Numerical Solutions of First Kind Linear Fredholm Integral Equations by Using Quarter-Sweep SOR With Piecewise Linear Collocation Method
QSSOR With Piecewise Linear Collocation Method for Fredholm Equations

Mohana Sundaram Muthuvalu, Jumat Sulaiman

School of Science and Technology Universiti Malaysia Sabah Kota Kinabalu, Malaysia

Abstract: Presently, there are many sciences and numerous applications especially in engineering can be transformed into integral equations problems. In this paper, the effectiveness of the Quarter-Sweep Successive Over-Relaxation (QSSOR) method by using the quarter-sweep linear collocation approximation equation to solve first kind linear integral equations of Fredholm type is investigated. Furthermore, the formulation and implementation of the proposed method are also presented. Through the numerical experiments, two examples are provided to illustrate the performance of the proposed method.

Key words: linear Fredholm equations; quarter-sweep iteration; collocation; Successive Over-Relaxation

INTRODUCTION

Integral equations are widely used as mathematical models for many and varied physical circumstances, and also occur as reformulations of other mathematical problems Atkinson, (1997). Integral equations can be encountered in various fields of science and numerous applications such as in elasticity, plasticity, heat and mass transfer, oscillation theory, fluid dynamics, filtration theory, electrostatics, electrodynamics, biomechanics, game theory, control, queuing theory, electrical engineering, economics and medicine Polyanin and Manzhirov, (1998). In this paper, linear first kind integral equations type of Fredholm is considered.

Generally, the linear integral equations of the first kind have the form

\[ \int_{a}^{b} K(x,t)y(t)dt = f(x), \quad a \leq x \leq b \]  

where the kernel \( K \) and the function \( f \) are given, and \( y \) is the unknown function to be determined. \( K(x,t) \) is called Fredholm kernel if the kernel in Eq. (1) is continuous on the square \( \Gamma = \{a \leq x \leq b, a \leq t \leq b\} \) or at least square integrable on this square. Then, Eq. (1) with constant integration limits and Fredholm kernel are called first kind linear Fredholm integral equations Polyanin and Manzhurov, (1998). Meanwhile, Eq. (1) also can be rewritten in the operator form as follows

\[ \kappa(y(t)) = \int_{a}^{b} K(x,t)y(t)dt \quad \kappa : S \to T \]  

Definition Maleknejad, et al.,(2006):

Let \( \kappa : S \to T \) be an operator from normed space \( S \) into a normed space \( T \), the equation \( \kappa y = f \) is called well-posed if \( \kappa \) is onto, one to one and \( \kappa^{-1} : T \to U \) is continuous. Otherwise the equation is called ill-posed.

A numerical approach to the solution of integral equations is an essential branch of scientific inquiry. As a matter of fact, some valid methods of solving linear Fredholm integral equations have been developed in recent years. To solve Eq. (1) numerically, we either seek to determine an approximate solution by using the projection method (Chen, et al., 2010; Kangro, 2010; Maleknejad et al., 2006; Muthuvalu and Sulaiman, 2009; Nair and Pereverzev, 2007; Rabbani, 2007), or use the quadrature method (Caldwell, 1994; Laurie, 2001;
The concept of the half-sweep iteration has been introduced by Abdullah, (1991) via the Explicit Decoupled Group (EDG) iterative method to solve two-dimensional Poisson equations. Following to that, applications of the half-sweep iteration iterative methods have been discussed in (Muthuvalu and Sulaiman, 2008; Muthuvalu and Sulaiman, 2008; 2009; Sulaiman et al., 2004; Sulaiman et al., 2007; 2008; Yousif and Evans, 1995). Meanwhile, Othman and Abdullah, (2000) extended the half-sweep iteration concept by introducing quarter-sweep iterative method via the Modified Explicit Group (MEG) iterative method to solve two-dimensional Poisson equations. Further studies to verify the effectiveness of the quarter-sweep iterative methods have been carried out; see (Hasan et al., 2008; Jha and Mishra, 2008; Rakhimov and Othman, 2009; Sulaiman et al., 2004; Sulaiman et al., 2009). Actually, the basic idea of the half- and quarter-sweep iterative methods is to reduce the computational complexities during iteration process, since the implementation of the half- and quarter-sweep iterations will only consider nearly half and quarter of all interior node points in a solution domain respectively. In this paper, we examined the applications of the half- and quarter-sweep iterative concepts with Successive Over-Relaxation (SOR) iterative method by using approximation equation based on projection method for solving problem (1). The standard SOR iterative method is also called as the Full-Sweep Successive Over-Relaxation (FSSOR) method. Meanwhile, combinations of the SOR method with half- and quarter-sweep iterations are called as Half-Sweep Successive Over-Relaxation (HSSOR) and Quarter-Sweep Successive Over-Relaxation (QSSOR) methods respectively.

The remainder of this paper is organized in following way. In Section II, the formulation of the full-, half- and quarter-sweep projection approximation equations will be elaborated. The latter section of this paper will discuss formulations of the FSSOR, HSSOR and QSSOR iterative methods in solving linear system generated from discretization of the Eq. (1). Meanwhile, some numerical results will be shown in fourth section to assert the effectiveness of the proposed method and Section V contains some conclusions.

**Collocation Approximation Equations:**

As afore-mentioned, discretization scheme based on projection method was used to discretize the linear Fredholm integral equations of the first kind. For introducing this method, a sequences of finite dimensional approximating subspaces \( V_n \subset V, n \geq 1 \) with \( V_n \) having dimension \( \kappa_n \) is chosen (for simplicity let \( \kappa_n = \kappa \)) and assume that \( V_n \) has a basis such as \( \{ \phi_1, \cdots, \phi_\kappa \} \). Then, a function for \( y_n \in V_n \) can be written as follows

\[
y_n(t) = \sum_{j=1}^{\kappa} c_j \phi_j(t). \tag{3}\n\]

where \( c_j \) is the unknown coefficients to be determined. By substituting Eq. (3) into (1), we have

\[
r_n(x) = \int_a^b K(x,t) \sum_{j=1}^{\kappa} c_j \phi_j(t) \, dt - f(x) \tag{4}\n\]

where \( r_n \) is called the residual equation when using \( y \approx y_n \). So, this residual in the operator form is defined as follows

\[
r_n = Ky_n - f. \tag{5}\n\]

Fig. 1 shows the finite grid networks in order to form the full-, half- and quarter-sweep projection approximation equations. Based on Fig. 1, the full-, half- and quarter-sweep iterative methods will compute approximate values onto node points of type \( \bullet \) only until the convergence criterion is reached. Then, other approximate solutions at remaining points (points of the different type) can be computed using the direct method (Abdullah, 1991; Othman and Abdullah, 2000).

To facilitate in formulating the projection approximation equation for problem (1), further discussion will be restricted onto piecewise linear collocation method. So, let collocation node points \( x_1, x_2, \cdots, x_\kappa \in \Gamma \) and impose requirements...
Fig. 1: a), b) and c) show the distribution of node points for the full-, half- and quarter-sweep cases respectively.

\[ r_n(x_i) = 0, \quad i = 1, 2, \cdots, \kappa. \]  

Then, we have

\[ \sum_{j=1}^{\kappa} f_j \int_a^b K(x_j, t) \phi_j(t) dt = f(x_j), \quad i = 1, 2, \cdots, \kappa \]  

and by solving Eq. (7) as stated above, the unknown coefficients \( c_j \) are determined. As part of writing Eq. (6) in a more abstract form, we introduce a projection operator \( P_n \) which maps \( V = C(\Gamma) \) onto \( V_n \). By given \( y \in C(\Gamma) \), we define \( P_n y \) to be the element of \( V_n \) which interpolates \( y \) at the nodes \( \{x_i, x_2, \cdots, x_n\} \) as follows

\[ P_n y(t) = \sum_{j=1}^{\kappa} a_j \phi_j(t). \]  

where \( a_j \) are determined by regard to the fact that

\[ P_n y(t_j) = y(t_j), \quad i = 1, 2, \cdots, \kappa. \]  

So, Eq. (5) can be rewritten as

\[ P_n K y_n = P_n f. \]  

Let \( \Gamma = [a, b] \) and \( n \geq 1 \), we define the constant size, \( h \) and discrete set points as follows respectively,

\[ h = \frac{b-a}{n} \]  

and

\[ x_j = a + jh, \quad j = 0, 1, \cdots, n \]  

By introducing the Lagrange basis functions for continuous piecewise linear interpolation
\[ \ell_i(x) = \begin{cases} 1 - \frac{|x-x_i|}{h}, & x_{i-1} \leq x \leq x_{i+1} \\ 0, & \text{otherwise} \end{cases} \] (13)

with the obvious adjustment of the definition for \( \ell_0(x) \) and \( \ell_n(x) \). Then, the projector operator is defined by

\[ P_n y(x) = \sum_{j=1}^{n} y(x_j) \ell_j(x). \] (14)

By applying the projector operator, Eq. (7) takes the simpler form

\[ \sum_{j=0}^{n} y_n(x_j) \int_{a}^{b} K(x_j, t) \ell_j(t) dy = f(x_i) \] (15)

for \( i = 0, p, 2p \cdots n - p \), and \( j = 0, p, 2p \cdots n - p, n \). The value of \( p \), which corresponds to 1, 2 and 4, represents the full-, half- and quarter-sweep cases respectively. The following linear system generated using piecewise linear collocation method can be easily shown as

\[ M y = f \] (16)

where the coefficient matrix \( M \) is a dense matrix. Meanwhile, \( f \) and \( y \) are the right hand side vector and unknown vector to be determined respectively.

**Successive Over-Relaxation Iterative Methods:**

As mentioned in Section I, FSSOR, HSSOR and QSSOR iterative methods will be applied to solve dense linear system generated from the discretization of the problem (1), as shown in Eq. (16). Let matrix \( M \) be decomposed into

\[ M = D - L - U \] (17)

where \( D, L, U \) are diagonal, strictly lower triangular and strictly upper triangular matrices respectively.

Thus, the general scheme for FSSOR, HSSOR and QSSOR methods can be written as

\[ D y^{(k+1)} = \omega L y^{(k)} + \omega U y^{(k)} + \omega f + (1 - \omega) D y^{(k)} \] (18)

where \( \omega \) is a weighted parameter.

Actually, the iterative methods attempt to find a solution to the system of linear equations by repeatedly solving the linear system using approximations to the vector \( y \). Iterations for iterative methods continue until the solution is within a predetermined acceptable bound on the error.

**Numerical Tests:**

In order to compare the performances of the proposed iterative methods described in the previous section, several experiments were carried out on the following two Fredholm integral equations problems. In this paper, we will consider well-posed equations and the case where \( a = 0 \) and \( b = 1 \).

**Example 1** (Rashed, 2003):

\[ \int_{0}^{1} K(x, t) y(t) dt = \frac{1}{6} (x^3 - x), \quad 0 < x < 1 \] (19)

with kernel

\[ K(x, t) = \begin{cases} t(x-1), & t < x \\ x(t-1), & x \leq t \end{cases} \]

The exact solution of the problem is

\[ y(x) = \frac{1}{6} x^3, \quad 0 < x < 1 \]
\( y(x) = x \).

**Example 2 (Rashed, 2003):**

\[
\int_{0}^{1} K(x,t) y(t) \, dt = e^x + (1-e)x - 1, \quad 0 < x < 1
\]

(20)

with kernel

\[
K(x,t) = \begin{cases} 
  f(x-1), & t \leq x \\
  x(t-1), & x < t
\end{cases}
\]

and the exact solution of the problem is given by

\( y(x) = e^x \).

There are three parameters considered in numerical comparison such as number of iterations, execution time and maximum absolute error. As comparisons, the Gauss-Seidel (GS) method acts as the comparison control of numerical results. Throughout the simulations, the convergence test considered the tolerance error of \( e^{-10} \) and carried out on several different values of \( n \). Meanwhile, the experimental values of \( \omega \) were obtained by running the program for different values of \( \omega \) and choosing the one(s) that gave the minimum number of iterations.

Results of numerical simulations, which were obtained from implementations of the GS, FSSOR HSSOR and QSSOR iterative methods for Examples 1 and 2, have been recorded in Tables I and II respectively. Meanwhile, reduction percentages of the number of iterations and execution time for the FSSOR, HSSOR and QSSOR methods compared with GS method have been summarized in Table III.

**Table 1:** Comparison of number of iterations, execution time (in seconds) and maximum absolute error for the iterative methods (example 1)

<table>
<thead>
<tr>
<th>Methods</th>
<th>Number of iterations</th>
<th>Execution time (seconds)</th>
<th>Maximum absolute error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1024</td>
<td>2048</td>
</tr>
<tr>
<td>GS</td>
<td>505</td>
<td>604</td>
<td>725</td>
</tr>
<tr>
<td>FSSOR</td>
<td>395</td>
<td>474</td>
<td>540</td>
</tr>
<tr>
<td>HSSOR</td>
<td>329</td>
<td>395</td>
<td>474</td>
</tr>
<tr>
<td>QSSOR</td>
<td>286</td>
<td>329</td>
<td>395</td>
</tr>
</tbody>
</table>

**Conclusions:**

In the previous section, it has shown that the full-, half- and quarter-sweep projection approximation equations based on piecewise linear collocation method can easily generate a linear system as shown in Eq. (16). Through numerical results obtained for Examples 1 and 2 (Tables I and II), it clearly shows that by applying the family of SOR methods can reduce number of iterations and execution time compared to the GS method; refer Table III. At the same time, it has been shown that applying the half- and quarter-sweep iteration reduces the computational time in the implementation of the iterative method. Besides that, approximate solutions for all SOR methods are also in good agreement compared to the GS method.
Overall, the numerical results show that the QSSOR method is a better method compared to the GS, FSSOR and HSSOR methods in the sense of number of iterations and execution time.

Table 2: Comparison of number of iterations, execution time (in seconds) and maximum absolute error for the iterative methods (example 2)

<table>
<thead>
<tr>
<th>Methods</th>
<th>( n )</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
<th>8192</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS</td>
<td></td>
<td>495</td>
<td>585</td>
<td>703</td>
<td>822</td>
</tr>
<tr>
<td>FSSOR</td>
<td></td>
<td>388</td>
<td>451</td>
<td>512</td>
<td>625</td>
</tr>
<tr>
<td>HSSOR</td>
<td></td>
<td>317</td>
<td>388</td>
<td>451</td>
<td>512</td>
</tr>
<tr>
<td>QSSOR</td>
<td></td>
<td>280</td>
<td>317</td>
<td>388</td>
<td>451</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>( n )</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
<th>8192</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS</td>
<td></td>
<td>35.89</td>
<td>153.45</td>
<td>690.43</td>
<td>2774.88</td>
</tr>
<tr>
<td>FSSOR</td>
<td></td>
<td>24.41</td>
<td>75.54</td>
<td>303.40</td>
<td>1801.44</td>
</tr>
<tr>
<td>HSSOR</td>
<td></td>
<td>6.53</td>
<td>29.31</td>
<td>84.32</td>
<td>398.22</td>
</tr>
<tr>
<td>QSSOR</td>
<td></td>
<td>4.77</td>
<td>7.22</td>
<td>48.54</td>
<td>161.81</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>( n )</th>
<th>1024</th>
<th>2048</th>
<th>4096</th>
<th>8192</th>
</tr>
</thead>
<tbody>
<tr>
<td>GS</td>
<td></td>
<td>8.1149E-9</td>
<td>8.4442E-10</td>
<td>9.5772E-11</td>
<td>2.2256E-11</td>
</tr>
<tr>
<td>FSSOR</td>
<td></td>
<td>1.7222E-9</td>
<td>6.3333E-10</td>
<td>7.0028E-11</td>
<td>1.4355E-11</td>
</tr>
<tr>
<td>HSSOR</td>
<td></td>
<td>2.9910E-9</td>
<td>1.7222E-9</td>
<td>6.3333E-10</td>
<td>7.0028E-11</td>
</tr>
<tr>
<td>QSSOR</td>
<td></td>
<td>9.2101E-9</td>
<td>2.9910E-9</td>
<td>1.7222E-9</td>
<td>6.3333E-10</td>
</tr>
</tbody>
</table>

Table 3: Reduction percentages of the number of iterations and execution time for the FSSOR, HSSOR and QSSOR methods compared with GS method

<table>
<thead>
<tr>
<th>Methods</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSSOR</td>
<td>19.92 - 25.52%</td>
<td>21.61 - 27.17%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSSOR</td>
<td>34.60 - 35.18%</td>
<td>33.67 - 37.72%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QSSOR</td>
<td>43.09 - 45.53%</td>
<td>43.43 - 45.82%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 1</th>
<th>Example 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSSOR</td>
<td>29.04 - 54.57%</td>
<td>31.98 - 56.06%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HSSOR</td>
<td>79.50 - 88.61%</td>
<td>80.89 - 87.79%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>QSSOR</td>
<td>87.19 - 95.57%</td>
<td>86.70 - 95.30%</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

REFERENCES


