A Galerkin-like approach to solve high-order integro-differential equations with weakly singular kernel

Şuayip Yüzbaşi*, Murat Karaçayır

Department of Mathematics, Faculty of Science, Akdeniz University, TR 07058 Antalya, Turkey

*Corresponding Author: Email: syuzbasi@akdeniz.edu.tr

Abstract

In this study, a Galerkin-like approach is applied to numerically solve high-order integro-differential equations having weakly singular kernel. The method includes taking inner product of a set of monomials with a vector obtained from the equation in question. The resulting linear system is then solved, yielding a polynomial as the approximate solution. Additionally, the technique of residual correction, which aims to increase the accuracy of the approximate solution, is discussed briefly. Lastly, the method and the residual correction technique are illustrated with several examples. The results are also compared with numerous existing methods from the literature.

Keywords: Galerkin method; inner product; integro-differential equations; residual error correction; weakly singular kernel.

1. Introduction

Integral and integro-differential equations arise rather naturally and frequently in science and engineering problems. Furthermore, many problems in which boundary and initial value problems are relevant can be stated in an equivalent setting that includes integral and integro-differential equations. This fact has made the emergence of a huge literature on the topic inevitable. Subjects which make heavy use of integral and integro-differential equations include electromagnetics (Volakis & Sertel, 2012), potential theory (Jaswon, 1963), optics (Pritchett & Trubatch, 2004), population dynamics (Lotka, 1939), heat transfer (Kulish & Novozhilov, 2004) and astrophysics (Huang, 1955). More on the applications and sources of integral equations can be found in Lonseth (1977). Readers interested in texts of historical import can refer to Bôcher (1914). Modern treatises on the field of integral equations include Wazwaz (2011) and Kress (1999). For a monograph on the history of Fredholm equations see Atkinson (2008).

Depending on the limits of the involved integrals, Fredholm, Volterra and Fredholm-Volterra are the basic types of integral and integro-differential equations
encountered in the field. Their complicated nature implies that analytic solutions for integral and integro-differential equations are not available in all but exceptional cases. In this respect, the type of the equation as well as the characteristics of the application makes the usage of certain numerical methods more appropriate for the problem under discussion. Among such numerical methods are the Taylor expansion method (Huang & Li, 2010), operational matrix method (Singh & Postnikov, 2013), Legendre pseudospectral method (El-Kady & Biomy, 2010), differential transform method (Tari et al., 2009), homotopy perturbation method (Ghasemi et al., 2007; Roul & Meyer, 2011; Yusufoglu, 2009), Bessel matrix method (Yüzbaşı et al., 2011) and Legendre polynomial method (Yalçınbaş et al., 2009).

Integral and integro-differential equations having singularities in one of their kernel functions are of special import, since they are encountered in a wide variety of applications. In this case, the complex behavior of the singular kernel makes it difficult to obtain a high-order solution to the problem. Therefore, the volume of work done in the area of singular equations is relatively small. Nevertheless, some authors have contributed to the topic by presenting several numerical schemes, especially in recent years. These schemes include collocation methods based on Bessel polynomials (Yüzbaşı & Sezer, 2012) and Bernstein polynomials (İşik et al., 2011), Discrete Galerkin method (Pedas & Tamme, 2008), a method based on Legendre multiwavelets (Lakestani et al., 2011), reproducing kernel method (Du et al., 2014) and the method proposed in Bougoffa et al. (2011).

In this study, our aim is to find an approximate solution to the linear Fredholm-Volterra integro-differential equation with weakly singular kernel and having the form

$$\sum_{i=0}^{j} P_i(x) y^{(i)}(x) = g(x) + \lambda_1 \int_0^x \frac{y^{(k)}(t)}{\sqrt{x-t}} \, dt + \lambda_2 \int_0^x K_1(x,t) y^{(l)}(t) \, dt$$

$$+ \lambda_3 \int_a^b K_2(x,t) y^{(m)}(t) \, dt, \quad 0 \leq x, t \leq b,$$

under the mixed conditions

$$\sum_{j=0}^{n-1} [a_{ij} y^{(j)}(a) + b_{ij} y^{(j)}(b)] = \gamma_i, \quad n = \max(J,k,l,m), \quad i = 0,1,\ldots,n-1. \quad (2)$$

Here superscripts in parantheses denote the order of differentiation, $y^{(0)}(x) = y(x)$ is the unknown function to be found, $a$ and $b$ are real numbers, $g(x), P_i(x), K_1(x)$ and $K_2(x)$ are known continuous functions defined on the interval $[0,b]$, $\lambda_1, \lambda_2, \lambda_3, a_{ij}, b_{ij}, \gamma_i$ are real or complex constants.
The organization of the paper is as follows: The solution method is explained in Section 2. The subject of Section 3 is the technique of residual correction, which aims to derive better approximations from a known one. Section 4 contains numerical examples, where comparisons with other methods are also made. Finally, the conclusion of the paper is given in Section 5.

2. Method of solution

In this section, the procedure we will use to solve Equation 1 is outlined. The same method was employed in Türkyılmazoğlu (2014) to obtain approximate solutions of high-order Fredholm integro-differential equations with possibly singular kernel functions.

Firstly, we make the assumption that the unique solution $y(x)$ of Equation 1 can be expressed as a power series of the form

$$y(x) = \sum_{k=0}^{\infty} a_k x^k.$$  

We then truncate this power series after the $(N+1)^{st}$ term so that

$$y_N(x) = \sum_{k=0}^{N} a_k x^k = X \cdot A$$  

where

$$X = [1 \ x \ x^2 \ \ldots \ x^N], \ A = [a_0 \ a_1 \ a_2 \ \ldots \ a_N]^T.$$  

Here, the coefficients $a_i$ are to be determined as the output of the method. The derivatives of $y_N(x)$ can be expressed as a product of matrices with the help of a special matrix. Namely, if we define $B$ to be the $(N+1) \times (N+1)$ matrix such that $B_{i,i+1} = i$ for $i = 1, 2, \ldots, N$ and $B_{i,j} = 0$ elsewhere, then the following equality holds:

$$y_N^{(i)}(x) = X B^i A = [1 \ x \ x^2 \ \ldots \ x^N] \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & N \\ 0 & 0 & 0 & \ldots & 0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}.$$  

Next, substituting the matricial expressions for $y_N(x)$ and $y_N^{(i)}(x)$ into Equation 1 gives us the equation

$$G(x)A = g(x)$$  

where

$$G(x) = \sum_{i=0}^{J} P_i(x) X^i - \lambda_1 \int_0^x \frac{TB^k}{\sqrt{x-t}} dt - \lambda_2 \int_0^x K_1(x,t) TB^l dt - \lambda_3 \int_0^b K_2(x,t) TB^m dt$$

and

$$T = [1 \quad t \quad t^2 \quad \ldots \quad t^N]$$.

In order to simplify the computation of \(G(x)\), we use the following equality for the first integral in its expression (Yusufoglu, 2009):

$$\int_0^x \frac{t^n}{\sqrt{x-t}} = \sqrt{\pi} x^{\frac{1+n}{2}} \frac{\Gamma(n+1)}{\Gamma(n + \frac{3}{2})}$$.

In this case, the first integral in the expression of \(G(x)\) becomes

$$\int_0^x \frac{TB^k}{\sqrt{x-t}} dt = \sqrt{\pi} \begin{bmatrix} x^2 \Gamma(1) & x^2 \Gamma(2) & \ldots & x^{2+N} \Gamma(1+N) \\ \Gamma\left(\frac{3}{2}\right) & \Gamma\left(\frac{5}{2}\right) & \ldots & \Gamma\left(\frac{3}{2} + N\right) \end{bmatrix} B^k$$.

Now, in order to convert Equation 5 into a system of linear equations in the unknowns \(a_i\), we take inner product of Equation 5 with the elements of the set \(\Phi = \{1, x, x^2, \ldots, x^N\}\), where the inner product of two functions \(f, g\) from the Hilbert space \(L^2[a,b]\) is defined by

$$< f, g > = \int_a^b f(x) g(x) dx$$.

Each inner product will result in a linear equation in the unknown coefficients \(a_i\). Finally, we are left with a linear system \(WA = G\) where the \((N+1) \times (N+1)\) matrix \(W\) and the column matrix \(G\) of length \(N+1\) are given by

$$W_{i,j} = < x^{i-1}, G(x)_{i,j} >, G_{i,1} = < x^{i-1}, g(x) >$$.

More explicitly, \(W\) and \(G\) are given by
Since the solution function should satisfy the initial conditions, we must feed them into the above system. For this purpose, we first express the initial conditions in matrix form. As a preparation, we define \( \mathbf{X}(a) \) as the row vector of length \( N+1 \) formed by substituting \( x \) by \( a \) in \( \mathbf{X} \), or more explicitly as

\[
\mathbf{X}(a) = [1 \ a \ a^2 \ \ldots \ a^N].
\]

Then, the initial conditions (2) can be written as

\[
\left\{ \sum_{j=0}^{n-1} a_j \mathbf{X}(0) \mathbf{B}^j + \sum_{j=0}^{n-1} b_j \mathbf{X}(b) \mathbf{B}^j \right\} \mathbf{A} = \gamma_i, i = 0,1,\ldots,n-1. \tag{7}
\]

Although this is enough, there is a more compact way of forming the row vectors that will represent the initial conditions. Simply define the two polynomials

\[
Q_i(x) = \sum_{j=0}^{n-1} a_j x^j \quad \text{and} \quad R_i(x) = \sum_{j=0}^{n-1} b_j x^j. \]

Then (7) can be arranged as

\[
\mathbf{C}_i \cdot \mathbf{A} = \left\{ \mathbf{X}(0)Q_i(\mathbf{B}) + \mathbf{X}(b)R_i(\mathbf{B}) \right\} \mathbf{A} = \gamma_i, i = 0,1,\ldots,n-1. \tag{8}
\]

In order to include the initial conditions in the discussion, we just replace the \( i \)-th row of \( \mathbf{W} \) by \( \mathbf{C}_{i-1} \) and the \( i \)-th entry of \( \mathbf{G} \) by \( \gamma_{i-1} \). This will yield the modified system \( \tilde{\mathbf{W}} \mathbf{A} = \tilde{\mathbf{G}} \) explicitly given by

\[
\tilde{\mathbf{W}} = \begin{bmatrix}
\mathbf{C}_0 \\
\vdots \\
\mathbf{C}_{n-1}
\end{bmatrix}
\begin{bmatrix}
< x^n, \mathbf{G}(x)_{1,1} > & \ldots & < x^n, \mathbf{G}(x)_{1,N+1} > \\
< x^{n+1}, \mathbf{G}(x)_{1,1} > & \ldots & < x^{n+1}, \mathbf{G}(x)_{1,N+1} > \\
\vdots & \vdots & \vdots \\
< x^N, \mathbf{G}(x)_{1,1} > & \ldots & < x^N, \mathbf{G}(x)_{1,N+1} >
\end{bmatrix}, \tag{9}
\]

\[
\tilde{\mathbf{G}} = \begin{bmatrix}
\gamma_0 & \ldots & \gamma_{n-1} & < x^n, g(x) > & < x^{n+1}, g(x) > & \ldots & < x^N, g(x) >
\end{bmatrix}^T.
\]
Finally, we compute the matrix of unknown coefficients as $\mathbf{A} = \mathbf{W}^{-1}\mathbf{G}$ and the approximate solution is given by

$$y_N(x) = a_0 + a_1x + \ldots + a Nx^N.$$  

3. Error estimation and residual correction

In this section, the error estimation of our method is performed based on the residual function of the integro-differential equation given by Equation 1. Then, starting with an approximate solution of Equation 1, the way of obtaining a better approximation from the original one is described.

Let us consider the residual function

$$R(x) = \sum_{i=0}^{J} P_i(x) y^{(i)}(x) - g(x) - \lambda_1 \int_{0}^{x} \frac{y^{(k)}(t)}{\sqrt{x-t}} dt - \lambda_2 \int_{0}^{x} K_1(x,t) y^{(i)}(t) dt$$

$$- \lambda_3 \int_{0}^{b} K_2(x,t) y^{(m)}(t) dt = 0 \tag{10}$$

of Equation 1. Substituting the approximate solution $y_N(x)$ in place of $y(x)$ we get

$$R_N(x) = \sum_{i=0}^{J} P_i(x) y_N^{(i)}(x) - g(x) - \lambda_1 \int_{0}^{x} \frac{y_N^{(k)}(t)}{\sqrt{x-t}} dt - \lambda_2 \int_{0}^{x} K_1(x,t) y_N^{(i)}(t) dt$$

$$- \lambda_3 \int_{0}^{b} K_2(x,t) y_N^{(m)}(t) dt. \tag{11}$$

Subtracting Equation 10 from Equation 11 and rearranging yields

$$R_N(x) = - \lambda_1 \int_{0}^{x} \frac{e_N^{(k)}(t)}{\sqrt{x-t}} dt - \lambda_2 \int_{0}^{x} K_1(x,t) e_N^{(i)}(t) dt - \lambda_3 \int_{0}^{b} K_2(x,t) e_N^{(m)}(t) dt$$

$$+ \sum_{i=0}^{J} P_i(x) e_N^{(i)}(x), \tag{12}$$

which is the same as Equation 1 with $g(x)$ replaced by $R_N(x)$ and $e_N(x) = y_N(x) - y(x)$ is the error function. In addition, since $y(x)$ and $y_N(x)$ both satisfy the initial conditions (2),

$$\sum_{j=0}^{n-1} \left[ a_j e_N^{(j)}(0) + b_j e_N^{(j)}(b) \right] = 0, \quad n = \max(J,k,l,m), \quad i = 0,1,\ldots,n-1 \tag{13}$$

are the initial conditions for Equation 12. Next, the method of Section 2 is applied to solve it for $e_N(x)$ with some choice of the parameter value $M$ and the approximate solution $e_{N,M}(x)$ is obtained as a result. Consequently, this new approximation is used to obtain a corrected solution
of Equation 1. In what follows, \( E_{N,M}(x) \) will denote the actual error of \( y_{N,M}(x) \) given by \( E_{N,M}(x) = y_{N,M}(x) - y(x) \).

### 4. Numerical examples

In this section, the method of Section 2 is applied to several examples.

**Example 1:** Our first example is the following linear Volterra integro-differential equation with weakly singular kernel taken from (Yüzbaşı & Sezer, 2012):

\[
y''(x) - x^2 y'(x) = 6x - \frac{1}{2} x^4 - \frac{16}{5} x^{5/2} + \int_0^x \frac{y'(t)}{\sqrt{x-t}} dt - \frac{5}{4} \int_0^x x y''(t) dt, 0 \leq x, t \leq 1
\]

The exact solution is \( y(x) = 1 + x^3 \). The parameters for this example are given by \( J = 2, k = 1, l = 2, m = 2, \lambda_1 = 1, \lambda_2 = -\frac{5}{4}, b = 1, P_0(x) = 0, P_1(x) = -x^2, P_2(x) = 1, K(x,t) = xt, g(x) = 6x - \frac{1}{2} x^4 - \frac{16}{5} x^{5/2} \). Applying the method outlined in Section 2 for any \( N \geq 3 \) gives the exact solution \( y_N(x) = 1 + x^3 \). Thus, this is an example where the method yields the exact solution in case the exact solution is a polynomial.

**Example 2:** Next, we consider the following Bagley-Torvik equation examined also in Işık et al. (2011); Yüzbaşı & Sezer (2012) and Huang & Li (2010):

\[
y''(x) + y(x) = e^x (2 + \text{erf}(\sqrt{x})) - \frac{1}{\sqrt{\pi}} \int_0^x \frac{y''(t)}{\sqrt{x-t}} dt, 0 \leq x, t \leq 1
\]

with the initial conditions \( y(0) = y'(0) = 1 \), where the Gaussian error function is given by

\[
\text{erf}(x) = \frac{2}{\sqrt{x}} \int_0^x e^{-t^2} dt.
\]
When implemented in MATLAB on a system with dual core 2.20 GHZ processor. It is seen that the present method is slightly outperformed by the Bernstein series solution of (Işık et al., 2011) for $N = 4$. However, we obtain solidly decreasing absolute error values when we increase $N$. Another advantage of the present method is that the absolute error values is more evenly distributed between the interval $[0,1]$, compared to the other methods. As for the implementation time, the values in the last row are completely acceptable. The improvement in accuracy provided by residual correction is exhibited in the following examples.

Example 3: Our third example is the linear weakly singular Volterra integral equation considered in (Saberi & Heidari, 2014):

$$y(x) = \sqrt{x} + \frac{1}{2} \pi x - \int_0^x \frac{y(t)}{\sqrt{x-t}} dt, 0 \leq x \leq 1.$$  \hspace{1cm} (18)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>0.1452E-4</td>
<td>0.1372E-3</td>
<td>0.5205E-3</td>
<td>0.4194E-4</td>
<td>0.2433E-5</td>
<td>0.1141E-6</td>
</tr>
<tr>
<td>0.4</td>
<td>0.3665E-4</td>
<td>0.0015</td>
<td>0.0013</td>
<td>0.8662E-4</td>
<td>0.4469E-5</td>
<td>0.3071E-6</td>
</tr>
<tr>
<td>0.6</td>
<td>0.3186E-4</td>
<td>0.0063</td>
<td>0.0018</td>
<td>0.1164E-3</td>
<td>0.6189E-5</td>
<td>0.8882E-6</td>
</tr>
<tr>
<td>0.8</td>
<td>0.3220E-3</td>
<td>0.0172</td>
<td>0.0022</td>
<td>0.1452E-3</td>
<td>0.7623E-5</td>
<td>0.2339E-5</td>
</tr>
<tr>
<td>1</td>
<td>0.0021</td>
<td>0.0369</td>
<td>0.0025</td>
<td>0.1663E-3</td>
<td>0.8682E-5</td>
<td>0.5349E-5</td>
</tr>
</tbody>
</table>

CPU Time (in seconds) | 0.5782 | 0.6900 | 0.9581 | 1.0163
Fig. 1. Graphics of the absolute errors of Bernstein collocation method, Taylor expansion method and the present method for different values of $N$ in Example 2.

Table 2. Absolute errors at some points and implementation time related to some parameters for Example 3.

<table>
<thead>
<tr>
<th>$x$</th>
<th>Trapezoid modified quadrature method (Saberi &amp; Heidari, 2014)</th>
<th>Present method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N = 5$</td>
<td>$N = 6$</td>
</tr>
<tr>
<td></td>
<td>$N = 7$</td>
<td>$N = 5, M=6$</td>
</tr>
<tr>
<td></td>
<td>$N = 5, M=8$</td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>0.345E-3</td>
<td>0.0125</td>
</tr>
<tr>
<td>0.4</td>
<td>0.0021</td>
<td>0.0045</td>
</tr>
<tr>
<td>0.6</td>
<td>0.003</td>
<td>0.0015</td>
</tr>
<tr>
<td>0.8</td>
<td>0.0036</td>
<td>0.0037</td>
</tr>
<tr>
<td>1</td>
<td>0.0039</td>
<td>0.2486E-13</td>
</tr>
<tr>
<td></td>
<td>CPU Time (in seconds)</td>
<td>0.5699</td>
</tr>
</tbody>
</table>

We have $\lambda_1 = \lambda_2 = 0$ as in the previous example and the exact solution is $y(x) = \sqrt{x}$. Using the method discussed in Section 2, we numerically solved this problem for $N = 5, 6$ and 7. Table 2 compares the results of the trapezoid modified quadrature method employed in Saberi & Heidari (2014) with the results of the present method for
these $N$ values. It can be inferred that the present method gives better results especially for increasing values of $x$. It is also seen that increasing the parameter $N$ gives rise to slightly more accurate results in general. In order to illustrate the method of residual correction, we attempt to make the approximate solution for $N=5$ more accurate and it is seen that the absolute error values are in general reduced as a result. The estimated and real absolute error values for $M=6$ and 8 are shown in Figure 2 and in the last two columns of Table 2. One can also observe that $|E_{N,M}(x)|$ is smaller when $|e_N(x)|$ and $|e_{N,M}(x)|$ are close and it is greater when they are not. This is true since we have $E_{N,M}(x) = e_N(x) - e_{N,M}(x)$.

Example 4: Our last example is the following nonsingular Fredholm integro-differential equation taken from Huang & Li (2010):

$$y'' + xy' - xy = e^x - 2\sin x + \int_{-1}^{1} \sin xe^{-t} y(t) dt, \quad -1 \leq x, t \leq 1$$  \hspace{1cm} (19)$$

subject to the conditions $y(0) = y'(0) = 1$. The exact solution is $y(x) = e^x$. Table 3 contains the results in terms of the absolute error function of the Taylor expansion method (Huang & Li 2010) for two different $n$ values and the present method for four different $N$ values. Looking at the table, the tendency of the absolute error function to grow as $x$ moves away from zero is seen at first. This is because the initial conditions are given at $x=0$. It can also be commented that applying the two methods with the same parameter values gives rise to close results. As for the present method, improving the solutions by increasing $N$ is possible, which is apparent from the table; but greater $N$ does not necessarily mean improved accuracy. An example of this is given by the fact that the results of $N=7$ are not better than those of $N=6$ although $7 > 6$. If we proceed with $N=8$, the results get worse rather quickly.

We applied the residual error correction technique to this problem for $(N,M) = (3,4), (3,5)$ and $(N,M) = (4,5), (4,6)$. Figure 3 depicts the absolute error values before and after the residual correction. It is seen that the technique provides a significant improvement for this problem. Interestingly, $|E_{3,4}(x)|$ turns out to be almost equal to $|e_4(x)|$. The same holds also for $|E_{3,5}(x)|$ and $|E_{4,5}(x)|$. The second part of Table 3 compares the error values related to the method of (Huang & Li 2010) with the error values of the corrected solutions. While the error values of the present method before residual correction was slightly worse, it is apparent from the table that this situation is no longer valid after residual correction. Especially the error values related to $(N,M) = (4,6)$ have become considerably better than the method of Huang & Li (2010) with $N=4$. 


Fig. 2. Graphics of the estimated and real absolute error functions after applying residual correction for $(N,M) = (5,6)$ and $(5,8)$ in Example 3.
### Table 3. Absolute errors at some points for Example 4.

<table>
<thead>
<tr>
<th>x</th>
<th>Taylor expansion method</th>
<th>Present method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>N = 3</td>
<td>N = 4</td>
</tr>
<tr>
<td>-1</td>
<td>0.0044</td>
<td>0.0011</td>
</tr>
<tr>
<td>-0.8</td>
<td>0.0019</td>
<td>0.4215E-3</td>
</tr>
<tr>
<td>-0.6</td>
<td>0.6283E-3</td>
<td>0.1229E-3</td>
</tr>
<tr>
<td>-0.4</td>
<td>0.1450E-3</td>
<td>0.2098E-4</td>
</tr>
<tr>
<td>-0.2</td>
<td>0.1746E-4</td>
<td>0.4569E-6</td>
</tr>
<tr>
<td>0</td>
<td>0.1107E-15</td>
<td>0.1107E-15</td>
</tr>
<tr>
<td>0.2</td>
<td>0.4986E-4</td>
<td>0.8151E-5</td>
</tr>
<tr>
<td>0.4</td>
<td>0.7995E-3</td>
<td>0.1418E-3</td>
</tr>
<tr>
<td>0.6</td>
<td>0.0052</td>
<td>0.9190E-3</td>
</tr>
<tr>
<td>0.8</td>
<td>0.0232</td>
<td>0.0039</td>
</tr>
<tr>
<td>1</td>
<td>0.0825</td>
<td>0.013</td>
</tr>
</tbody>
</table>

### 5. Conclusion

In this paper, we have presented a Galerkin-like approach for the approximate solution of weakly singular integro-differential equations. The method comprises taking inner product of a basis set of monomials with an expression obtained from the equation in question. It turns out that this approach yields the exact solution in case the solution is a polynomial. It is also revealed that our approach gives better or close results
compared to several popular methods present in the literature. Residual error correction technique to improve the accuracy of approximate solutions has also been discussed. Simulation results show that great improvements in the approximate solutions can be achieved as a result of employing this technique.

6. Acknowledgments

The authors are supported by the Scientific Research Project Administration of Akdeniz University. We also thank two anonymous referees for their valuable suggestions which helped to improve this manuscript greatly.

![Fig. 3. Graphics of the absolute, estimated and corrected errors after residual correction for several values of $N$ and $M$ in Example 4.](image)

References


A Galerkin-like approach to solve high-order integro-differential equations with weakly singular kernel


Submitted : 06/11/2014

Revised : 02/06/2015

Accepted : 27/07/2015
مقارنة شبيهة غالركن لحل معادلات تكامل – تفاضلية ذات نواع ضعيفة الانفرد

سياق قراري، مراد كراكر
قسم الرياضيات، كلية العلوم، جامعة أكدينز، تركيا 07058 أنطاليا، تركيا
المؤلف: البريد الإلكتروني: syuzbasi@akdeniz.edu.tr

خلاصة

نقوم في هذا البحث باستخدام مقارنة شبيهة لمقارنة غالركن لحل معادلات تكامل – تفاضلية من مرتبتها علية لها نواع ضعيفة الانفرد. وتشتمل طويلة,long, على استخدام الجداء الداخلي لمجموعة من وحدات الجذور ومتجمهرة نحصل عليه من المعادلة من تحت الدراسة. ثم نقوم بحل النظام الخطي الذي نحصل عليه فينتج عن ذلك حدودية كحل تقريبي. إضافة إلى ذلك، فأنا ندرس تقنية التصحيح الراسمي بإيجاز حيث يقوم هذا التصحيح بزيادة دقة الحل التقريبي. أخيراً تقوم بتوضيح تقنية التصحيح الراسمي من خلال أمثلة عديدة. كما نقوم بمقارنة نتائجنا بالعديد من الطرق الموجودة و المشورة.