NUMERICAL SOLUTION FOR THE TIME–SPACE FRACTIONAL PARTIAL DIFFERENTIAL EQUATIONS BY USING THE WAVELET MULTI–SCALE METHOD

H. AMINIKHAH[1], M. TAHMASEBI[2], M. MOHAMMADI ROOZBAHANI[3]

In this paper, a practical method for numerical solutions of the time-space fractional partial differential equations (FPDEs) is presented. The wavelet method based on multiple resolutions is used to solve the FPDE. This method transforms the given FPDE and the boundary conditions to matrix equations with unknown wavelet coefficients which can be solved by a sequential evaluation of two systems, with significantly less computational effort. Theoretical considerations are discussed. For illustration the accuracy and efficiently of the method some numerical examples are presented

Keywords: Fractional diffusion equation; Wavelet numerical method; Multi resolutions Method; Fractional differential equations.

1. Introduction

There is a vast literature on efficient methods for fractional partial differential equations. These equations are important as they arise naturally in many applied areas [1-4, 8, 11, 15, 17, 18-20]. Almost all methods of solving FPDEs are the generalizations of the same strategies for the solutions of PDEs. Multiscale wavelet method for the solution of PDEs are used in many works [6, 10, 14, 16, 22]. Also Mclaren et. al. handled multiscaling collocation method in different way [13]. Their method keeps the different levels of resolution consistent with each other which has a property similar to domain decomposition methods. In the present work, we are interested to combine Adams fractional and the multiscale techniques to solve the fractional partial differential equations efficiently. The main objective of the present work is to extended the multiscaling collocation method in [13] for FPDE. We intend to consider a kind of "generalized diffusion" equation which is referred to the space-time FDE with Robin condition boundary,

[1]Department of Applied Mathematics, School of Mathematical Sciences, University of Guilan, P.O. Box 41938-1914, Rasht, Iran, e-mail: hossein.aminikhah@gmail.com
[2]Department of Applied Mathematics, Faculty of Mathematical Sciences, Tarbiat Modares University, P.O. Box 14115-134 ,Tehran, Iran, e-mail: tahmasebi@modares.ac.ir
[3]Department of Applied Mathematics, School of Mathematical Sciences, University of Guilan, P.O. Box 41938-1914, Rasht, Iran, e-mail: mahmoudroozbahani@gmail.com

$$D_t^{\beta} u(x,t) = Lu(x,t), x \in \Omega = [0,n],\ 0 \le t \le T,\ 0 \le \beta \le 1,$$
$$\begin{cases} u(x,0) = g(x), \\ c_1 u(0,t) + c_2 u_x(0,t) = c_3, \\ c_4 u(n,t) + c_5 u_x(n,t) = c_6, \end{cases} \tag{1}$$

where $L = \sum_{k=0}^{N} f_k(.) D_x^{\alpha_k}$, and $f_k(x)$ are sufficiently well behaved functions on Ω and the operator D_x^{α} is Caputo fractional derivative of order α defined by [15]

$$D_x^{\alpha} F(x) = \frac{1}{\Gamma(m+1-\alpha)} \int_0^x (x-\tau)^{m-\alpha} \frac{\partial^{m+1}}{\partial \tau^{m+1}} F(\tau) d\tau, \tag{2}$$

where $m < \alpha \le m+1, m \in \mathbb{N}$.

In this approach, we utilize cubic B-spline wavelets which are symmetric, compactly supported and smooth enough. The paper is organized as follows:

In section 2 the basic definitions and required properties of the wavelets are briefly mentioned. In section 3 the fractional derivative matrix was approximated by collocation method. In section 4 the wavelets and scaling functions were reshaped to satisfy the boundary conditions exactly. In section 5 we employ the fractional Adams method for time discretization FPDE, then by the operational matrices we convert the FPDE to a linear system. Finally, by multi-resolution method in some subdomains this system divides to some smaller systems which each of them has different resolution and less computation than the primary system, then by combining the solutions of these systems, we derive an approximation of the true solution with less computation. In section 6, the stability of the method is investigated. In section 7 numerical examples are given to demonstrate the validity of the proposed method.

2. Wavelet analysis Preliminaries and notations

In this section, we present some notations, definitions and preliminary facts of the wavelet theory which will be used further in this work. The discrete wavelets constitute a family of functions constructed from dilation and translation of a function called the mother wavelet $\psi(x)$. They are defined by

$$\psi_{j,k}(x) = 2^{\frac{-j}{2}} \psi(2^{-j} x - k). \tag{3}$$

The best way to understand wavelets is through a multi-resolution analysis [5]. Given a function $f(x) \in L_2(\mathbb{R})$, a multi-resolution analysis (MRA) of $L_2(\mathbb{R})$ produces a sequence of subspaces $V_0 \subset \cdots \subset V_j \subset V_{j-1} \subset \cdots$ such that the projections of f onto these spaces give finer approximations of the function f as $j \to -\infty$. There exists $\phi \in V_0$ such that $\phi(x-k), k \in \mathbb{Z}$ is a Riesz basis in V_0. The function ϕ is called the scaling function. As a consequence of above definition, V_j is spanned by $\phi_{j,k}(x) = 2^{-j/2}\phi(2^{-j}x - k)$. One may construct wavelets by first completing the spaces V_j to the space V_{j-1} by the space W_j, i.e., $V_{j-1} = V_j \oplus W_j$. In such away there exists a function ψ such that W_j is spanned by $\psi(2^{-j}x - k)$. The space W_j include all the functions in V_{j-1} that are orthogonal to all those in V_j under $L_2(\mathbb{R})$-inner product. The set of functions which form a basis for the space W_j are called wavelets [3, 4]. For the inclusion $V_0 \subset V_{-1}$ and $W_0 \subset V_{-1}$ there are two important identities:

$$\phi(t) = \sqrt{2}\sum_k h_k \phi(2t-k),\ \psi(t) = \sqrt{2}\sum_k g_k \phi(2t-k). \tag{4}$$

For more details, refer to [5].

Definition (Biorthogonal wavelets): Two functions $\psi, \tilde{\psi} \in L_2(\mathbb{R})$ are called biorthogonal wavelets if each of the sets $\{\psi_{jk} : j,k \in \mathbb{Z}\}$ and $\{\tilde{\psi}_{jk} : j,k \in \mathbb{Z}\}$ be a Riesz basis of $L_2(\mathbb{R})$ and they are biorthogonal to each other in the following sense

$$\langle \psi_{j,k}, \tilde{\psi}_{l,m} \rangle = \delta_{j,l}\delta_{k,m} \text{ for all } j,k,l,m \in \mathbb{Z}.$$

Designing biorthogonal wavelets allows more freedom than orthogonal wavelets. One of them is the possibility of constructing symmetric wavelet functions. Since they define a multi-resolution analysis, the dual functions must satisfy

$$\tilde{\phi}(x) = \sum_k \tilde{h}_k \tilde{\phi}(2x-k) \text{ and } \tilde{\psi}(x) = \sum_k \tilde{g}_k \tilde{\phi}(2x-k). \tag{5}$$

In this work we will use biorthogonal wavelets whose scaling functions are the cubic B-splines:

$$\phi(x) = B_3(x) = \frac{1}{6}\sum_{k=0}^{4}\binom{4}{k}(-1)^k (x-k)_+^3, \tag{6}$$

where $x_+^n = \begin{cases} x^n & x > 0, \\ 0 & x \le 0. \end{cases}$

The fractional derivative of cubic B-spline $B_3(x)$ is given in [11]:

$$D_x^{\alpha} B_3(x) = \frac{1}{\Gamma(4-\alpha)}\sum_{k=0}^{4}\binom{4}{k}(-1)^k (x-k)_+^{3-\alpha}. \tag{7}$$

2.1. Fast Wavelet Transform (FWT)

From $V_{j-1} = V_j \oplus W_j$, every function $v_{j-1} \in V_{j-1}$ can be written uniquely as the sum of a function $v_j \in V_j$ and a function $w_j \in W_j$. Then there exist some coefficients such that

$$\begin{aligned} v_{j-1} &= v_j(x) + w_j(x), \\ \sum_k a_{j-1,k}\varphi_{j-1,k}(x) &= \sum_k a_{j,k}\varphi_{j,k}(x) + \sum_k b_{j,k}\varphi_{j,k}(x). \end{aligned} \tag{8}$$

In other words, we have two representations of the function v_{j-1}, one as an element in V_{j-1} and associated with the sequence $\{a_{j-1,k}\}$, and another as a sum of elements in V_j and W_j associated with the sequences $\{a_{j,k}\}$ and $\{b_{j,k}\}$. The following relations show how to pass between these representations. From (5) and (8) and biorthogonal property of wavelets,

$$a_{j,k} = \sum_i \tilde{h}_i a_{j-1,2k+i}, \tag{9}$$

and

$$b_{j,k} = \sum_i \tilde{g}_i a_{j-1,2k+i}. \tag{10}$$

These formulas define the FWT, let $\overrightarrow{a_{j-1}}$, $\overrightarrow{a_j}$ and $\overrightarrow{b_j}$ be vectors which contain coefficients $\{a_{j-1,k}\}$, $\{a_{j,k}\}$ and $\{b_{j,k}\}$ respectively, then the FWT maps the vector $\overrightarrow{a_{j-1}}$ onto vectors $\overrightarrow{a_j}$ and $\overrightarrow{b_j}$:

$$FWT\left[\overrightarrow{a_{j-1}}\right] = \begin{bmatrix} \overrightarrow{a_j} \\ \overrightarrow{b_j} \end{bmatrix}.$$

For numerical purposes we have to reverse the process to define Inverse fast wavelet transform (IFWT). To do this, taking the inner product of each side of (8) with $\phi_{j-1,k}$, we derive

$$a_{j-1,k} = \sum_n h_{k-2n}\, a_{j,n} + \sum_l g_{k-2l}\, b_{j,l}, \tag{11}$$

so we can define IFWT as following:

$$IFWT\begin{bmatrix} \overrightarrow{a_j} \\ \overrightarrow{b_j} \end{bmatrix} = \left[\overrightarrow{a_{j-1}}\right].$$

3. Matrix approximations

In this work we need operational matrix M^{α} to approximate D_x^{α} on V_j where $0 \le \alpha \le 1$. We will use a collocation based method to calculate them. First we want to approximate any function of $L_2(\mathbb{R})$ as finite series of wavelets and scaling functions. For a fixed $j \le 0$, we use V_j as a basic space, we add extra spaces W_j for increasing the resolution. Let $f_{|V_j}$ denote the projection $f \in L_2(\mathbb{R})$ onto V_j. From $V_{j-1} = V_j \oplus W_j$ we have

$$f_{|V_{j-1}}(x) = \sum_{i=1}^{N} b_{ji}\phi_{j,i}(x) + \sum_{i=1}^{N+1} a_{ji}\psi_{j,i}(x) = \sum_{i=1}^{2N+1} a_{j-1i}\psi_{j-1,i}(x), \tag{12}$$

where $b_{ji} = \left\langle f, \tilde{\phi}_{ji} \right\rangle$, $a_{ji} = \left\langle f, \tilde{\psi}_{ji} \right\rangle$ and $a_{j-1i} = \left\langle f, \tilde{\psi}_{j-1i} \right\rangle$. Since we use compact support wavelet basis, so this property guarantees that in the bounded domain Ω the sum only contains finitely nonzero terms. Thus the function $f_{|\, V_j \oplus Wj}$ in the bounded domain Ω can be expressed as a vector $F = \begin{bmatrix} a \\ b \end{bmatrix}$ what we will usually call "the vector form of $f_{|\, V_j \oplus Wj}$", the vector a contains all coefficients $\{a_{j,k}\}$ and b contains all coefficients $\{b_{j,k}\}$. If F was restricted to subdomain Λ, then only coefficients must be considered whose functions $\phi_{j,i}$ and $\psi_{j.i}$ have support in Λ, we represent it by symbol $F_\Lambda = \begin{bmatrix} a_\Lambda \\ b_\Lambda \end{bmatrix}$. Let P_j be the cubic matrix which converts F the vector form of $f_{|\, V_j}$ into d the vector whose elements are values of f in $x_k = k2^j$, $0 \le k \le N$. Then we have $P_j F = d$. Also, we construct the matrix P_j^α which converts F the vector form of $f_{|\, V_j}$ into d_α the vector whose elements are values of $D_x^\alpha f(x)$ in $x_k = k2^j$, $0 \le k \le N$. Then we have $P_j^\alpha F = d_\alpha$.

Since the basic form of the function f is in the space $V_j \oplus W_j$ so we need the FWT and IFWT for transforming the vectors from the space V_{j-1} to the space $V_j \oplus W_j$ and vice versa. This content is expressed in the following diagram:

$$\begin{array}{lccc} V_j \oplus W_j: & \sum_k a_{jk}\phi_{jk} + \sum_k d_{jk}\psi_{jk} & \xrightarrow{M^\alpha} & \sum_k b_{j-1k}\phi_{jk} + \sum_k c_{j-1k}\psi_{jk} \\ & IFWT \downarrow & & FWT \uparrow \\ V_{j-1}: & \sum_k a_{j-1,k}\phi_{j-1,k} & & \sum_k b_{j-1,k}\phi_{j-1,k} \end{array}$$

$$P_j^\alpha \quad \sum_k a_{j-1,k} D_x^\alpha \phi_{j-1,k}(x_i) \quad (P_j)^{-1}$$

This diagram shows that we can make the fractional derivative matrix as following

$$M^\alpha = FWT \times \left(P_{j-1}\right)^{-1} \times P_j^\alpha \times IFWT. \tag{13}$$

In this method we need to decompose the matrix M^α into some blocks such that the partitions of the matrix M^α must be compatible with the partition of the vector $F = \begin{bmatrix} a \\ b \end{bmatrix}$,

$$M^\alpha \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} Aa + Bb \\ Ca + Db \end{bmatrix}. \tag{14}$$

In addition, if we use the restricted vector $F_\Lambda = \begin{bmatrix} a_\Lambda \\ b_\Lambda \end{bmatrix}$, then the matrix M^α must be adapted with the size of F_Λ, we denote by M^α_Λ.

$$M^\alpha_\Lambda \begin{bmatrix} a_\Lambda \\ b_\Lambda \end{bmatrix} = \begin{bmatrix} A_\Lambda & B_\Lambda \\ C_\Lambda & D_\Lambda \end{bmatrix} \begin{bmatrix} a_\Lambda \\ b_\Lambda \end{bmatrix} = \begin{bmatrix} A_\Lambda a_\Lambda + B_\Lambda b_\Lambda \\ C_\Lambda a_\Lambda + D_\Lambda b_\Lambda \end{bmatrix}. \tag{15}$$

3.1. Advection matrix

One further requirement is the multiplication by the space independent function $g(x)$. We create the linear operator G to approximate the multiplication.

$$FWT \times \left(P_j\right)^{-1} \times G \times P_j \times IFWT, \tag{16}$$

G is a diagonal matrix with the values of function g in $x_k = k2^j$, $0 \le k \le N$.

In the general case, we can represent the operator $L = \sum_k f_k(.) D_x^{\alpha_k}$ in the following matrix form.

$$\begin{aligned} M &= \sum_k FWT \times \left(P_j\right)^{-1} \times F_k \times P_j \times IFWT \times FWT \times \left(P_j\right)^{-1} \times P_j^{\alpha_k} \times IFWT \\ &= \sum_k FWT \times \left(P_j\right)^{-1} \times F_k \times P_j^{\alpha_k} \times IFWT, \end{aligned} \tag{17}$$

where F_k is a diagonal matrix with the values of function f_k in some locations.

4. Boundary conditions

In this section we reshape the wavelets and the scaling functions in V_j whose their support contains $x = 0 (x = n)$, in such a way that they satisfy in the boundary conditions (1), rest are zero at these points. For example in $x = 0$ only three cubic B-spline functions are nonzero, we can make the reshaped scaling function in $x = 0$ by combining these functions:

$$\phi(x) = a\phi_{j,-1}(x) + b\phi_{j,0}(x) + c\phi_{j,+1}(x). \tag{18}$$

The function ϕ must satisfy the boundary conditions, also $L\phi(x)$ must be consistent with these reshaped functions in the boundary conditions, so we have

$$\begin{cases} c_1\phi(0) + c_2\phi'(0) = c_3, \\ c_1 L\phi(0) + c_2 L\phi'(0) = c_3, \end{cases} \tag{19}$$

the coefficients a, b and d are obtained from the above system equations.

5. The proposed method

We consider the fractional Adams method for solving FPDE (1), This method was first studied by Diethelm, Ford and Freed [7]. Their method for solving equation (20) is as follows:

$$D^{\beta} y(t) = f\left(t, y(t)\right),\ y(0) = y_0,\ 0 < \beta < 1, \tag{20}$$

$$y_{n+1} = y_0 + \frac{h^{\beta}}{\Gamma(\beta+2)}\left(\sum_{k=0}^{n} c_{k,n+1} f\left(t_k, y_k\right) + c_{n+1,n+1} f\left(t_{n+1}, y_{n+1}\right)\right), \tag{21}$$

where

$$c_{k,n+1} = \begin{cases} n^{\beta+1} - (n-\beta)(n+1)^{\beta}, & k = 0, \\ (n-k+2)^{\beta+1} + (n-k)^{\beta+1} - 2(n-k+1)^{\beta+1}, & 1 \le k \le n, \\ 1, & k = n+1. \end{cases} \tag{22}$$

and $h = \frac{T}{N}$, $\{t_k = kh : k = 0,1,..,N\}$, $y_k \approx y(t_k)$.

Let u_n denote the exact solution $u(.,t_n)$ and $\tilde{u}_n$ denote approximation solution of it. So approximation solution for the time space-fractional diffusion equation (1) by using the fractional Adam's method would be

$$\tilde{u}_{n+1} = \tilde{u}_0 + \frac{h^{\beta}}{\Gamma(\beta+2)}\left\{\sum_{k=0}^{n} c_{k,n+1} L\tilde{u}_k + L\tilde{u}_{n+1}\right\}. \tag{23}$$

Now we can take the space V_{j-1} to approximate the solution of equation (23). If we consider the vector form $\begin{bmatrix} a^k \\ b^k \end{bmatrix}$ of $u(x,t_k)$ in $V_j \oplus W_j$, Then from (23) and the definition of M we have

$$\left(I - \frac{h^{\beta}}{\Gamma(\beta+2)} M\right)\begin{bmatrix} a^{n+1} \\ b^{n+1} \end{bmatrix} = \begin{bmatrix} a^{0} \\ b^{0} \end{bmatrix} + \frac{h^{\beta}}{\Gamma(\beta+2)}\sum_{k=0}^{n} c_{k,n+1} M \begin{bmatrix} a^{k} \\ b^{k} \end{bmatrix}. \tag{24}$$

We use the multiscaling method to solve this system to avoid growing our calculations and has more accurate solution in the subdomain Λ. This means that once we solve the system in a space V_j and domain Ω. Once again we solve the system in a finer space V_{j-1} and subdomain Λ. Combination of these two systems makes suitable accuracy and the less calculations than the solutions of the system in the space V_{j-1} on domain Ω. In the beginning we consider the matrix M in the space V_{j-1}, since in the first step we will not be using all of M so we decompose the matrix M into some blocks $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$.

We only consider the block A which operates as the operator L in the space V_j over all domain Ω. Then using time stepping scheme (23) we find an approximation for a^{n+1} which we denote by a_{Λ}^{Tm}

$$\left(I-\frac{h^{\beta}}{\Gamma(\beta+2)}A\right)a^{Tm}=a^{0}+\frac{h^{\beta}}{\Gamma(\beta+2)}\sum_{k=0}^{n}c_{k,n+1}Aa^{k}. \tag{25}$$

Next, we are looking for vector correction where $a^{n+1}=a^{Tm}+a^{Cr}$. What we want now is to solve the system on Λ in the V_{j-1}. Consider the fractional Adam's method for this case

$$\left(I-\frac{h^{\beta}}{\Gamma(\beta+2)}\begin{bmatrix}A & B\\ C & D\end{bmatrix}\right)_{\Lambda}\begin{bmatrix}a^{n+1}\\ b^{n+1}\end{bmatrix}_{\Lambda}=\begin{bmatrix}a^{0}\\ b^{0}\end{bmatrix}_{\Lambda}+\frac{h^{\beta}}{\Gamma(\beta+2)}\sum_{k=0}^{n}c_{k,n+1}\begin{bmatrix}A & B\\ C & D\end{bmatrix}_{\Lambda}\begin{bmatrix}a^{k}\\ b^{k}\end{bmatrix}_{\Lambda}. \tag{26}$$

Since $a_{\Lambda}^{n+1}=a_{\Lambda}^{Tm}+a_{\Lambda}^{Cr}$ thus

$$\begin{aligned}\left(I-\frac{h^{\beta}}{\Gamma(\beta+2)}\begin{bmatrix}A & B\\ C & D\end{bmatrix}\right)_{\Lambda}&\left(\begin{bmatrix}a^{Cr}\\ b^{n+1}\end{bmatrix}_{\Lambda}+\begin{bmatrix}a^{Tm}\\ 0\end{bmatrix}_{\Lambda}\right)\\ &=\begin{bmatrix}a^{0}\\ b^{0}\end{bmatrix}_{\Lambda}+\frac{h^{\beta}}{\Gamma(\beta+2)}\sum_{k=0}^{n}c_{k,n+1}\begin{bmatrix}A & B\\ C & D\end{bmatrix}_{\Lambda}\begin{bmatrix}a^{k}\\ b^{k}\end{bmatrix}_{\Lambda},\end{aligned} \tag{27}$$

then

$$\begin{aligned}\left(I-\frac{h^{\beta}}{\Gamma(\beta+2)}\begin{bmatrix}A & B\\ C & D\end{bmatrix}\right)_{\Lambda}&\begin{bmatrix}a^{C_r}\\ b^{n+1}\end{bmatrix}_{\Lambda}\\ &=\begin{bmatrix}0\\ b^{0}\end{bmatrix}_{\Lambda}+\frac{h^{\beta}}{\Gamma(\beta+2)}\begin{bmatrix}\sum_{k=0}^{n}c_{k,n+1}Bb^{k}\\ C\left(a^{T_m}+\sum_{k=0}^{n}c_{k,n+1}a^{k}\right)+D\sum_{k=0}^{n}c_{k,n+1}b^{k}\end{bmatrix}_{\Lambda}.\end{aligned} \tag{28}$$

The vector $\begin{bmatrix}a^{n+1}\\ b^{n+1}\end{bmatrix}_{\Lambda}$ is obtained by solving the above system. The last step is increase the accuracy of the approximated vector $\begin{bmatrix}a^{Tm}\\ 0\end{bmatrix}_{\Omega}$. We construct the vector $\begin{bmatrix}a^{n+1}\\ b^{n+1}\end{bmatrix}_{\Omega}$ by replacing the elements of $\begin{bmatrix}a^{Tm}\\ 0\end{bmatrix}_{\Omega}$ by the elements of $\begin{bmatrix}a^{n+1}\\ b^{n+1}\end{bmatrix}_{\Lambda}$, the only ones that are related to subdomain Λ. This completes the method.

Now, we present the algorithm of the proposed method. In this algorithm j, h and g are resolution level, time step and initial function respectively. If the vector $a=[a_0,a_2,\ldots,a_N]^T$ be the vector form of a function in V_j then we suppose the restricted vector a_{Λ} is $[a_r,\ldots,a_{s+1}]^T$.

5.1. Algorithm:

1. $v \leftarrow [g_0, g_1, \ldots, g_{2N}]$ where $N = m2^{-j}$ and $g_k = g(k2^j)$ $k = 0, \ldots, 2N$

2. $\begin{bmatrix} a \\ b \end{bmatrix} \leftarrow FWT \times P_j^{-1} \times v$

3. Constructing matrix M by using (17)

4. Blocking the matrix $M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$ where $A \leftarrow M(1:N+1, 1:N+1)$, $B \leftarrow M(1:N+1, N+1:2N+1)$ and so on for C and D

5. $M_\Lambda = \begin{bmatrix} A_\Lambda & B_\Lambda \\ C_\Lambda & D_\Lambda \end{bmatrix}$ Limiting M to subdomain Λ, where $A_\Lambda \leftarrow A(r:s+1, r:s+1)$, $B_\Lambda \leftarrow B(r:s+1, r:s)$ and so on for C_Λ and D_Λ

6. For $n = 0$ to k do

7. $\begin{bmatrix} a_n \\ b_n \end{bmatrix} \leftarrow \begin{bmatrix} a \\ b \end{bmatrix}$

8. $a_\Lambda \leftarrow a(r:s+1)$, $b_\Lambda \leftarrow b(r:s)$

9. Solve the system (25) to get vector a^{Tm}

10. $a_\Lambda^{Tm} \leftarrow a^{Tm}(r:s+1)$

11. Solve the system (28) to get vector $\begin{bmatrix} a_\Lambda^{Cr} \\ b_\Lambda \end{bmatrix}$

12. $a_\Lambda \leftarrow a_\Lambda^{Cr} + a_\Lambda^{Tm}$,

13. $a^{Tm}(r:s+1) \leftarrow a_\Lambda$, $b(r:s) \leftarrow b_\Lambda$, $v \leftarrow \begin{bmatrix} a^{Tm} \\ b \end{bmatrix}$

14. End for

6. Stability

In order to show stability of the approximate solution, we recall discrete Gronwall lemma.

Lemma (Discrete Gronwall Lemma): If $\{y_k\}$, $\{f_k\}$, and $\{g_k\}$ are nonnegative sequences and

$$y_n \le f_n + \sum_{0 \le k \le n} g_k y_k \qquad for\ n \ge 0, \tag{29}$$

then

$$y_n \le f_n + \sum_{0 \le k \le n} f_k g_k \exp\left(\sum_{k \le i \le n} g_i \right) \qquad for\ n \ge 0. \tag{30}$$

If, in addition, $\{f_k\}$ is nondecreasing then

$$y_n \le f_n \exp\left(\sum_{0\le i\le n} g_i\right) \quad for\ n \ge 0. \tag{31}$$

Now, let $\tilde{u}_n$ denote the approximation solution of equation (23), and U_n be the vector form of $\tilde{u}_n$ in the V_j and $\overline{U}_n = \left[\tilde{u}(x_0,t_n),\tilde{u}(x_1,t_n),\ldots,\tilde{u}(x_N,t_n)\right]^T$ where $x_k = k2^j$, $0 \le k \le N$. Since $P_j U_n = \overline{U}_n$, from equation (23) we have

$$P_j^{-1}\overline{U}_{n+1} = P_j^{-1}\overline{U}_0 + \frac{h^\beta}{\Gamma(\beta+2)}\left(\sum_{k=0}^{n} c_{k,n+1} M P_j^{-1}\overline{U}_k + M P_j^{-1}\overline{U}_{n+1}\right). \tag{32}$$

Choosing h small enough that $\frac{h^\beta}{\Gamma(\beta+2)}\|M\| < \frac{1}{2} < 1$ guarantees nonsingularity of the matrix $I - \frac{h^\beta}{\Gamma(\beta+2)} M$, then

$$\left\|\left(I - \frac{h^\beta}{\Gamma(\beta+2)} M\right)^{-1}\right\| \le \frac{1}{1 - \frac{h^\beta}{\Gamma(\beta+2)}\|M\|} < 2,$$

therefore, we have

$$\begin{aligned}\|\overline{U}_{n+1}\| \le & \left\|P_j\left(I - \frac{h^\beta}{\Gamma(\beta+2)} M\right)^{-1} P_j^{-1}\right\|\|\overline{U}_0\| \\ & + \frac{h^\beta}{\Gamma(\beta+2)}\left\|P_j\left(I - \frac{h^\beta}{\Gamma(\beta+2)} M\right)^{-1} P_j^{-1} M\right\| \sum_{k=0}^{n} c_{k,n+1}\|\overline{U}_k\|.\end{aligned} \tag{33}$$

Since $\left\|\left(I - \frac{h^\beta}{\Gamma(\beta+2)} M\right)^{-1}\right\| < 2$ and $h = \frac{T}{N}$, we have

$$\|\overline{U}_{n+1}\| \le 2\|P_j\|\|P_j^{-1}\|\|\overline{U}_0\| + \frac{2\left(\frac{T}{N}\right)^\beta}{\Gamma(\beta+2)}\|P_j\|\|P_j^{-1}\|\|M\| \sum_{k=0}^{n} c_{k,n+1}\|\overline{U}_k\|. \tag{34}$$

By applying Gronwall's inequality, we obtain

$$\|\overline{U}_{n+1}\| \le C_1\|\overline{U}_0\|\exp\left(C_2 T^\beta \sum_{k=0}^{n} \frac{c_{k,n+1}}{N^\beta}\right), \tag{35}$$

since $\frac{c_{k,n+1}}{N^\beta} = \frac{(n-k+2)^{\beta+1}}{N^\beta} + \frac{(n-k)^{\beta+1}}{N^\beta} - 2\frac{(n-k+1)^{\beta+1}}{N^\beta} \le \frac{2}{N}$ is bounded and increasing function with respect to β so we have

$$\|\overline{U}_{n+1}\| \le C_1\|\overline{U}_0\|\exp\left(C_2 T^\beta 2\right), \tag{36}$$

where $C_1 = 2\|P_j\|\|P_j^{-1}\|$ and $C_2 = \frac{4h}{\Gamma(\beta+2)}\|P_j\|\|P_j^{-1}\|\|M\|$, this completes the proof of stability.

7. Numerical Examples

Example 7.1. We consider the following time space fractional differential equation

$$D_t^{\beta} u(x,t) = D_t^{\alpha} u(x,t),$$

the initial condition and the boundary conditions are as follows:

$$u(1,t) = u(20,t) = 0, 0 \le t \le 2, \ u(x,0) = \begin{cases} \exp\left(\frac{-1}{1-(x-2)^2}\right), & 3 < x < 4, \\ 0, & \textit{otherwise.} \end{cases}$$

For comparison, the example 1 was solved numerically in different levels of resolutions. Table 1 shows the convergence when j decreases, also the Figure 1 shows in different times the approximated results satisfy the boundary conditions exactly.

Table 1. The errors are the difference between the V_j results and The V_{j-1} results (V_j / V_{j-1}) with h=0.01 at t=0.5.

$\alpha = \beta = 0.9$						
x	V_{-3}/V_{-4}	V_{-4}/V_{-5}	V_{-5}/V_{-6}	V_{-6}/V_{-7}	V_{-7}/V_{-8}	solution on V_{-8}
1.5	0.0030298	-0.000261	-6.163E-6	-3.025E-7	-1.516E-8	3.5574E-7
2	0.0007083	7.9038E-6	8.4961E-7	9.6163E-8	1.0941E-8	0.11612065
2.5	0.000198	-9.662E-6	1.9387E-9	-1.41E-9	7.848E-11	0.28671766
3	0.000521	-1.194E-5	-3.205E-7	-1.114E-8	4.754E-10	0.29474804
3.5	-0.000733	-0.000135	-1.081E-5	-8.237E-7	-6.865E-8	0.10448197
$\alpha = \beta = 0.7$						
x	V_{-3}/V_{-4}	V_{-4}/V_{-5}	V_{-5}/V_{-6}	V_{-6}/V_{-7}	V_{-7}/V_{-8}	solution on V_{-8}
1.5	-3.29E-6	-2.831E-7	-1.968E-9	-6.91E-11	8.928E-12	2.3211E-9
2	0.0002645	1.4228E-6	-2.444E-7	-6.756E-9	-7.375E-10	0.00223121
2.5	0.0002652	3.0337E-6	-2.905E-7	-2.917E-9	-2.156E-10	0.03290537
3	0.0002006	3.0142E-6	-1.995E-7	-1.829E-10	1.453E-10	0.08704351
3.5	0.0001583	3.6176E-6	3.9999E-9	1.5924E-8	2.7148E-9	0.12114837

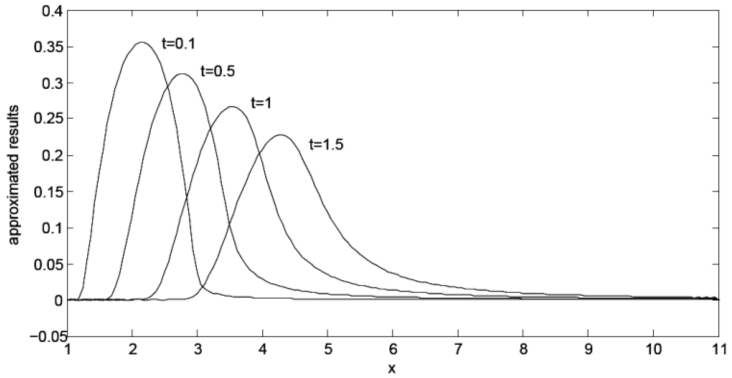

Fig. 1. This figure shows approximated solution in different times in example 1

Example 7.2.we consider the following fractional equation:

$$D_t^{\alpha}u(x,t)=\frac{\partial^2}{\partial x^2}u(x,t)-u(x,t)$$

The initial condition and the boundary conditions are as follows:

$$\begin{cases} u(x,0)=B_3(x-3)+2B_3(x-3)+B_3(x-3)-B_3(x-3)-2B_3(x-3)-B_3(x-3), \quad 1<x<11 \\ u(1,t)=u(11,t)=0, \quad 0\le t\le 1 \end{cases}$$

The example was solved by presented multi-scaled method with V_{-5} on Ω and V_{-6} on Λ. The figure 2 and table 2 show that the accuracy can be improved by enlarging subdomain.

Table 2.

The errors are the difference between the multi-scale results and The results obtained using V_{-6}, with h=0.01 at t=0.5.

x	2.5	4.5	5.5	7.5	8.5	10.5
error $\Lambda=[4,8]$	-2.0330E-6	4.3833E-7	-2.2797E-7	2.1657E-7	-2.1980E-6	1.0790E-6
error $\Lambda=[2,10]$	-4.4215E-8	8.7666E-8	-4.5595E-8	4.3315E-8	-8.7919E-8	2.1579E-7
solution on V_{-6}	0.07538483	1.44352103	1.19066365	-0.6054038	-1.4118151	-0.3922880

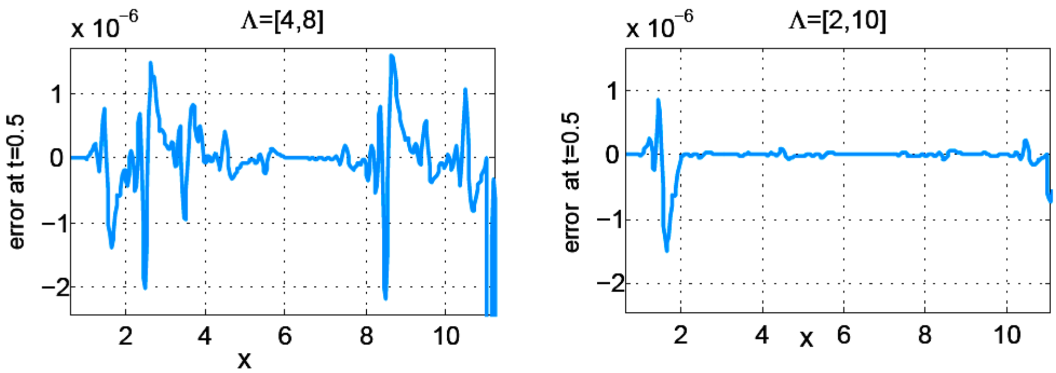

Fig. 2. This figure compare the errors of the presented method in different subdomains.

8. Conclusions

In this work a practical approach for solving time space fractional partial differential equation is presented. Multi scaling method via wavelets is used to increase resolution in some locations, furthermore the computations are reduced because of the compact support of wavelets, and also wavelets are employed in such a way that they satisfy the boundary conditions exactly. The method can be extended to nonlinear FPDEs that is now under progress.

Acknowledgement

We are very grateful to two anonymous referees for their careful reading and valuable comments which led to the improvement of this paper. We would like to thank Dr. Donald A. McLaren for his useful comments.

R E F E R E N C E S

[1] *B. Baeumer, M. M. Meerschaert, D. A. Benson and S. W. Wheatcraft*, ''Subordinated advection–dispersion equation for contaminant transport'', Water Resour. Res., **Vol. 37**, no. 6, 2001, pp. 1543–1550.

[2] *D. Benson, R. Schumer, M. Meerschaert and S. Wheatcraft*, ''Fractional dispersion, Levy motions and the MADE tracer tests'', Transport Porous Media, **Vol. 42**, no. 1, 2001, pp. 211–240.

[3] *D. Benson, S. Wheatcraft and M. Meerschaert*, ''Application of a fractional advection–dispersion equation'', Water Resour. Res., **Vol. 36**, no. 6, 2000, pp. 1403–1412.

[4] *D. Benson, S. Wheatcraft and M. Meerschaert*, ''The fractional-order governing equation of Levy motion'', Water Resour. Res., **Vol. 36**, no. 6, 2000, pp. 1413–1424.

[5] *C. Blatter*, ''Wavelets, A Primer'', A K Peters/CRC Press, 2002.

[6] *W. Dahmen, A. Kurdila and P. Oswald*, ''Multiscale Wavelet Methods for Partial Differential Equations'', Academic Press, Toronto, 1997.

[7] *K. Diethelm, N.J. Ford and A.D. Freed*, ''Detailed error analysis for a fractional Adams method'', Numer. Algorithms, **Vol. 36**, no. 1, 2004, pp. 31–52.

[8] *R. Gorenflo, F. Mainardi, E. Scalas and M. Raberto*, ''Fractional calculus and continuous-time finance III, The diffusion limit, Mathematical finance (Konstanz, 2000)'', Trends in Math., Birkhuser, Basel, 2001, pp. 171–180.

[9] *J.C. Goswami and Andrew K. Chan*, ''Fundamentals of Wavelets. Theory, Algorithms and Applications'', John Wiley and Sons, New York, 1999.

[10] *J. S. Hesthaven and L. M. Jameson*, ''A Wavelet optimized adaptive multi-domain method'', Journal of Computational Physics, **Vol. 145**, no. 1, 1998, pp. 280-296.

[11] *X. Li*, ''Numerical solution of fractional equations using cubic b-spline wavelet collocation method'', Commun. Nonlinear Sci. Numer. Simulat., **Vol. 17**, no. 10, 2012, pp. 3934-3946.

[12] *S. Mallat*, ''A Wavelet Tour of Signal Processing'', Academic Press, New York, 1999.

[13] *D. A. McLaren*, ''Sequential and localized implicit wavelet-based solvers for stiff partial differential equations'', Ph.D. Thesis, 2012.

[14] *M. Mehra and N.K.-R. Kevlahan*, ''An adaptive multilevel wavelet solver for elliptic equations on an optimal spherical geodesic grid'', SIAM J. Sci. Comput., **Vol. 30**, no. 6, 2008, pp. 3073–3086.

[15] *K.S. Miller and B. Ross*, ''An Introduction to the Fractional Calculus and Fractional Differential Equations'', John Wiley and Sons, New York, 1993.

[16] *S. Muller and Y. Stiriba*, ‘‘A multilevel finite volume method with multiscale-based grid adaptation for steady compressible flows’’, J. Comput. Appl. Math., **Vol. 277**, no. 2, 2009, pp. 223–233.
[17] *I. Podlubny*, ‘‘Fractional Differential Equations’’, Academic Press, SanDiego, CA, 1999.
[18] *M. Raberto, E. Scalas and F.Mainardi*, ‘‘Waiting-times and returns in high-frequency 5nancial data: an empirical study’’, Physica A, **Vol. 314**, no. 1, 2002, pp. 749–755.
[19] *L. Sabatelli, S. Keating, J. Dudley and P. Richmond*, ‘‘Waiting time distributions in 5nancial markets’’, The European Physical Journal B, **Vol. 27**, no. 2, 2002, pp. 273–275.
[20] *R. Schumer, D.A. Benson, M.M. Meerschaert and B. Baeumer*, ‘‘Multiscaling fractional advection–dispersion equations and their solutions’’, Water Resour. Res., **Vol. 39**, no. 1, 2003, pp. 1022–1032.
[21] *W. Sweldens*, ‘‘The Construction and Application of Wavelets in Numerical Analysis’’, Ph.D. Thesis, Leuven, 1995.
[22] *O.V. Vasilyev and S. Paolucci*, ‘‘A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain’’, J. Comput. Phys., **Vol. 125**, no. 2, 1996, pp. 498–512.