
Mohsen Rabbani1*, S.H.Kiasoltani2

1 Department of Mathematics, Sari Branch, Islamic Azad University, Sari, Iran
2 Department of Mathematics, Qaemshahr Branch, Islamic Azad University, Qaemshahr, Iran

Received: 29 November 2010
Accepted: 31 January 2011

Abstract

In this article we use discrete collocation method for solving Fredholm–Volterra integro–differential equations, because these kinds of integral equations are used in applied sciences and engineering such as models of epidemic diffusion, population dynamics, reaction–diffusion in small cells. Also the above integral equations with convolution kernel will be solved by discrete collocation method. In this method we approximate solution of problem by no smooth piecewise polynomial. Numerical results show a high accuracy and validity discrete collocation method.

Keywords: Discrete Collocation Method, Fredholm–Volterra Integro, Differential Equations.

1 Introduction

For solving Volterra integral equations see[3, 4]. Also for chemical absorption kinetics was used from Volterra integral equations in[5]. Several numerical methods for solving Fredholm integral equations are in[1, 7]. Fredholm integro–differential equation by using Petrove-Galerkin is solved in [8] also by spline wavelet and spline is solved [6, 2] respectively. Fredholm-Volterra integro–differential equation particular with convolution kernel is important and we consider this equation:

\[
\begin{align*}
y'(t) + y(t) &= f(t) + \int_0^t k_1(t, \tau) y(\tau) d\tau + \int_0^t k_2(t, \tau) y(\tau) d\tau, \quad t \in [0, T], \\
y(0) &= \alpha,
\end{align*}
\]

(1)

where \( f(t) \) and \( k_i(t, \tau) \) for \( i = 1, 2 \) are known functions also \( k_i(t, \tau) \) can be convolution kernel, and \( y(t) \) is unknown function.

*Corresponding author
E-mail address: mrabbani@iausari.ac.ir
We decide to approximate $y(t)$ by Lagrange polynomial interpolation such that it obtain by non-smooth piecewise polynomial space. So we use uniform mesh $I_n$ on $[0,T]$,

\[ I_n = \{t_n = nh; \quad n=0,1,...,N, \quad Nh=T \}. \]

Also,

\[ y(\tau) \approx u(\tau) = \begin{cases} u_0(\tau), & t_0 < \tau \leq t_1, \\ u_1(\tau), & t_1 < \tau \leq t_2, \\ \vdots \\ u_{N-1}(\tau), & t_{N-1} < \tau \leq t_N, \end{cases} \]

where $u(\tau)$ is an element of the non-smooth piecewise polynomial with degree less than $m$, in other words:

\[ u(\tau) \in S_m^{-1}(I_n) = \{u(\tau) \mid u(\tau) = u_n(t) \in \pi_{m-1}, \quad \tau \in (t_n, t_{n+1}], \quad n = 0,...,N - 1\}, \]

\[ t_0 = 0, \quad t_1 = \frac{T}{N},...,T_N. \quad (for \ simplification, T = 1). \]

For finding $u_n(t)$, $n = 0,1,...,(N-1)$ in (3), we use Lagrange interpolation with $(c_i,y_{n,i})$ points for $n = 0,1,...,(N-1)$ and $i = 0,1,...,m$, where $y_{n,i}$ is approximation $y(t_{n,i})$ and $c_i$, $i = 1,...,m$ are $m$-points Gauss as collocation parameters that they can be found by roots of Legendre polynomial on $[0,1]$, for example in the case of $m = 4$ they are obtained the following form:

\[ p(s) = \sqrt{7}(20s^3 - 30s^2 + 12s - 1) = 0, \]

\[ c_1 = \frac{5 - \sqrt{15}}{10}, \quad c_2 = \frac{1}{2}, \quad c_3 = \frac{5 + \sqrt{15}}{10}, \]

also we choose $c_4 = 1$ as end-point of $[0,1]$.

By considering $m$-points Gauss, we define mesh points as follows:

\[ t_{n,i} = t_n + c_i h, \quad i = 1,2,...,m, \quad n = 0,1,...,(N-1). \]

According to condition of interpolation we can write

\[ u(t_{n,i}) = y_{n,i}, \quad i = 1,...,m, \quad n = 0,1,...,(N-1), \]

and,

\[ u_n(t_n + \theta h) = \sum_{i=1}^{m} L_i(\theta) y_{n,i}, \quad \theta \in [0,1], \quad n = 0,1,...,(N-1), \]

by choosing $\tau = t_n + \theta h$ then $\tau \in (t_n, t_{n+1})$, in this way $u(t)$ is restricted to subintervals such as $(t_n, t_{n+1})$.

For solving Eq.(1), it can be written the following form:
y'(t) + y(t) = f(t) + \int_0^t k_1(t, \tau) y(\tau) d\tau + \int_t^1 k_2(t, \tau) y(\tau) d\tau.

Now, we introduce discrete form Eq. (1),

\begin{align*}
F_n(t) &= f(t) + \int_0^t k_1(t, \tau) y(\tau) d\tau + \int_t^1 k_2(t, \tau) y(\tau) d\tau, \\
y'(t) + y(t) &= F_n(t) + \int_0^t k_1(t, \tau) y(\tau) d\tau + \int_t^1 k_2(t, \tau) y(\tau) d\tau.
\end{align*}

For numerical solution of Eqs. (8–9), we use (3–7) as follows:

\begin{align*}
F_n(t_{n,i}) &= f(t_{n,i}) + \int_0^{t_{n,i}} k_1(t_{n,i}, \tau) u(\tau) d\tau + \int_0^{t_{n,i}} k_2(t_{n,i}, \tau) u(\tau) d\tau, \\
\sum_{i=1}^m L_n'(t_{n,i}) y_{n,i} + y_{n,i} &= F_n(t_{n,i}) + \int_{t_{n,i}}^{t_{n,i+1}} k_1(t_{n,i}, \tau) u(\tau) d\tau + \int_{t_{n,i}}^{t_{n,i+1}} k_2(t_{n,i}, \tau) u(\tau) d\tau,
\end{align*}

where \(u(\tau)\) is introduced by (3). At first \(F_n(t_{n,i})\), \(n=0,1,...,(N-1)\), \(i=1,2,...,m\) are computed by (10), then they are used in system (11) for obtaining \(y_{n,i}\) s.

In (10) for \(n=j\) we have:

\begin{align*}
F_j(t_{j,i}) &= f(t_{j,i}) + \int_0^{t_{j,i}} [k_1(t_{j,i}, \tau) + k_2(t_{j,i}, \tau)] u(\tau) d\tau, \\
\sum_{i=1}^m L_j'(t_{j,i}) y_{j,i} + y_{j,i} &= F_j(t_{j,i}) + \int_{t_{j,i}}^{t_{j,i+1}} [k_1(t_{j,i}, \tau) + k_2(t_{j,i}, \tau)] u(\tau) d\tau,
\end{align*}

From (3) and (7), we can write

\begin{align*}
F_0(t_{0,i}) &= f(t_{0,i}), \quad i=1,2,...,m, \\
F_j(t_{j,i}) &= f(t_{j,i}) + \sum_{p=1}^{N-1} \left[ k_1(t_{j,i}, \tau) + k_2(t_{j,i}, \tau) \right] u_p(\tau) d\tau, \\
j=1,...,(N-1),
\end{align*}

By substituting \(F_n(t_{n,i})\), \(n=0,1,...,(N-1)\) and \(i=1,2,...,m\) in (11) \(y_{n,i}\) s are obtained by the following system:

\begin{align*}
\sum_{i=1}^m L_n'(t_{n,i}) y_{n,i} + y_{n,i} &= F_n(t_{n,i}) + \int_{t_{n,i}}^{t_{n,i+1}} k_1(t_{n,i}, \tau) u_n(\tau) d\tau + \sum_{p=n}^{(N-1)} \int_{t_p}^{t_{n,i}} k_2(t_{n,i}, \tau) u_p(\tau) d\tau, \\
n=0,1,...,(N-1),
\end{align*}

where \(L_\theta(\theta) = \prod_{i=1}^m \frac{(\theta-c_i)}{(c_q-c_i)}\), \(\tau = t_p + \theta h\), so \(\theta = \frac{\tau-t_p}{h}\) and
\[ u_p(\tau) = \sum_{q=1}^{m} L_q(\theta) y_{n,q} = \sum_{q=1}^{m} L_q \left( \frac{\tau - t}{h} \right) y_{n,q}, \quad p = 0, \ldots, (N-1). \] (14)

By solving algebraic system (13) we find \( y_{n,i}, S, i = 1,2,\ldots,m, n = 0,1,\ldots, (N-1) \), then by (14), \( u_p(\tau), p = 1,\ldots, (N-1) \) are found and also \( u(t) \) as a approximation of \( y(t) \) is introduced.

3 Application

In this section, we solve two examples of Fredholm–Volterra integro–differential equations by discrete collocation method.

Example 1

Consider Fredholm–Volterra integro–differential equations with convolution kernel:

\[ \begin{cases} y'(t) + y(t) = 36e^{-t} - 15e^{-\frac{t}{3}} - \frac{t^4}{6} + t^2 + 4t + 7 + \int_{0}^{t} (t-\tau)^2 y(\tau)d\tau + \int_{0}^{1} e^{t-\tau} y(\tau)d\tau, \\ y(0) = 5, \end{cases} \]

with exact solution \( y(t) = 5 + 2t + t^2 \).

In the case of \( m = 4, h = \frac{1}{2}, T = 1 \), we use discrete collocation method and (12-14) then interpolation is obtained the following form and numerical results are shown in table 1,

\[ u(t) = \begin{cases} 5 + 2t + t^2 - 1.38584 \times 10^{-10} t^3, & 0 < t \leq \frac{1}{2} \\ 5 + 2t + t^2 + 3.91083 \times 10^{-10} t^3, & \frac{1}{2} < t \leq 1, \end{cases} \]

<table>
<thead>
<tr>
<th>( t_i )</th>
<th>( | u(t) - y(t) |_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>4.1 \times 10^{-9}</td>
</tr>
<tr>
<td>0.1</td>
<td>3.7 \times 10^{-9}</td>
</tr>
<tr>
<td>0.2</td>
<td>3.4 \times 10^{-9}</td>
</tr>
<tr>
<td>0.3</td>
<td>3.2 \times 10^{-9}</td>
</tr>
<tr>
<td>0.4</td>
<td>2.9 \times 10^{-9}</td>
</tr>
<tr>
<td>0.5</td>
<td>2.7 \times 10^{-9}</td>
</tr>
<tr>
<td>0.6</td>
<td>1.4 \times 10^{-9}</td>
</tr>
<tr>
<td>0.7</td>
<td>1.4 \times 10^{-9}</td>
</tr>
<tr>
<td>0.8</td>
<td>1.4 \times 10^{-9}</td>
</tr>
<tr>
<td>0.9</td>
<td>1.4 \times 10^{-9}</td>
</tr>
<tr>
<td>1</td>
<td>1.5 \times 10^{-9}</td>
</tr>
</tbody>
</table>
According numerical results, it is obvious that the proposed method has a high accuracy.

Example 2
In this example we solve Fredholm-Volterra integro-differential equations as follows:
\[
\begin{align*}
  y'(t) + y(t) &= 6 - 6e^t + 8t + 4t^2 + t^3 + \frac{\cos t}{t^2} + \int_0^t e^{-\tau} y(\tau)d\tau + \int_0^t \frac{\sin(\tau)}{\tau} y(\tau)d\tau, \\
y(0) &= 0,
\end{align*}
\]
with exact solution \( y(t) = t^2 \).

By choosing \( m = 4, \ h = \frac{1}{2}, \ T = 1 \) in the discrete collocation method and use (12 –14) interpolation \( u(t) \) is introduced by,
\[
\begin{align*}
  u(t) &= \begin{cases} 
    2.44001 \times 10^{-7} - 2.43645 \times 10^{-7} t + 9.22112 \times 10^{-9} t^2, & 0 < t \leq \frac{1}{2} \\
    -2.717 \times 10^{-7} + 3.10075 \times 10^{-7} t + 2.08193 \times 10^{-8} t^3, & \frac{1}{2} < t \leq 1
  \end{cases}
\end{align*}
\]
and absolute errors in some points are given in table 2.

<table>
<thead>
<tr>
<th>( t )</th>
<th>( |u(t) - y(t)|_1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>( 2.4 \times 10^{-7} )</td>
</tr>
<tr>
<td>0.1</td>
<td>( 2.2 \times 10^{-7} )</td>
</tr>
<tr>
<td>0.2</td>
<td>( 2.0 \times 10^{-7} )</td>
</tr>
<tr>
<td>0.3</td>
<td>( 1.8 \times 10^{-7} )</td>
</tr>
<tr>
<td>0.4</td>
<td>( 1.6 \times 10^{-7} )</td>
</tr>
<tr>
<td>0.5</td>
<td>( 1.5 \times 10^{-7} )</td>
</tr>
<tr>
<td>0.6</td>
<td>( 1.2 \times 10^{-7} )</td>
</tr>
<tr>
<td>0.7</td>
<td>( 1.1 \times 10^{-7} )</td>
</tr>
<tr>
<td>0.8</td>
<td>( 9.5 \times 10^{-8} )</td>
</tr>
<tr>
<td>0.9</td>
<td>( 8.2 \times 10^{-8} )</td>
</tr>
<tr>
<td>1</td>
<td>( 7.0 \times 10^{-8} )</td>
</tr>
</tbody>
</table>

By considering numerical results are shown in the table 2, we conclude the discrete collocation method has a high accuracy and it can be used for other problems.

4 Conclusion
In this work, we discrete integral equations to use collocation method, thus we could obtain the numerical results with high accuracy. Also these results are shown ability and validity the proposed method.
References