$

Then Eq. (11) becomes

$$\left[\tilde{u}(x)\right]=\left[f(x)\right]+\mathrm{X}(x)\mathrm{C}^T\mathrm{KC}\Phi(x)\mathrm{C}^T\mathrm{WU} \tag{13}$$

where (See equations (14)).

Moreover, by replacing $\tilde{u}(x)$ defined by Eq. (13) into the both sides of Eq. (2) we find that (See equations (15)).

(15)

where

$$\tilde{\Phi}(x)=\int_0^x \tilde{\mathrm{X}}(t)\mathrm{C}^T\mathrm{KC}\Phi(t)dt \tag{16}$$

Simplifying Eq. (16) yields (See equations (17)).

Thus, we find that the solution of the system given by Eq. (17), yields the unknown Barycentric coefficients column matrix U, and thereby the interpolated unknown function $\tilde{u}(x)$ can be found.

3. Computational results

Four examples are given to illustrate the performance of the presented method. The computations have been done by using MATLAB Version 2016. From examples 1, 2 where the kernels and the given data functions are analytic algebraic functions, it turns out that the numerical solutions are obtained in explicit and finite formulas equal to the exact solutions. From examples 3,4 where the kernel and the given data functions are transcendental functions it turns out that the obtained numerical solutions are of higher accuracy and strongly converge to the exact solutions. It is noticed that, for any value of x, where $0\le x\le b<\infty$ the real number b is canceled during the Workflow of the solution's computation regardless of the choice of the step-size h, the smoothness of the kernel or the given data function.

Example (1)

Consider Volterra integral equation

$$u(x)=x^2+\frac{1}{12}x^4+\int_0^x(t-x)u(t)dt \tag{18}$$

whose exact solution is given by $u(x)=x^2$ [15], Using equation (17) we find that the unknown coefficients matrix U for $n=4$ is given by

$$\mathrm{U}=\begin{bmatrix}0 & \frac{b^2}{16} & \frac{b^2}{4} & \frac{9b^2}{16} & b^2\end{bmatrix}^T \tag{19}$$

and by substituting U into equation (3), the parameter b was canceled and the obtained numerical solution is found to be $\tilde{u}(x)=x^2$ which equals the exact one.

Example (2)

Consider Volterra integral equation

$$u(x)=6x-x^3+\frac{1}{2}\int_0^x tu(t)dt \tag{20}$$

whose exact solution is given by $u(x)=6x$ [15], Using equation (17) we find that the unknown coefficients matrix for $n=3$ is given by

$$\mathrm{U}=\begin{bmatrix}0 & 2b & 4b & 6b\end{bmatrix}^T \tag{21}$$

and by substituting U into equation (3), the parameter b was canceled and the obtained numerical solution is found to be $\tilde{u}(x)=6x$ which equals the exact one.

Table 3.1: A comparison between the exact solution $u(x)$ and the obtained numerical solutions $\tilde{u}_n(x)$ for $n=\overline{1,7}$.

x_i	$u(x_i)$	$\tilde{u}_1(x_i)$	$\tilde{u}_2(x_i)$	$\tilde{u}_3(x_i)$
0	1	1	1	1
0.1	0.990049833749168	0.968050347308118	0.965096305667170	0.988178473645131
0.2	0.960789439152323	0.934092478088200	0.928118475157479	0.958942383177158
0.3	0.913931185271228	0.896281806357759	0.886615276672535	0.913383573359188
0.4	0.852143788966211	0.853001926140921	0.838473730638968	0.853093508386680
0.5	0.778800783071405	0.802941876883807	0.782021261784757	0.780349487999808
0.6	0.697676326071031	0.745186603608021	0.716150010913991	0.698236287516647
0.7	0.612626394184416	0.679304059941753	0.640443209534905	0.610676356016801
0.8	0.527292424043049	0.605409739908463	0.555278992272820	0.522344123621700
0.9	0.444858066222941	0.524190280411857	0.461885146424586	0.438449774594771
1	0.367879441171442	0.436873273418402	0.362320969443342	0.364395266601207

x_i	$\tilde{u}_4(x_i)$	$\tilde{u}_5(x_i)$	$\tilde{u}_6(x_i)$	$\tilde{u}_7(x_i)$
0	1	1	1	1
0.1	0.990939275505837	0.990164089651141	0.990037561529195	0.990047676065065
0.2	0.961107565361279	0.960789174086447	0.960792540339213	0.960790403405768
0.3	0.913706636330270	0.913889128796197	0.913932829453292	0.913931021656491
0.4	0.851860784514063	0.852144354186609	0.852142094909402	0.852143575920709
0.5	0.778786940171890	0.778834418636473	0.778800911920815	0.778801108367714
0.6	0.697885514536014	0.697681724406560	0.697677859968795	0.697676059667288
0.7	0.612745046454548	0.612596826541250	0.612625506983340	0.612626076109454
0.8	0.527056408788599	0.527331391492625	0.527290421867404	0.527292939160052
0.9	0.444440392353419	0.445064982676276	0.444858871252218	0.444854760897668
1	0.368205007251191	0.367973730488135	0.367868437935520	0.367877867570473

Table 3.2: The absolute errors $E_n\left(x_i\right)$ for $n=\overline{1,7}$ respectively.

x_i	$E_1\left(x_i\right)$	$E_2\left(x_i\right)$	$E_3\left(x_i\right)$
0	0	0	0
0.1	0.0219994864410504	0.0249535280819984	0.00187136010403710
0.2	0.0266969610641231	0.0326709639948446	0.00184705597516510
0.3	0.0176493789134693	0.0273159085986933	0.000547611912040669
0.4	0.000858137174709284	0.0136700583272439	0.000949719420469064
0.5	0.0241410938124018	0.00322047871335240	0.00154870492840298
0.6	0.0475102775369894	0.0184736848429600	0.000559961445615431
0.7	0.0666776657573371	0.0278168153504884	0.00195003816761563
0.8	0.0781173158654140	0.0279865682297717	0.00494830042134842
0.9	0.0793322141889163	0.0170270802016446	0.00640829162817008
1	0.0689938322469598	0.00555847172810048	0.00348417457023570

$E_4\left(x_i\right)$	$E_5\left(x_i\right)$	$E_6\left(x_i\right)$	$E_7\left(x_i\right)$
0	0	0	0
0.000889441756668385	0.000114255901972449	1.22722199727976e-05	2.15768410327666e-06
0.000318126208955549	2.65065876248194e-07	3.10118688984229e-06	9.64253444646346e-07
0.000224548940958580	4.20564750308605e-05	1.64418206349826e-06	1.63614737203055e-07
0.000283004452148350	5.65220397819033e-07	1.69405680972012e-06	2.13045502350795e-07
1.38428995151152e-05	3.36355650683018e-05	1.28849410141463e-07	3.25296309333645e-07
0.000209188464982524	5.39833552892777e-06	1.53389776424984e-06	2.66403742954502e-07
0.000118652270131414	2.95676431658620e-05	8.87201076382738e-07	3.18074962035730e-07
0.000236015254449495	3.89674495763215e-05	2.00217564438887e-06	5.15117003385690e-07
0.000417673869522017	0.000206916453335260	8.05029276740310e-07	3.30532527303973e-06
0.000325566079748929	9.42893166928083e-05	1.10032359226797e-05	1.57360096902925e-06

Table 4.1: A comparison between the exact solution $u(x)$ and the obtained numerical solutions $\tilde{u}_n(x)$ for $n=\overline{1,7}$.

x_i	$u(x_i)$	$\tilde{u}_1(x_i)$	$\tilde{u}_2(x_i)$	$\tilde{u}_3(x_i)$	$\tilde{u}_4(x_i)$	$\tilde{u}_5(x_i)$	$\tilde{u}_6(x_i)$	$\tilde{u}_7(x_i)$
0	1	1	1	1	1	1	1	1
0.1	1.11622262725640	1.21533188814540	1.11476788201315	1.11613430515533	1.11633736248470	1.11625829217227	1.11622185745171	1.11621974396128
0.2	1.27025886848658	1.46447076966676	1.26457190300607	1.26991838048057	1.27051329380398	1.27032301025571	1.27025794784522	1.27025553101232
0.3	1.47134610025784	1.75243134084269	1.45877445591563	1.47069377062370	1.47164373498000	1.47140582087768	1.47134456027996	1.47134270332222
0.4	1.73051664926387	2.08452153477496	1.70850780207946	1.72962220303298	1.73079080916449	1.73057282787897	1.73051502846970	1.73051227444740
0.5	2.06090158837516	2.46612229196902	2.02704799459443	2.05985116830124	2.06116957987742	2.06098150303398	2.06090030093477	2.06089653537330
0.6	2.47808156853109	2.90233526466490	2.43014761985728	2.47668097626638	2.47841312858372	2.47819138265434	2.47807865985192	2.47807570084991
0.7	3.00049153413101	3.39747124496145	2.93621840274771	2.99777419930890	3.00090496581660	3.00058560405535	3.00048739374878	3.00048394139057
0.8	3.64988712272765	3.95437068100348	3.56618946350098	3.64355264198063	3.65021832846967	3.64993379386447	3.64988875477847	3.64987904742878
0.9	4.45188163119408	4.57358679617322	4.34277993773981	4.43810960064105	4.45173140894547	4.45203453516188	4.45187604273048	4.45187166064407
1	5.43656365691809	5.25252525252525	5.28882522687427	5.41126116976399	5.43549981619957	5.43708812145705	5.43649564155370	5.43653854380627

Table 4.2: The absolute errors $E_n\left(x_i\right)$ for $n=\overline{1,7}$ respectively.

x_i	$E_1\left(x_i\right)$	$E_2\left(x_i\right)$	$E_3\left(x_i\right)$	$E_4\left(x_i\right)$	$E_5\left(x_i\right)$	$E_6\left(x_i\right)$	$E_7\left(x_i\right)$
0	0	0	0	0	0	0	0
0.1	0.0991092608889950	0.00145474524325651	8.83221010747892e-05	0.000114735228297524	3.56649158643751e-05	7.69804697986487e-07	2.88329512088126e-06
0.2	0.194211901180181	0.00568696548050451	0.000340488006010720	0.000254425317398832	6.41417691327195e-05	9.20641358037599e-07	3.33747425651509e-06
0.3	0.281085240584849	0.0125716443422179	0.000652329634147852	0.000297634722160600	5.97206198351685e-05	1.53997788232552e-06	3.39693562323440e-06
0.4	0.354004885511090	0.0220088471844180	0.000894446230893964	0.000274159900615523	5.61786150936605e-05	1.62079417243000e-06	4.37481647841409e-06
0.5	0.405220703593856	0.0338535937807261	0.00105042007392164	0.000267991502257647	7.99146588206234e-05	1.28744039562534e-06	5.05300186048885e-06
0.6	0.424253696133806	0.0479339486738120	0.00140059226471623	0.000331560052625690	0.000109814123250729	2.90867916907800e-06	5.86768118404635e-06
0.7	0.396979710830435	0.0642731313833029	0.00271733482210612	0.000413431685584609	9.40699243385801e-05	4.14038223395252e-06	7.59274043593550e-06
0.8	0.304483558275836	0.0836976592266647	0.00633448074701670	0.000331205742020835	4.66711368178885e-05	1.63205081804563e-06	8.07529886603930e-06
0.9	0.121705164979139	0.109101693454271	0.0137720305530280	0.000150222248604948	0.000152903967796547	5.58846360032561e-06	9.97055001050740e-06
1	0.184038404392838	0.147738430043821	0.0253024871540992	0.00106384071852439	0.000524464538960601	6.80153643894954e-05	2.51131118265135e-05

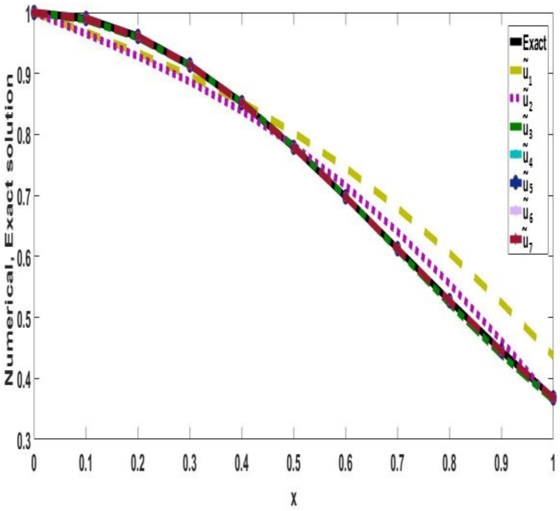

Fig 3.1 The exact solution and the obtained numerical solutions for $b=1$ *and* $n=\overline{1,7}$.

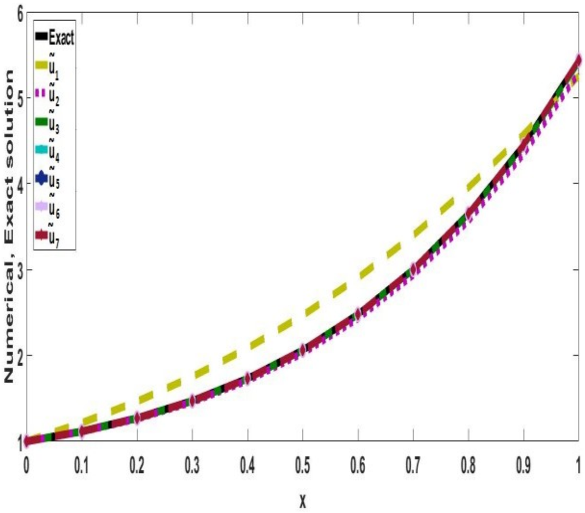

Fig. 4.1 The exact solution and the obtained numerical solution for $b=1$ *and* $n=\overline{1,7}$.

Example (3)

Consider Volterra integral equation

$$u(x)=e^{-x^2}+\frac{x\left(1-e^{-x^2}\right)}{2}-\int_0^x xt\,dt \qquad (22)$$

Whose exact solution is given by $u(x)=e^{-x^2}$ [10]. Using equations (3), (17), it is found that the obtained numerical solutions $\tilde{u}_n(x_i)$ for $n=7, b=1$ are strongly converge to the exact one. The obtained numerical solutions for $n=\overline{1,7},\ b=1$ are shown in table 3.1. The absolute errors $E_n(x_i)$ for $n=\overline{1,7},\ b=1$ are given in table 3.2. In Figure 3.1, plotted are the graphs of the exact solution $u(x)$ and the obtained numerical solutions $\tilde{u}_n(x_i)$ for $n=\overline{1,7},\ b=1$.

Example (4)

Consider Volterra integral equation

$$u(x)=1+x^2+\int_0^x \frac{1+x^2}{1+t^2}dt \qquad (23)$$

Whose exact solution is given by $u(x)=e^x(1+x^2)$ [11]. Using equations (3), (17), it is found that the obtained numerical solutions $\tilde{u}_n(x_i)$ for $n=7, b=1$ are strongly converge to the exact one. The obtained numerical solutions for $n=\overline{1,7},\ b=1$ respectively are shown in table 4.1.

The absolute errors $E_n(x_i)$ for $n=\overline{1,7},\ b=1$ are given in table 4.2. In Figure 4.1, plotted are the graphs of the exact solution $u(x)$ and the obtained numerical solutions $\tilde{u}_n(x_i)$ for $n=\overline{1,7},\ b=1$

.

4. Conclusion

The Lagrange functions with uniformly spaced interpolation nodes are expanded into Maclaurin polynomials. Therefore, an adjustment and improved Barycentric Lagrange - Maclaurin formula was established in a matrix form. This new formula was used to solve Volterra integral equations of the second kind. The kernel is interpolated with respect to both two variables. Thus, it is transformed into a product of five matrices, where only one matrix dependent on the given kernel. The given data function is also interpolated by the new Barycentric Lagrange- Maclaurin interpolating polynomial of the same degree as well as the kernel. By the same way, the unknown function is interpolated and represented by four matrices and was substituted twice into the integral equation. Thus, the solution of the considered Volterra equation is found to be equivalent to the solution of a linear system in matrix form. The illustrated numerical examples show the efficiency and genuineness of the given method. Moreover, it turns out that the numerical solutions are obtained in explicit formulas that are equal to the exact solutions for algebraic kernel and given data functions. If the kernel and the given data functions are transcendental functions the solutions were found of higher accuracy and strongly converge to the exact solutions.

Reference

[1] E. S. Shoukralla and M. A. Markos, "*The economized monic Chebyshev polynomials for solving weakly singular Fredholm integral equations of the first kind*", Vol.12, No.1, pp.1-10, October 2018.

[2] Shidong Jiang and Vladimir Rokhlin, "*Second kind integral equations for the classical potential theory on open surfaces II*", Journal of Computational Physics, Vol.195, No.1, pp.1-16, 2004.

[3] E. S. Shoukralla, M. Kamel, and M. A. Markos, "*A new computational method for solving weakly singular Fredholm integral equations of the first kind*", The 13th IEEE International Conference on Computer Engineering and Systems (ICCES 2018), Cairo, Egypt, pp. 202-207 December2018.

[4] Pius Kumar and G. C. Dubey, "*An application of Volterra integral equation by expansion of Taylor's series in the problem of heat transfer and electrostatics*", IOSR Journal of Mathematics (IOSR-JM), Vol.11, No. 5, pp.59-62, Sep-Oct 2015.

[5] Nasibeh Mollahasani and Mahmoud Mohseni Moghadam, "*Two new operational methods for solving a kind of fractional Volterra integral equations*", Asian-European Journal of Mathematics, Vol.9, No.2, pp.1-13, 2016.

[6] V. Myrhorod, and I. Hvozdeva, "*On one solution of Volterra integral equations of second kind*", American Institute of Physics conference AIP conference, (2016).

[7] Sudhanshu Aggarwal, Nidhi Sharma and Raman Chauhan, *"Solution of Linear Volterra Integral Equations of Second Kind Using Mohand Transform*", International Journal of Research in Advent Technology, Vol.6, No.11, pp. 3098- 3102, November 2018.

[8] Hongyan Liu, Jin Huang, Yubin Pan, Jipei Zhang, *"Barycentric interpolation collocation methods for solving linear and nonlinear high-dimensional Fredholm integral equations*", Journal of Computational and Applied mathematics, Volume 327, pp.141-154, January 2018.

[9] Zahra Masouri, "*Numerical expansion-iterative method for solving second kind Volterra and Fredholm integral equations using block -pulse functions*", Advanced Computational Techniques in Electromagnetics, Vol. 2012, pp.1-7, 2012.

[10] K. Maleknejad , H. Derili, " *Numerical solution of integral equations by using combination of Spline-collocation method and Lagrange interpolation*", Applied Mathematics and Computation, Vol .175, No.2, pp. 1235–1244, 2006.

[11] Weiming Wang, "*A mechanical algorithm for solving the Volterra integral equation", Applied Mathematics and Computation*", Vol.172, No.2, pp. 1323-1341, 2006.

[12] Jean-Paul Berrut, Lloyd N Trefethen, "*Barycentric Lagrange Interpolation*", Society for Industrial and Applied Mathematics, Vol. 46, No. 3, pp. 501-517, 2004.

[13] N. J. Higham, "*The numerical stability of Barycentric Lagrange interpolation*", *IMA Journal of Numerical Analysis*, Vol.24, No.4, pp.547-556, October2004.

[14] K Jing, N Kang and G Zhu, *"Convergence rates of a family of barycentric osculatory rational interpolation*", Journal of Applied Mathematics and Computing, Vol.53, No.1-2, February2017.

[15] Abdul-Majid Wazwaz, *"*A *First Course in Integral Equations - Solutions Manual, 2nd ed*", World Scientific Publishing Co. Pte. Ltd, May 2015.