Numerical Solutions of Volterra Integral Equations of the Second Kind using Lagrange interpolation via the Vandermonde matrix

E S Shoukralla¹ and B M Ahmed²*

¹ Department of Engineering Mathematics and Physics, Faculty of Electronic Engineering, Menoufia University, El Mostashfa Street, Menouf, Egypt.
² Department of Engineering Mathematics and Physics, Faculty of Engineering and Technology, Future University in Egypt, End of 90th Street, Fifth Settlement, New Cairo, Cairo, Egypt.

Corresponding author*: Basma.magdy@fue.edu.eg

Abstract. A new method is established for solving Volterra integral equations of the second kind using Lagrange interpolation through the Vandermonde approach. The goal is to minimize the interpolation errors of the high-degree polynomials on equidistance interpolations by redefining the original Lagrange functions in terms of the monomial basis. Accordingly, the complexity of the calculations is significantly reduced, and time is saved. To achieve this, the given data and the unknown functions are interpolated using Lagrange polynomials of the same degree via the Vandermonde matrix. Moreover, the interpolant unknown function is substituted twice into both sides of the integral equation so that the solution is reduced to an equivalent matrix equation without any need to apply collocation points. The error norm estimation is proved to be equal to zero. It was found that the obtained Vandermonde numerical solutions were equal to the exact ones, the calculation time was remarkably reduced, the round-off error was significantly reduced and the problematics due to the high-degree interpolating polynomial was completely faded regardless of whether the given functions were analytical or not. Thus, interpolation via the Vandermonde matrix ensures the accuracy, efficiency and authenticity of the presented method.

Keywords: Integral Equations, Volterra, Second Kind. Lagrange, Interpolation. Vandermonde matrix, Computational Methods.

1. Introduction

Integral equations of the second kind arise in scattering, acoustics, elasticity, fluid mechanics, irradiative transfer and many other engineering problems where the independent variable varies in the time or energy domain. There are many published methods [1-3] for solving integral equations of various types and kinds, singular and non-singular. It makes sense that these methods can be applied when solving Volterra equations, but here we attempt to find another approach. In fact, there are various published techniques and methods to solve Volterra integral equations of the second kind [4-6]. Some papers introduce a solution using Lagrange interpolations [7,8], while the focus of this paper is on constructing a new straightforward method for solving Volterra integral equations of the second kind with the strategical idea of minimizing the error caused by increasing the degree of the interpolated polynomials, particularly in the case of equally spaced nodes and near the end point of the
integration domain. The presented method is based on Lagrange interpolation through the Vandermonde approach [9], which gives the possibility to transform the Lagrange basis functions into the monomial basis. Thus, the problematic situation which arises when the degree of the interpolant polynomial increases, causing oscillatory polynomial, is treated.

Upon the implementation of this technique, each unknown function and the given data function were replaced by a product of three matrices, while the kernel remained without interpolation. The advantages of this method are not only to simplify the calculations, reduce the errors and save time but also to gain access to an equivalent matrix equation, from which the unknown coefficients of the unknown function are found without any need to apply collocation points. This achievement becomes a reality through the substitution of the approximate unknown function two consecutive times into both sides of the integral equation in order to obtain a simple matrix equation. Consequently, the obtained matrix equation is solved using the method of undetermined coefficients, and thereby the unknown interpolate coefficients of the unknown function can be found without any need to apply the collocation points technique. Two examples are solved by the proposed method, and the solutions are found to be equal to the exact ones regardless of whether the given function and the unknown function are analytic functions or not, and regardless of whether the kernel is a degenerate type kernel or not.

2. Methodology

2.1. Vandermonde Lagrange Interpolation

Consider Volterra integral equation of the second kind

$$u(x) = f(x) + \int_0^x k(x,t)u(t)dt; 0 \leq x \leq b < \infty.$$  \hspace{1cm} (1)

where \( f(x) \) and \( k(x,t) \) are given functions, and \( u(x) \) is the unknown function. Moreover, it is assumed that \( \|k(x,t)\|_2 \leq M < \infty \), \( u(x) \in C^{n+1}[0,b] \) and \( \max_{x \in [0,b]} f(x) = N \); \( N \) is a real number. Let \( \tilde{u}(x) \) be the Lagrange interpolating polynomial of degree \( n \) that interpolates \( u(x) \) at the \( (n+1) \) equally spaced distinct nodes \( \{x_i\}_{i=0}^n \subseteq [0,b] \). By choosing a step size \( h > 0 \) such that \( h = \frac{b}{n} \) we can get \( (n+1) \) equidistant interpolation nodes \( x_i = ih; \; i=\overline{0,n} \). Then we have

$$\tilde{u}(x) = \sum_{j=0}^n a_j x^j; \ j=0,n; \ \tilde{u}(x_i) = u(x_i) = u_i \quad \forall \; x_i = i\bigg(\frac{b}{n}\bigg); \ i=\overline{0,n}. \hspace{1cm} (2)$$

Let \( U = [a_j]_{j=0}^n \) is the unknown coefficients row matrix, then we get \( \tilde{u}(x) \) in the matrix form

$$\tilde{u}(x) = U P(x); \ P(x) = \left[ x^j \right]_{j=0}^n.$$ \hspace{1cm} (3)

where, \( P(x) \) are a monomial column matrix and the coefficients \( \{a_j\}_{j=0}^n \) can be determined by solving the linear algebraic system of equations

$$X U = A; \ X = [b_{ij}]_{i,j=0}^n; \ b_{ij} = x^j; \ b_{00} = 1; \ A = [u(x_i)]_{i=0}^n.$$ \hspace{1cm} (4)

Here \( X \) is the Vandermonde square matrix and \( A \) denotes the unknown coefficients row matrix. By the virtue of Eq. (3) we find \( \tilde{u}(x) \) in the matrix form
\[ \tilde{u}(x) = \Lambda^T \tilde{X} P(x); \quad \tilde{X} = \left(X^{-1}\right)^T. \]  

(5)

Now, replacing \( \tilde{u}(t) \) subjected to Eq. (5) with \( u(t) \) in the right side of Eq. (1) we find that

\[ [u(x)] = [f(x)] + \int_0^x \Lambda^T \tilde{X} P(t) k(x,t) dt = [f(x)] + \Lambda^T \tilde{X} \Phi(x). \]

(6)

where

\[ \Phi(x) = [\phi_i (x)]_{i=0}^n; \quad \phi_i (x) = \int_0^x t^i k(x,t) dt; \quad 0 \leq x \leq b. \]

(7)

From Eqs. (1), (6) we find that

\[ \Lambda^T \tilde{X} \Phi(x) = \int \left[ f(t) + \Lambda^T \tilde{X} \Phi(t) \right] k(x,t) dt. \]

(8)

Like \( \tilde{u}(x) \), the given data function \( f(t) \) can be interpolated in the matrix form

\[ \left[ \tilde{f}(t) \right] = F^T \tilde{X} P(t); F = [f(t_i)]_{i=0}^n; t_i = i\left(\frac{b}{n}\right); i = 0, n. \]

(9)

Consequently, replacing \( f(t) \) of Eq. (8) by its interpolant polynomial \( \tilde{f}(t) \) of Eq. (9) we get

\[ \Lambda^T \tilde{X} \left( \Phi(x) - \tilde{\Phi}(x) \right) = F^T \tilde{X} \Phi(x). \]

(10)

where

\[ \tilde{\Phi}(x) = [\tilde{\phi}_i (x)]_{i=0}^n; \quad \tilde{\phi}_i (x) = \int_0^x \phi_i (t) k(x,t) dt. \]

(11)

Due to the possibility of the matrix \( \tilde{X} \) with monomial basis to be ill-conditioned, particularly for large \( n \), it is necessary to impose the following condition

\[ \| \tilde{X} \|_2 = \sqrt{\rho(\tilde{X}^T \tilde{X})} < 1. \]

(12)

Where \( \rho(\tilde{X}) \) is the spectral radius of the matrix \( \tilde{X} \). Thus, by solving the matrix equation (10) by using the method of undetermined coefficients, the unknown coefficients matrix \( \Lambda^T \) can be calculated, and thereby the unknown function \( u(x) \) of Eq. (1) can be found using Eq. (5).

2.2. Error Norm Estimation

Rewrite Eq. (1) in the form \( u = Tu \), where the operator \( T \) is defined by \( Tu = f + Ku \), where

\[ Ku = \int_0^x k(x,t) u(t) dt \]

is the Volterra operator acting in \( L_2(0,x];0 \leq x \leq b. \) sample the total error norm of the given approximation by \( E_n(x) \) such that \( E_n(x) = \| Tu - \tilde{Tu} \|_2 \) where \( \| \cdot \|_2 \) denotes the Euclidean norm in \( \mathbb{R}^2 \), where \( T\tilde{u} = \tilde{f} + K\tilde{u} \), then we have
\[ \|Tu - T\tilde{u}\|_2 \leq \left( \|f - \tilde{f}\|_2 + (Ku - K\tilde{u}) \right) \leq \|f - \tilde{f}\|_2 + \|Ku - K\tilde{u}\|_2. \]  

Here, we have

\[ \|f - \tilde{f}\|_2 = \left[ \int_a^b |f(x)|^2 \, dx + \int_a^b |\tilde{f}(x)|^2 \, dx - 2\int_a^b |f(x)| \tilde{f}(x) \, dx \right] \left(1 + \frac{1}{2}\right). \]  

Assume that

\[ \int_0^b |f(x)|^2 \, dx \leq \eta; \int_0^b |\tilde{f}(x)|^2 \, dx = \int_0^b |\tilde{f}(x)|^2 \, dx \leq r_2. \]  

where \( \eta, r_2 \) real numbers. Since

\[ \int_a^b |f(x)| \tilde{f}(x) \, dx \leq \frac{1}{2} \int_a^b |f(x)|^2 \, dx + \frac{1}{2} \int_a^b |\tilde{f}(x)|^2 \, dx = \frac{1}{2} (\eta + r_2). \]  

Then, we find that \( \|f - \tilde{f}\|_2 = 0 \). In the same context, we have

\[ \|Ku - K\tilde{u}\|_2 = \left[ \int_0^x k(x,t)u(t) \, dt - \int_0^x k(x,t)\tilde{u}(t) \, dt \right] \left[ \int_0^x k(x,t)u(t) \, dt - \int_0^x k(x,t)\tilde{u}(t) \, dt \right] \left(1 + \frac{1}{2}\right). \]  

By Cauchy – Bunyakowski inequality, we have

\[ \int_0^t \int_0^x k(x,t)u(t) \, dt \, dx \leq \int_0^t \left[ k(x,t) \right]_2^2 \, dx = \left[ k(x,t) \right]_2^2 \int_0^x \left[ k(x,t) \right]_2^2 \, dx \]  

\[ = \frac{(Mt)^2}{2} \left[ k(x,t) \right]_2^2 ; 0 \leq t \leq b. \]  

And

\[ \int_0^t \int_0^x k(x,t)\tilde{u}(t) \, dt \, dx \leq \int_0^t \left[ k(x,t) \right]_2^2 \, dx = \left[ k(x,t) \right]_2^2 \int_0^x \left[ k(x,t) \right]_2^2 \, dx \]  

\[ = \frac{(Mt)^2}{2} \left[ k(x,t) \right]_2^2 ; 0 \leq t \leq b. \]  

Since,

\[ \int_0^t \int_0^x k(x,t)\tilde{k}(x) \, dt \, dx \leq \frac{1}{2} \int_0^t \int_0^x k(x,t) \, dt \, dx + \frac{1}{2} \int_0^t \int_0^x \tilde{k}(x,t) \, dt \, dx \]  

\[ = \frac{(Mt)^2}{2} \left[ k(x,t) \right]_2^2 ; 0 \leq t \leq b. \]
Then

\[
-2 \int_0^x k(x,t)u(t)\,dt \times \int_0^x k(x,t)\tilde{u}(t)\,dt \leq -\int_0^1 \|k(x,t)u(t)\|\,dx - \int_0^1 \|k(x,t)\tilde{u}(t)\|\,dx
\]

\[
= -\frac{(Mt)^2}{2} \|u(t)\|^2 - \frac{(Mt)^2}{2} \|\tilde{u}(t)\|^2 .
\]  

(21)

Therefore, and after substituting from Eqs. (18), (19), and (21), we find that \( \|Ku - \tilde{K}\tilde{u}\| = 0 \). Thus, we have proved that

\[ E_n(x) = \|T u - T\tilde{u}\|_2 = 0 . \]  

(22)

3. Computational results

The computations were made using MATLAB Version 2018. For the two illustrated examples, the degrees of the interpolant polynomial \( \tilde{u}_n(x) \) are chosen equal to \( n=5,10 \), and 15 respectively, while \( b=1 \). Moreover, it is found that \( \|\tilde{X}\|_2 < 1 \) for the given two examples. For example 1, where the kernel and the given data functions are analytic algebraic functions, the numerical Vandermonde solutions are equal to the exact solutions. For example 2, where the kernel and the given data functions are non-algebraic functions, the solutions are of high accuracy and strongly converge to the exact solutions rather close to the end point with average absolute error \( \leq 10^{-13} \) and a CPU time of \( \leq 50 \) seconds for \( n=15 \). Superior to the solutions obtained by other methods.

Example (1)

Consider Volterra integral equation

\[
u(x) = 1 + x + \int_0^x (x-t)u(t)\,dt.
\]  

(23)

whose exact solution is given by \( u(x) = e^x \) [5]. The obtained solution \( \tilde{u}(x) \) equals the exact one. The Vandermonde numerical solutions \( \tilde{u}_n(x) \) and the absolute errors \( E_n(x) \) are shown in tables 1 and 2, while the graphs of the exact solution \( u(x) \) and the Vandermonde numerical solutions \( \tilde{u}_n(x) \) are plotted in figure 1. The CPU time was 7.845, 20.744, and 37.966 seconds respectively.

Table 1. A comparison of the exact solutions \( u(x) \) with the Vandermonde solutions \( \tilde{u}_n(x) \).

| \( x \) | \( u(x) \) | \( u_5(x) \) | \( u_{10}(x) \) | \( u_{15}(x) \) |
|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 |
| 0.1 | 1.10517091807565 | 1.10517091666667 | 1.10517091807565 | 1.10517091807565 |
| 0.2 | 1.22140275816017 | 1.22140266666667 | 1.22140275816017 | 1.22140275816017 |
| 0.3 | 1.34985880757600 | 1.34985775000000 | 1.34985880757596 | 1.34985880757600 |
| 0.4 | 1.49182469764127 | 1.49182469764108 | 1.49182469764018 | 1.49182469764127 |
| 0.5 | 1.64872127070013 | 1.64872127068737 | 1.64872127070001 | 1.64872127070013 |
| 0.6 | 1.82211880039051 | 1.82211880029486 | 1.82211880039051 | 1.82211880039051 |
| 0.7 | 2.01375270740488 | 2.01375270694458 | 2.01375270740488 | 2.01375270740488 |
| 0.8 | 2.22554092849247 | 2.22554092818768 | 2.22554092849247 | 2.22554092849247 |
| 0.9 | 2.45960311156954 | 2.45960310266210 | 2.45960311156954 | 2.45960311156954 |
| 1 | 2.71828182845905 | 2.71828182845899 | 2.71828182845905 | 2.71828182845899 |
Example (2)
Consider Volterra integral equation
\[
\begin{align*}
u(x)=& \cos(x)-e^{x}\sin(x)+\int_{0}^{x}e^{s}u(t)dt. 
\end{align*}
\]
whose exact solution is given by \( u(x)=\cos(x) \) [6]. The obtained interpolant Vandermonde solutions \( \tilde{u}(x_i) \) are found equals the exact one. The Vandermonde numerical solutions \( \tilde{u}_n(x_i) \) and the absolute errors \( E_n(x_i) \) for \( b=1 \) are shown in tables 3 and 4, while the graphs of the exact solution \( u(x_i) \) and the Vandermonde numerical solutions \( \tilde{u}_n(x_i) \) are plotted in figure 2. The CPU time is 12.235, 26.434, and 49.329 seconds respectively.

\begin{table}
\centering
\caption{The absolute errors \( E_n(x_i) \).}
\begin{tabular}{c|c|c|c}
\hline
\( x_i \) & \( E_5(x_i) \) & \( E_{10}(x_i) \) & \( E_{15}(x_i) \) \\
\hline
0 & 0 & 0 & 0 \\
0.1 & 1.40898115397192e-09 & 0 & 0 \\
0.2 & 9.14935032181319e-08 & 4.44089209850063e-16 & 0 \\
0.3 & 1.05757600321610e-06 & 4.55191440096314e-14 & 0 \\
0.4 & 6.03097460372126e-06 & 1.08690834110803e-12 & 0 \\
0.5 & 2.33540334615423e-05 & 1.27624577572760e-11 & 0 \\
0.6 & 7.08003905089960e-05 & 9.56517087757902e-11 & 0 \\
0.7 & 1.81290803809908e-04 & 5.2589571374962e-10 & 0 \\
0.8 & 4.10261825801062e-04 & 2.30478525153899e-09 & 0 \\
0.9 & 8.44861156950039e-04 & 8.49489268458899e-09 & 9.32587340685132e-15 \\
1 & 1.61516179237875e-03 & 2.73126610217389e-08 & 5.10702591327572e-14 \\
\hline
\end{tabular}
\end{table}

\begin{figure}
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{The exact solution and the Vandermonde numerical solutions.}
\end{figure}
Table 3. A comparison of the exact solutions $u(x_i)$ with the Vandermonde solutions $\tilde{u}_n(x_i)$.

| $x_i$ | $u(x_i)$ | $\tilde{u}_5(x_i)$ | $\tilde{u}_{10}(x_i)$ | $\tilde{u}_{15}(x_i)$ |
|------|---------|-------------------|-------------------|-------------------|
| 0.1  | 0.995004165278026 | 0.995019053699074 | 0.995004165278180 | 0.995004165278026 |
| 0.2  | 0.980066577841242 | 0.980069913598738 | 0.980066577841439 | 0.980066577841242 |
| 0.3  | 0.955336489125606 | 0.955342971306947 | 0.955336489124338 | 0.955336489125606 |
| 0.4  | 0.921060994002885 | 0.92110674641305 | 0.921065003690109 | 0.921060994002885 |
| 0.5  | 0.877582561890373 | 0.87768529677859 | 0.877582561484332 | 0.877582561890370 |
| 0.6  | 0.82535614906786 | 0.825876102799883 | 0.82535614906933 | 0.82535614906786 |
| 0.7  | 0.764842187284489 | 0.766198022746671 | 0.764842170979717 | 0.764842187284039 |
| 0.8  | 0.696706709347165 | 0.699744982662323 | 0.696706638882236 | 0.69670670934876 |
| 0.9  | 0.621609968270664 | 0.627814742144536 | 0.621609712274312 | 0.621609968251668 |
| 1    | 0.540302305777236 | 0.552033129293392 | 0.540301494897931 | 0.540302305777236 |

Table 4. The absolute errors $E_n(x_i)$.

| $x_i$ | $E_5(x_i)$ | $E_{10}(x_i)$ | $E_{15}(x_i)$ |
|------|-----------|--------------|--------------|
| 0    | 0         | 0            | 0            |
| 0.1  | 1.48884210480205e-05 | 1.54654067330284e-13 | 0            |
| 0.2  | 3.33575749655868e-06 | 1.5021375231784e-13 | 0            |
| 0.3  | 6.48218134091600e-06 | 1.26765264951700e-12 | 0            |
| 0.4  | 4.69806038420355e-05 | 3.47766260233584e-11 | 0            |
| 0.5  | 1.85967787485919e-04 | 4.0604086759436e-10 | 2.99760216648972e-15 |
| 0.6  | 5.40487890204333e-04 | 3.0057252246990e-09 | 4.49640324973188e-14 |
| 0.7  | 1.35585462182100e-03 | 1.63047718659826e-08 | 4.49085213460876e-13 |
| 0.8  | 3.03827331515749e-03 | 7.0469298910454e-08 | 3.2894797961688e-12 |
| 0.9  | 6.20473878731775e-03 | 2.5596352449550e-07 | 1.899613795414e-11 |
| 1    | 1.17308234525252e-02 | 8.10970208302372e-07 | 9.0903951032832e-11 |

Figure 2. The exact solution and the Vandermonde numerical solutions.

4. Conclusion
A new computational method has been developed to solve Volterra integral equations of the second kind. The method is based on Lagrange interpolation with uniformly spaced interpolation nodes.
through the Vandermonde matrix. The presented method significantly saved time and minimized errors resulting from the complex calculations caused by the increase of the polynomial’s degree.

Moreover, the advantage of this method has contributed to simplifying the solution’s steps by replacing the interpolate unknown function twice into the integral equation, so that the solution is reduced to the solution of one matrix equation without any need to apply the collocation points approach. This means that the exact solutions can be obtained while depending only on the choice of the step size and the smallest value of parameter $b$, regardless of the smoothness of the kernel and the given data function. Thus, Lagrange interpolation via the Vandermonde matrix ensures solutions superior to other methods.

5. References

[1] Kendall E A 2010 *The Numerical Solution of Integral Equations of the Second Kind* (England: Cambridge University Press) pp 455-461.

[2] Shoukralla E S and Markos M A 2018 The economized monic Chebyshev polynomials for solving weakly singular Fredholm integral equations of the first kind *Asian-Eur. J. Math.* 12 (1) 2050030.

[3] Shoukralla E S, Kamel M and Markos M A 2018 A New Computational Method for Solving Weakly Singular Fredholm Integral Equations of the First Kind 13th Int. Conf. on Computer Engineering and Systems (ICCES) IEEE pp 202-207.

[4] Aggarwal S, Sharma N and Chauhan R 2018 Solution of linear Volterra integral equations of second kind using Mohand transform *Int. J. Res. Advent Tech.* 6 (11) 3098-3102.

[5] Izadian J, Salahshour S, and Soheil S 2012 A numerical method for solving Volterra and Fredholm integral equations using homotopy analysis method *AWER Proc. Inf. Technol. Comput. Sci.* 1 406-411.

[6] Wang W 2006 A mechanical algorithm for solving the Volterra integral equation *Appl. Math. Comput.* 172 (2) 1323-1341.

[7] Uma G, Prabahakar V and Hariharan S 2016 Numerical integration using hybrid of block-pulse functions and lagrange polynomial *Int. J. Pure Appl. Math.* 106 (5) 33-44.

[8] Mustafa M M and Ghanim I N 2014 Numerical solution of linear Volterra-Fredholm integral equations using Lagrange polynomials *Math. Theory and Model.* 4 (5) 137-146.

[9] Gander W 2005 Change of basis in polynomial interpolation *Numer. Linear Algebra Appl.* 12 (8) 769-778.