Implementation of the Half-Sweep AOR Iterative Algorithm for Space-Fractional Diffusion Equations

Andang Sunarto\textsuperscript{a}, Jumat Sulaiman\textsuperscript{a}, Azali Saudi\textsuperscript{b}

\textsuperscript{a}Faculty of Science and Natural Resources, Universiti Malaysia Sabah, Malaysia
\textsuperscript{b}Faculty of Computing and Informatics, Universiti Malaysia Sabah, Malaysia

Graphical abstract

Abstract

In this paper, we consider the numerical solution of one dimensional space-fractional diffusion equation. The half-sweep AOR (HSAOR) iterative method is applied to solve linear system generated from discretization of one dimensional space-fractional diffusion equation using Caputo’s derivative operator and half-sweep implicit finite difference scheme. Furthermore, the formulation and implementation of HSAOR iterative method to solve the problem are also presented. Two examples and comparisons with FSAOR iterative method are given to show the effectiveness of the proposed method. From numerical results obtained, it has shown that the HSAOR iterative method is superior as compared with the FSAOR methods.

Keywords: HSAOR, space-fractional, Caputo, implicit finite difference scheme

1.0 INTRODUCTION

In this paper we focus on numerical solution for one dimensional space-fractional diffusion equations. Generally, linear space-fractional diffusion equations (SFDE’s) given as follows

\[ \frac{\partial U(x,t)}{\partial t} = a(x)\frac{\partial^\beta U(x,t)}{\partial x^\beta} + b(x)\frac{\partial U(x,t)}{\partial x} + c(x)U(x,t) + f(x,t) \quad (1) \]

with initial condition

\[ U(x,0) = f(x), \quad 0 \leq x \leq \ell, \]

and boundary conditions

\[ U(0,t) = g_0(t), \quad U(\ell,t) = g_1(t), \quad 0 < t \leq T. \]

We describe some necessary definitions and mathematical preliminaries of the fractional derivative theory which are required for our subsequent development of the approximation equation for the problem in Eq. (1).

Definition 1.[1,2] The Riemann-Liouville fractional integral operator, \( J^\beta \), of order \( \beta \) is defined as

\[ J^\beta f(x) = \frac{1}{\Gamma(\beta)} \int_0^x (x-t)^{\beta-1} f(t) dt, \quad \beta > 0, \quad x > 0 \quad (2) \]

Definition 2.[2,3] The Caputo’s fractional partial derivative operator, \( D^\beta \), of order \( \beta \) is defined as

\[ D^\beta f(x) = \frac{1}{\Gamma(m-\beta)} \int_0^x \frac{f^{(m)}(t)}{(x-t)^{\beta-m+1}} dt, \quad \beta > 0 \quad (3) \]

with \( m-1 < \beta \leq m, \quad m \in \mathbb{N}, \quad x > 0. \)

We have the following properties when \( m-1 < \beta \leq m, \quad x > 0: \)

\[ D^\beta_k = 0, \quad (k \text{ is constant }). \]
\[ D^\beta x^n = \begin{cases} \frac{0}{\Gamma(n+1)} x^{-\beta}, & \text{for } n \in \mathbb{N}_0 \text{ and } n < [\beta] \\ \frac{\Gamma(n+1)}{\Gamma(n+1-\beta)} x^{-\beta}, & \text{for } n \in \mathbb{N}_0 \text{ and } n \geq [\beta] \end{cases} \]

where function \([\beta]\) denotes the smallest integer greater than or equal to \(\beta\), \(\mathbb{N}_0 = \{0,1,2,\ldots\}\) and \(\Gamma()\) is the gamma function.

The linear space-fractional diffusion equations occur in multiple diversified phenomena such as physics, finance and biology problems. Therefore numerical treatment is preferred in order to diagnose and solve the problems. In many application areas, it is necessary to use the numerical approach to obtain an approximation solution for the problem in Eq. (1) such as method of line (MOL) [4], implicit finite element method [5], grid-based schemes and Monte-Carlo method [6]. Based on extension work [7], in this paper, discretization scheme based on Caputo’s fractional derivative operator together with implicit finite difference scheme will be implemented to discretize the problem in Eq. (1). Thus, the generated linear system will be solved by using Half-Sweep AOR (HSAOR) iterative method.

Basically, the proposed HSAOR method is inspired by the concept of half-sweep iteration which is introduced by Abdullah [8] via the Explicit Decoupled Group (EDG) iterative method to solve two-dimensional Poisson equations. Actually, the half-sweep iteration concept is essential to reduce the computational complexities during iterative process, because the implementation of half-sweep iteration will only consider nearly half of all node point in a solution domain respectively [9]. In addition to the advantage of this iteration concept, the implementations of this concept in various partial differential equations were further investigated [10, 11, 12]. However, most of problems which have been solved by them are categorized as partial differential equation of integer order. In this work, we discretized space-fractional diffusion equation using implicit finite difference scheme with Caputo’s derivative operator in order to examine the implementation of HSAOR iteration method in solving the resultant linear system of equations. The standard AOR iterative method also known as the FSAOR iterative method is implemented as control method in order to investigate the performance of HSAOR iterative method.

### 2.0 Half-Sweep Caputo’s Implicit Finite Difference Approximation Equations

In this section, the space-fractional diffusion equation (1) is solved. In order to find solution in Eq. (1), let us define \( h = \frac{\ell}{m+1} \), where, \( m=n+1 \) is positive even integer.

By implementing definition (2) we obtain

\[
\frac{\partial^\beta U(x_t, t_n)}{\partial x^\beta} = \frac{1}{\Gamma(2-\beta)} \sum_{j=0,2}^{i-2} g_j^{(2)} \left( U_{i+j+2n} - 2U_{i+j,n} + U_{i+j-2n} \right) \left( \frac{j+1}{2} \right)^{2-\beta} - \left( \frac{j-1}{2} \right) \left( \frac{j+1}{2} \right)^{2-\beta}
\]

Then the discrete approximation equation (4) can be written as

\[
\frac{\partial^\beta U(x_t, t_n)}{\partial x^\beta} = \sigma_{2h} \sum_{j=0,2}^{i-2} g_j^{(2)} \left( U_{i+j+2n} - 2U_{i+j,n} + U_{i+j-2n} \right)
\]

where \( \sigma_{2h} = \frac{(2h)^{2-\beta}}{\Gamma(3-\beta)} \) and \( g_j^{(2)} = \left( \frac{j+1}{2} \right)^{2-\beta} - \left( \frac{j-1}{2} \right) \left( \frac{j+1}{2} \right)^{2-\beta} \).

Then, using implicit finite difference scheme and Caputo’s derivative operator in Eq. (4), we obtain half-sweep Caputo’s implicit finite difference approximation as

\[
\lambda \left( U_{i,n-2} - U_{i,n} \right) = a_i \sigma_{2h} \sum_{j=0,2}^{i-2} g_j^{(2)} \left( U_{i+j+2n} - 2U_{i+j,n} + U_{i+j-2n} \right) + b_i \left( U_{i+2n} - U_{i,n} \right) + C_i U_{i,n} + f_i^n
\]

for \( i = 2,4,\ldots,m-2 \). To simplify the above approximation equation, we get

\[
\lambda U_{i,n-2} = a_i \sigma_{2h} \sum_{j=0,2}^{i-2} g_j^{(2)} \left( U_{i+j+2n} - 2U_{i+j,n} + U_{i+j-2n} \right) - b_i \left( U_{i+2n} - U_{i,n} \right) - C_i U_{i,n} + \lambda U_{i,n} - f_i^n
\]

Again, Eq.(7) can be shown

\[
\lambda U_{i,n} = a_i \sigma_{2h} \sum_{j=0,2}^{i-2} g_j^{(2)} \left( U_{i+j+2n} - 2U_{i+j,n} + U_{i+j-2n} \right) + b_i \left( U_{i+2n} - U_{i,n} \right) - C_i U_{i,n} + \lambda U_{i,n-2} + f_i^n
\]

where

\[
a_1 = a_i \sigma_{2h}, \quad b_1 = \frac{b_i}{2}, \quad c_1 = \frac{C_i}{2}, \quad F_1^n = f_i^n,
\]

Let the series term in Eq. (8) be expanded to get the following approximation equation

\[
-R_1 + a_1 U_{i,2,n} + s_1 U_{i,1} + p_1 U_{i,2,n} + q_1 U_{i,1} + r_1 U_{i,2,n} = f_i^n
\]

Where
linear system (10) would be generally described in Algorithm 1.

Algorithm 1: HSAOR method
i. Initialize \( \widetilde{U} = 0 \) and \( \varepsilon \leftarrow 10^{-10} \).
ii. For \( j = 0, 1, \ldots, n-1 \) implement
   a. For \( i = 1, 2, \ldots, m \) calculate
      \[
      \widetilde{U}^{(k+1)} = (D - \omega L)^{-1} \left[ \beta V + (\beta - \omega) L + (1 - \beta) D \right] \widetilde{U}^{(k)} + \beta (D - \omega L)^{-1} f
      \]
   b. Convergence test. If the convergence criterion \( \| \widetilde{U}^{(k+1)} - \widetilde{U}^{(k)} \| \leq \varepsilon = 10^{-10} \) is satisfied, go to next time level. Otherwise go back to Step (a).
iii. Display approximate solutions.

However if \( p = 1 \), Algorithm 1 will be named as FSAOR

4.0 CONVERGENCE OF AOR METHOD

In this section we will show that convergence of AOR method. We have AOR method for the solution (1) has the form:

\[
\widetilde{U}^{(k+1)} = (D - \omega L)^{-1} \left[ \beta V + (\beta - \omega) L + (1 - \beta) D \right] \widetilde{U}^{(k)} + \beta (D - \omega L)^{-1} f
\]  

with \( n = 0, 1, 2, \ldots \), where

\[
L_{\omega, \beta} = (D - \omega L)^{-1} \left[ (1 - \beta) D + (\beta - \omega) L + \beta V \right] = D - \beta (D - \omega L)^{-1} A
\]

Theorem 4.1. [25](a) If the AOR method (13) converges \( \rho(L_{\omega, \beta}) < 1 \) for some \( \beta, \omega \neq 0 \), then exactly one of the following statements hold:

[i]. \( \omega \in (0, 2) \) and \( \beta \in (-\infty, 0) \cup (0, \infty) \).

[ii]. \( \omega \in (-\infty, 0) \cup (2, \infty) \) and \( \beta \in \left( \frac{2\omega}{(2 - \omega)} , 0 \right) \cup (0, 2) \)

(b) If the AOR method with \( \omega = 0 \) converges \( \rho(L_{0, \beta}) < 1 \) then \( 0 < \beta < 2 \).

Proof. (a) It is known that the eigenvalues \( \lambda_j \) of \( L_{\omega, \beta} \) (\( \beta, \omega \neq 0 \)) are connected with the eigenvalues \( \xi_j \) of \( L_{\omega, 0} = L_{0, \omega} \) (\( L_0 \) is the SOR iteration matrix) by the relationship [20]

\[
\lambda_j = \left( \frac{1}{\omega} + \frac{\beta}{\omega} \right) \xi_j, \quad j = 2(2)m - 2.
\]

From (15) we get \( \xi_j = 1 - \omega \left( \frac{1}{\omega} + \frac{\beta}{\omega} \right) \lambda_j, \quad j = 2(2)m - 2 \). We also have that \( \prod_{j=2,4, \ldots}^{m-2} \xi_j = (1 - \omega)^\eta \). Therefore \[ \prod_{j=2,4, \ldots}^{m-2} \left( 1 - \omega \left( \frac{1}{\omega} + \frac{\beta}{\omega} \right) \right) = (1 - \omega)^\eta \]
and since \( | \lambda_j | < 1, \quad j = 2(2)m - 2 \) from hypothesis, we obtain
\begin{align*}
&\left(1 - \frac{\omega}{\beta}\right)^n = \prod_{j=2,4,..}^{m-2} \left(1 - \frac{\omega}{\beta^2}\right) \leq \prod_{j=2,4,..}^{m-2} \left(1 - \frac{\omega}{\beta} + \frac{\omega}{\beta^2} j^2\right) \\
&< \prod_{j=2,4,..}^{m-2} \left(1 - \frac{\omega}{\beta} + \frac{\omega}{\beta^2} \right) = \left(1 - \frac{\omega}{\beta} + \frac{\omega}{\beta^2}\right)^n, \text{ that is } |1 - \omega| < \left|1 - \frac{\omega}{\beta}\right|
\end{align*}

or equivalently $|\beta(1 - \omega)| < |\beta - \omega| + |\omega|$. \hfill (16)

It can be shown (16) hold if and only if exactly one of the following statement hold:

(i). $\omega \in (0,2)$ and $\beta \in (-\infty,0) \cup (0,\infty)$,

(ii). $\omega \in (-\infty,0) \cap (2,\infty)$ and $\beta \in \left(\frac{2\omega}{2 - \omega},0\right) \cup (0,2)$

and proof of part (a) is completed.

(b). If $\omega = 0$, then $L_{0,\beta} = (1 - \beta)D + \beta(L + U) = (1 - \beta)D + \beta B$. If $\mu_j$, $j = 2(2m-2)$ are the eigenvalues of $B$, then for the eigenvalues $\lambda_j$ of $L_{0,\beta}$ we get

\begin{equation}
\lambda_j = 1 - \beta + \mu_j, \quad j = 2(2m-2), \tag{17}
\end{equation}

with imply $\mu_j = \frac{1}{\beta}(\beta - 1 + \lambda_j)$, $j = 2(2m-2)$ \hfill (18)

But, since $\text{tr} B = 0$ we get

\begin{equation}
\sum_{j=2,4,..}^{m-2} u_j = \sum_{j=2,4,..}^{m-2} \frac{1}{\beta}(\beta - 1 + \lambda_j), \tag{19}
\end{equation}

From (19) we have $\sum_{j=2,4,..}^{m-2} \lambda_j = \left(\frac{m}{2} - 1\right)(1 - \beta)$ and consequently

\begin{equation*}
\left| \frac{m}{2} - 1 \right| (1 - \beta) = \sum_{j=2,4,..}^{m-2} |\lambda_j| \leq \sum_{j=2,4,..}^{m-2} |\lambda_j| < n,
\end{equation*}

since $|\lambda_j| < 1$, $j = 2(2m-2)$ from the hypothesis. Therefore

\begin{equation*}
\left| \frac{m}{2} - 1 \right| (1 - \beta) < n, \text{ or equivalently } 0 < \beta < 2.
\end{equation*}

5.0 NUMERICAL EXPERIMENTS

For the numerical experiments, two examples were considered to verify the effectiveness of the implementation of the HSAOR iterative method. To comparison between FSAOR and HSAOR methods, three criteria will be considered such as number of iterations (K), execution time (second) and maximum error at three different values of $\beta = 1.2$, $\beta = 1.5$ and $\beta = 1.8$ with different mesh sizes as 128, 256, 512, 1024 and 2048. In implementations of two numerical experiments, the convergence test considered the tolerance error $\varepsilon = 10^{-10}$. Results of numerical experiments, which were obtained from implementations of the FSAOR and HSAOR iterative method have been recarded in Tables 1 and 2 respectively.

<table>
<thead>
<tr>
<th>M</th>
<th>Method</th>
<th>$\beta = 1.2$</th>
<th>$\beta = 1.5$</th>
<th>$\beta = 1.8$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>K</td>
<td>Time</td>
<td>Max Error</td>
</tr>
<tr>
<td>128</td>
<td>FSAOR</td>
<td>65</td>
<td>1.32</td>
<td>2.37e-02</td>
</tr>
<tr>
<td></td>
<td>HSAOR</td>
<td>46</td>
<td>0.53</td>
<td>2.24e-02</td>
</tr>
<tr>
<td>256</td>
<td>FSAOR</td>
<td>128</td>
<td>10.00</td>
<td>2.44e-02</td>
</tr>
<tr>
<td></td>
<td>HSAOR</td>
<td>77</td>
<td>2.94</td>
<td>2.37e-02</td>
</tr>
<tr>
<td>512</td>
<td>FSAOR</td>
<td>270</td>
<td>84.05</td>
<td>2.47e-02</td>
</tr>
<tr>
<td></td>
<td>HSAOR</td>
<td>129</td>
<td>19.88</td>
<td>2.44e-02</td>
</tr>
<tr>
<td>1024</td>
<td>FSAOR</td>
<td>577</td>
<td>125</td>
<td>2.49e-02</td>
</tr>
<tr>
<td></td>
<td>HSAOR</td>
<td>278</td>
<td>179.11</td>
<td>2.47e-02</td>
</tr>
<tr>
<td>2048</td>
<td>FSAOR</td>
<td>1150</td>
<td>540</td>
<td>2.52e-02</td>
</tr>
<tr>
<td></td>
<td>HSAOR</td>
<td>606</td>
<td>424</td>
<td>2.49e-02</td>
</tr>
</tbody>
</table>
Table 2 Comparison between number of iterations (K), the execution time (seconds) and maximum errors for the iterative methods using example 2 at $\beta = 1.2, 1.5, 1.8$

<table>
<thead>
<tr>
<th>Method</th>
<th>$\beta = 1.2$</th>
<th>$\beta = 1.5$</th>
<th>$\beta = 1.8$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>K</td>
<td>Time</td>
<td>Max Error</td>
</tr>
<tr>
<td>FSAOR</td>
<td>128</td>
<td>0.93</td>
<td>1.80e-01</td>
</tr>
<tr>
<td>HSAOR</td>
<td>34</td>
<td>0.45</td>
<td>1.73e-01</td>
</tr>
<tr>
<td>FSAOR</td>
<td>97</td>
<td>3.58</td>
<td>1.84e-01</td>
</tr>
<tr>
<td>HSAOR</td>
<td>55</td>
<td>2.67</td>
<td>1.81e-01</td>
</tr>
<tr>
<td>FSAOR</td>
<td>106</td>
<td>18.71</td>
<td>5.39e-01</td>
</tr>
<tr>
<td>HSAOR</td>
<td>97</td>
<td>17.52</td>
<td>1.84e-01</td>
</tr>
<tr>
<td>FSAOR</td>
<td>213</td>
<td>168</td>
<td>5.45e-01</td>
</tr>
<tr>
<td>HSAOR</td>
<td>209</td>
<td>150.23</td>
<td>1.86e-01</td>
</tr>
<tr>
<td>FSAOR</td>
<td>815</td>
<td>398</td>
<td>1.92e-01</td>
</tr>
<tr>
<td>HSAOR</td>
<td>456</td>
<td>273</td>
<td>1.86e-01</td>
</tr>
</tbody>
</table>

Example 1: [3]

We consider the following space-fractional initial boundary value problem

$$
\frac{\partial^\beta U(x,t)}{\partial x^\beta} = d(x) \frac{\partial^\beta U(x,t)}{\partial t^\beta} + p(x,t),
$$

at finite domain $0 \leq x \leq 1$, with the diffusion

$$d(x) = \Gamma(1.2)x^{0.5}.$$

The source function $p(x,t) = (x^2 + 1)\cos(t + 1) - 2x \sin(t + 1)$, with the initial condition $U(x,0) = (x^2 + 1)\sin(t)$ and the boundary conditions $U(0,t) = \sin(t + 1), U(1,t) = 2\sin(t + 1)$, for $t > 0$. The exact solution of this problem is $U(x,t) = (x^2 + 1)\sin(t + 1)$.

Examples 2: [3]

We consider the following space-fractional initial boundary value problem

$$
\frac{\partial^\beta U(x,t)}{\partial t} = \Gamma(1.2)x^\beta \frac{\partial^\beta U(x,t)}{\partial x^\beta} + 3x^2(2x - 1)e^{-x},
$$

with the initial condition $U(x,0) = x^2 - x^3$ and zero Dirichlet conditions. The exact solution of this problem is $U(x,t) = x^2(1 - x)e^{-x}$.

6.0 CONCLUSION

In this work, we discussed the implementation of the HSAOR iterative algorithm which uses two accelerated parameter. The HSAOR Algorithm has performance good speedup and efficiency for computational time and number of iterations. Again, the HSAOR algorithm has shown their superiority over the FSAOR algorithm. For our future works, this study can be extended to investigate on the use of the AOR to combined with the concept quarter-sweep iterative family [26, 27, 28].

References