The Multi-scale Method for Solving Nonlinear Time Space Fractional Partial Differential Equations

Hossein Aminikhah, Mahdieh Tahmasebi, and Mahmoud Mohammadi Roozbahani

***Abstract*—In this paper, we present a new algorithm to solve a kind of nonlinear time space-fractional partial differential equations on a finite domain. The method is based on B-spline wavelets approximations, some of these functions are reshaped to satisfy on boundary conditions exactly. The Adams fractional method is used to reduce the problem to a system of equations. By multi-scale method this system is divided into some smaller systems which have less computations. We get an approximated solution which is more accurate on some subdomains by combining the solutions of these systems. Illustrative examples are included to demonstrate the validity and applicability of our proposed technique, also the stability of the method is discussed.**

***Index Terms*—Adams fractional method, B-spline wavelets, multi-scale method, nonlinear fractional partial differential equations.**

I. INTRODUCTION

IN the last few decades fractional differential equations have found applications in several different disciplines of science and technology including physics, biology, engineering [1]−[3], viscoelasticity [4], finance [5]−[7], hydrology [8]−[13], and control systems [14]. Several numerical methods are proposed to solve these equations such as the finite difference methods [15]−[20], Laplace transformation method [21], [22], Fourier transformation method [23], [24], the Adomian decomposition method [25], variational iteration method [26] and multi-scale methods [27]−[32]. Also Aminikhah *et al.* handled multiscaling collocation method to solve linear Fractional Partial Differential Equations (FPDE) [33]. They combined Adams fractional method and the multi-scale techniques to solve the linear fractional partial differential equations. This paper continues this line of approach. We intend to consider a kind of the nonlinear time-space FDE with Robin condition boundary,

$$D_t^{\beta} u(x,t) = \sum_{k=1}^{S} f_k(x)\, D_x^{\alpha_k} u(x,t)\, D_x^{\beta_k} u(x,t)$$
$$\begin{cases} u(x,0) = g(x) \\ c_1 u(0,t) + c_2 u_x(0,t) = c_3 \\ c_4 u(n,t) + c_5 u_x(n,t) = c_6 \end{cases} \tag{1}$$

where $x \in \Omega = [0, m]$, $0 \le t \le T$, and $f_k(x)$ are bounded functions on Ω and the operator D_x^{α}, (D_t^{β}) is caputo space time fractional derivative of order $\alpha_k(\beta)$ defined by [2]

$$D_x^{\alpha} u(x,t) = \begin{cases} \frac{\partial^{\alpha}}{\partial \tau^{\alpha}} u(x,t), & \alpha = m \in \mathbb{N} \cup \{0\} \\ \frac{1}{\Gamma(m-\alpha)} \int\limits_0^x (x-\tau)^{m-\alpha} \frac{\partial^m}{\partial \tau^m} u(\tau,t)\, d\tau \\ \qquad m-1 < \alpha < m \end{cases} \tag{2}$$

$$D_t^{\beta} u(x,t) = \begin{cases} \frac{\partial^{\beta}}{\partial \tau^{\beta}} u(x,t), & \beta = p \in \mathbb{N} \cup \{0\} \\ \frac{1}{\Gamma(p-\beta)} \int\limits_0^t (t-\tau)^{p-\beta} \frac{\partial^p}{\partial \tau^p} u(x,\tau)\, d\tau \\ \qquad p-1 < \beta < p. \end{cases} \tag{3}$$

First, we present the differential operator in matrix form by using collocation method. Second, we need a time stepping scheme to convert FPDE to an implicit linear system. Finally, by dividing the domain Ω into several smaller subdomains, the system can be divided into smaller systems, then each of them will have 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.

The paper is organized as follows: In section II the basic definitions and required properties of the wavelet are briefly mentioned. In the next two sections we provided essential tools for constructing our method. In section III the fractional derivative matrix was approximated by collocation method. In section IV the wavelets and scaling functions were reshaped to satisfy the boundary conditions, so the approximated solution will be exact in the boundary points. In section V we employ the fractional Adams method for time discretization FPDE then describe how to construct a system by operational matrices where introduced in previous sections, finally by multi-resolution method in some subdomains this system divides to some smaller systems which involve less computation. In section VI, the stability of the method is investigated. In section VII numerical examples are given to demonstrate the validity of the proposed method.

II. PRELIMINARIES AND NOTATIONS

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

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

where

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

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

Recommended by Associate Editor Dingyü Xue.

H. Aminikhah and M. M. Roozbahani are with the Department of Applied Mathematics, University of Guilan, Rasht 41938-19141, Iran (e-mail: aminikhah@guilan.ac.ir; mahmoudroozbahani@gmail.com).

M. Tahmasebi is with the Department of Applied Mathematics, Faculty of Mathematical Sciences, Tarbiat Modares University, Tehran 14115-137, Iran (e-mail: tahmasebi@modares.ac.ir).

Digital Object Identifier 10.1109/JAS.2016.7510058

The best way to understand wavelets is through a multi-resolution analysis.

Definition 1: (Multi-resolution Analysis) A multi-resolution analysis of $L_2(\mathbb{R})$ with inner product $\langle .,.\rangle$, is defined as a sequence of closed subspaces $V_j \subset L_2(\mathbb{R})$, $j \in \mathbb{Z}$ with the following properties

1) $\cdots \subset V_1 \subset V_0 \subset V_{-1} \subset \cdots$.

2) $\underset{j\in\mathbb{Z}}{\cup} V_j$ is dense in $L_2(\mathbb{R})$ and $\underset{j\in\mathbb{Z}}{\cap} V_j = \emptyset$.

3) If $f(x) \in V_0$ then $f\left(2^{-j}x\right) \in V_j$.

4) If $f(x) \in V_0$ then $f\left(2^{-j}x - k\right) \in V_j$.

5) $\phi(x-k)$, $k \in \mathbb{Z}$ is a Riesz basis in V_0. As a consequence of Definition 1, V_j is spanned by $\phi_{j,k}(x) = 2^{-j/2}\phi\left(2^{-j}x - k\right)$. One may construct wavelets by first completing the spaces V_j to the space V_{j-1} by means of a space W_j, i.e., $V_{j-1} = V_j \oplus W_j$. From the inclusion $V_0 \subset V_{-1}$ we have the important identity, called scaling equation of the form

$$\phi(t) = \sqrt{2}\sum_k h_k \phi(2x-k) \tag{5}$$

where $h_k = \dfrac{\sqrt{2}}{2^4}\begin{pmatrix} 4 \\ k \end{pmatrix}, 0 \le k \le 4$.

Also from $W_0 \subset V_{-1}$ we have

$$\psi(t) = \sqrt{2}\sum_k g_k \phi(2x-k). \tag{6}$$

For more details refer to [34]−[37].

The dual cubic B-spline bases involve another multi-resolution analysis of $L_2(\mathbb{R})$, it is usually denoted by $\left\{\widetilde{V}_j\right\}$. The dual scaling function $\widetilde{\phi}$ is biorthogonal to ϕ in the following sense

$$\left\langle \phi_{j,k}, \tilde{\phi}_{l,m}\right\rangle = \delta_{j,l}\delta_{k,m}, for\ all\ j,k,l,m \in \mathbb{Z}.$$

$\widetilde{\phi}(x-k)$, $k \in \mathbb{Z}$ produces a Riesz basis for the space $\widetilde{V}_0$. The dual wavelet $\widetilde{\psi}$ is constructed by taking linear combinations of the dual scaling functions

$$\widetilde{\psi}(x) = \sqrt{2}\sum_k \widetilde{g}_k \widetilde{\phi}(2x-k). \tag{7}$$

Also from $\widetilde{V}_0 \subset \widetilde{V}_{-1}$ we have

$$\widetilde{\phi}(x) = \sqrt{2}\sum_k \widetilde{h}_k \widetilde{\phi}(2x-k) \tag{8}$$

where $\widetilde{g}_k = (-1)^k h_{1-k}$ and $\widetilde{h}_k = (-1)^k g_{1-k}$. The function $\tilde{\psi} \in L_2(\mathbb{R})$ is biorthogonal to ψ.

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

Designing biorthogonal wavelets allows more freedom than orthogonal wavelets. One of them is the possibility of constructing symmetric wavelet functions.

Any function of V_j can be represented by finite series of cubic B-splines. Let $f_{|V_j}$ denote the projection $f \in L_2(\mathbb{R})$ onto V_j. We can obtain the cubic B-spline expansion of $f_{|V_j}$ as

$$f_{|\ V_{j-1}}(x) = \sum_{i=0}^{2N} a_{j-1,\,i}\phi_{j-1,\,i}(x) \tag{9}$$

where $a_{j-1,\,i} = \left\langle f, \tilde{\phi}_{j-1,\,i}\right\rangle$ $N = m2^{-j}$. Note that the cubic B-splines have compact support, so this property guarantees that in the bounded domain $\Omega = [0, m]$ the sum only contains finite nonzero terms. We have $V_{j-1} = V_j \oplus W_j$, this means that $f_{|\ V_{j-1}}$ can also be represented by the expansion

$$\begin{aligned} f_{|\ V_j \oplus W_j}(x) &= \sum_{i=0}^{N} a_{j\,i}\phi_{j\,i}(x) + \sum_{i=0}^{N-1} b_{j\,i}\psi_{j\,i}(x) \\ &= \begin{bmatrix} a \\ b \end{bmatrix}[\Phi\ \Psi] \end{aligned} \tag{10}$$

where $a^T = [a_{j0},\, a_{j1},\, \ldots,\, a_{j\,N}]$, $b^T = [b_{j0},\, b_{j1}, \ldots, b_{j\,N-1}]$, $b_{j\,i} = \langle f, \widetilde{\psi}_{j\ i}\rangle$, $a_{ji} = \langle f, \widetilde{\phi}_{j\,i}\rangle$, $\Phi = [\phi_{j0},\, \phi_{j1},\, \ldots,\, \phi_{j\,N}]$ and $\Psi = [\psi_{j0},\, \psi_{j1},\, \ldots,\, \psi_{j\,N-1}]$.

The vector $F = \begin{bmatrix} a \\ b \end{bmatrix}$ is called "the vector form of f in $V_j \oplus W_j$". Let, Λ be the subdomainn of Ω then some member of the vector F is assigned to Λ. We represent restricted F to subdomain Λ by $F_\Lambda = \begin{bmatrix} a_\Lambda \\ b_\Lambda \end{bmatrix}$.

Definition 2: (the Fast Wavelet Transform) FWT converts the scaling function coefficients in space V_j to scaling function and wavelet function coefficients in space $V_j \oplus W_j$. From (7) and (8) and biorthogonal property of wavelets we have:

$$a_{j,k} = \sum_i \tilde{h}_i a_{j-1,2k+i} \text{and} b_{j,k} = \sum_i \tilde{g}_i a_{j-1,2k+i}. \tag{11}$$

The fast wavelet transform embodied by these two equations, and indeed FWT maps $\overrightarrow{a_{j-1}}$ onto $\overrightarrow{a_j}$ and $\overrightarrow{b_j}$, $FWT\left[\overrightarrow{a_{j-1}}\right] = \begin{bmatrix} \overrightarrow{a_j} \\ \overrightarrow{b_j} \end{bmatrix}$.

The inverse fast wavelet transform recursively uses the formula

$$a_{j-1,k} = \sum_n h_{k-2n}\, a_{j,n} + \sum_l g_{k-2l}\, b_{j,l}\,. \tag{12}$$

Equation (12) is constructed from (9) and (10) biorthogonal property of wavelets. We obtain IFWT by using (12) as $IFWT\begin{bmatrix} \overrightarrow{a_j} \\ \overrightarrow{b_j} \end{bmatrix} = \left[\overrightarrow{a_{j-1}}\right]$.

III. Matrix Approximations

In the following we need operational matrix M^α to approximate D_x^α on V_j where $0 \le \alpha \le 2$. Producing M^α takes a number of steps, starting with the construction of the matrix P_j which is a square matrix that converts the vector form of f in V_j into the actual values of f at some points:

$$P_j F = \overline{F} \tag{13}$$

where $P_j = \left[\phi_{j\,k}(x_i)\right]_{k,\,i}$, $F = [a_{j\,0},\, a_{j\,1},\, \ldots,\, a_{j\,N}]^T$, $\overline{F} = [\phi_{j\,0}(x_0),\, \phi_{j\,1}(x_1),\, \ldots,\, \phi_{j\,N}(x_N)]$, $f(x) = \sum\limits_{k=0}^{N} a_{jk}\phi_{jk}(x)$, $x_i = i\,2^j$, for $0 \le i \le N = m2^{-j}$.

Next, we construct the matrix P_j^α which converts the vector form of $D_x^\alpha f$ in V_j into the actual values of $D_x^\alpha f$ at some points:

$$P_j^\alpha F = \overline{F}^\alpha \tag{14}$$

where $P_j^\alpha = [D_x^\alpha \phi_{j\,k}(x_i)]_{k,i}$, $\overline{F}^\alpha = [D_x^\alpha \phi_{j\,0}(x_0),\, D_x^\alpha \phi_{j\,1}(x_1), \ldots,\, D_x^\alpha \phi_{j\,N}(x_N)]$, $x_i = i\,2^j$, for $0 \le i \le N = m2^{-j}$.

$$M^\alpha = FWT \times (P_{j-1})^{-1} \times P_j^\alpha \times IFWT. \tag{15}$$

In the below, we illustrate the function of the fractional derivative matrix.

$$\begin{array}{lll}
V_j \oplus W_j : \sum\limits_k a_{jk}\phi_{jk} + \sum\limits_k d_{jk}\psi_{jk}, & & \sum\limits_k b_{j-1k}\phi_{jk} + \sum\limits_k c_{j-1k}\psi_{jk} \\
\quad IFWT \downarrow & & \quad FWT \uparrow \\
V_{j-1} : \sum\limits_k a_{j-1,k}\phi_{j-1,k} & & \sum\limits_k b_{j-1,k}\phi_{j-1,k} \\
\quad P_j^\alpha \searrow & \sum\limits_k a_{j-1,k} D_x^\alpha \phi_{j-1,k}(x_i) & \nearrow (P_j)^{-1}
\end{array}$$

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

$$FWT \times (P_j)^{-1} \times G \times P_j \times IFWT$$

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

IV. Boundary Conditions

The basic idea for constructing the boundary wavelets and boundary scaling functions can be described as follows, for the case where $\Omega = [0, m]$. Firstly we take all wavelets and scaling functions that are located near the boundary, we are thus constructing the specific linear combinations of these functions that satisfy the fixed non zero boundary conditions (I). We separate the function u into two functions.

$$u(x,t) = v(x,t) + B(x) \tag{16}$$

where the function v satisfies the zero boundary conditions

$$\begin{cases} v(x,0) = g(x) - B(x) \\ c_1 v(0,t) + c_2 v_x(0,t) = 0 \\ c_4 v(n,t) + c_5 v_x(n,t) = 0 \end{cases}$$

and

$$\begin{cases} c_1 B(0) + c_2 B'(0) = c_3 \\ c_4 B(n) + c_5 B'(n) = c_6. \end{cases}$$

Now, we only reshape wavelets at $x = 0$, the construction of other scaling functions and wavelets at the boundary is nearly same as this. We let ψ denote the combination of the wavelets in V_j which their support contains $x = 0$.

$$\psi(x) = a\psi_{j,-1}(x) + b\psi_{j,0}(x) + c\psi_{j,+1}(x). \tag{17}$$

Here ψ must satisfy the boundary conditions

$$c_1\psi(0) + c_2\psi'(0) = 0. \tag{18}$$

On the other hand we need to represent $\sum\limits_{k=1}^{S} f_k(x) D_x^{\alpha_k}\psi(x) D_x^{\beta_k}\psi(x)$ with the reshaped wavelet $\psi(x)$ which satisfies the boundary condition so we must have

$$\begin{aligned} & c_1 \sum_{k=1}^{S} f_k(0) D_x^{\alpha_k}\psi(0) D_x^{\beta_k}\psi(0) \\ & \quad + c_2 \frac{\partial}{\partial x}_{|x=0} \sum_{k=1}^{S} f_k(x) D_x^{\alpha_k}\psi(x) D_x^{\beta_k}\psi(x) = 0. \end{aligned} \tag{19}$$

So a nonlinear system is obtained from (18) and (19), and we get the coefficients a, b and d from solving this system.

V. The Proposed Method

We consider the fractional Adams method for solving FPDE (1). This method was first studied by Diethelm, Ford and Freed [38]. Their method for solving (20) is as follows:

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

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

where

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

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

Also we can show $0 < c_{k,n+1} \le 2$ for all $0 \le k \le n+1$ and $0 < \beta < 1$.

The predictor y_{n+1}^P is determined by

$$y_{n+1}^P = y_0 + \frac{h^\beta}{\Gamma(\beta+1)} \sum_{k=0}^{n} b_{k,\,n+1} f(t_k\,, y_k)$$

where $b_{k,\,n+1} = (n+1-k)^\beta - (n-k)^\beta$.

So the approximate solution for the time space-fractional (1) by using the fractional Adam's method would be

$$\begin{aligned} u(x, t_{n+1}) = \; & u(x, t_0) \\ & + \frac{h^\beta}{\Gamma(\beta+2)} \sum_{k=0}^{n} c_{k,n+1} (Nu(x, t_k)) \\ & + \frac{h^\beta}{\Gamma(\beta+2)} Nu^P(x, t_{n+1}) \end{aligned} \tag{22}$$

where $N(u(x,t)) = \sum\limits_{i=1}^{S} f_i(x) D_x^{\alpha_i} u(x,t) D_x^{\beta_i} u(x,t)$ and $u^P(x, t_{n+1}) = u(x, t_0) + \frac{h^\beta}{\Gamma(\beta+1)} \sum\limits_{k=0}^{n} b_{k,\,n+1} N(u(x, t_k))$.

Now we can take the space V_{j-1} to approximate the solution of (22). If we consider $\begin{bmatrix} a^k \\ b^k \end{bmatrix}$ as a vector form of $u(x, t_k)$

in V_{j-1}, then from (22) and the definition of M^{α} from (15) we have

$$\begin{aligned}\begin{bmatrix} a^{n+1} \\ b^{n+1} \end{bmatrix} &= \begin{bmatrix} a^{0} \\ b^{0} \end{bmatrix} \\ &+ \frac{h^{\beta}}{\Gamma(\beta+2)}\Bigg\{\sum_{k=0}^{n} c_{k,n+1}\sum_{r=1}^{S} F_r \\ &\times \operatorname{diag}\left(M^{\alpha_r}\begin{bmatrix} a^{k} \\ b^{k} \end{bmatrix}\right) M^{\beta_r}\begin{bmatrix} a^{k} \\ b^{k} \end{bmatrix} \\ &+ \sum_{r=1}^{S} F_r \operatorname{diag}\left(M^{\alpha_r}\begin{bmatrix} a^{P} \\ b^{P} \end{bmatrix}\right) \\ &\times M^{\beta_r}\begin{bmatrix} a^{n+1} \\ b^{n+1} \end{bmatrix}\Bigg\}\end{aligned} \tag{23}$$

where

$$\begin{aligned}\begin{bmatrix} a^{P} \\ b^{P} \end{bmatrix} &= \begin{bmatrix} a^{0} \\ b^{0} \end{bmatrix} + \frac{h^{\beta}}{\Gamma(\beta+1)}\sum_{k=0}^{n} b_{k,n+1}\sum_{r=1}^{S} F_r \\ &\times \operatorname{diag}\left(M^{\alpha_r}\begin{bmatrix} a^{k} \\ b^{k} \end{bmatrix}\right) M^{\beta_r}\begin{bmatrix} a^{k} \\ b^{k} \end{bmatrix}\end{aligned} \tag{24}$$

also we can write as a system:

$$\left(I - \frac{h^{\beta}}{\Gamma(\beta+2)}M\right)\begin{bmatrix} a^{n+1} \\ b^{n+1} \end{bmatrix} = \begin{bmatrix} \overline{a_n} \\ \overline{b_n} \end{bmatrix} \tag{25}$$

where

$$\begin{aligned} M &= \sum_{r=1}^{S} F_r \operatorname{diag}\left(M^{\alpha_r}\begin{bmatrix} a^{P} \\ b^{P} \end{bmatrix}\right) M^{\beta_r}, \\ \begin{bmatrix} \overline{a_n} \\ \overline{b_n} \end{bmatrix} &= \begin{bmatrix} a^{0} \\ b^{0} \end{bmatrix} + \frac{h^{\beta}}{\Gamma(\beta+2)}\sum_{k=0}^{n} c_{k,n+1}\sum_{r=1}^{S} F_r \\ &\times \operatorname{diag}\left(M^{\alpha_r}\begin{bmatrix} a^{k} \\ b^{k} \end{bmatrix}\right) M^{\beta_r}\begin{bmatrix} a^{k} \\ b^{k} \end{bmatrix}.\end{aligned} \tag{26}$$

Solving system in the Finer space V_{j-1} produces more accurate solution, But if the system becomes larger then calculations are also increased. To increase the accuracy of solution in some places of domain Ω and to avoid increase of our calculations we use the multiscaling method. This means that once we solve the system in a space V_j and domain Ω that we call large scale system. Once again we solve the system in a Finer space V_{j-1} and subdomain Λ that we call small scale system. Combination of these two systems provides suitable accuracy and less calculations than the solutions of the system achieved in the space V_{j-1} on domain Ω. If we want to get a more accurate solution in a subdomain Λ, we need to do the following process (we can do this process for several subdomains). 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 The elements of M will have to be broken up, into a block decomposition of the form

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}.$$

We only consider the block A which operates as operator N in the space V_j over all domain Ω. Then using time stepping scheme (22) we find an approximation for a^{n+1} which we will call a_{Λ}^{Tm}

$$\left(I - \frac{h^{\beta}}{\Gamma(\beta+2)}A\right) a^{Tm} = \overline{a_n}. \tag{27}$$

The small-scale system is expressed in terms of a matrix M_{Λ} of the form

$$M_{\Lambda} = \begin{bmatrix} A_{\Lambda} & B_{\Lambda} \\ C_{\Lambda} & D_{\Lambda} \end{bmatrix}$$

where A_{Λ}, B_{Λ}, C_{Λ} and D_{Λ}, are composed of the elements from A, B, C and D that are related to Λ. We have the time step solved at the large-scale resolution V_j on all Ω. What we want now is to solve the system on Λ at the small scale resolution V_{j-1}. However, a^n and the newly calculated a_{Λ}^{Tm} have to be included. So, we take the components of $\overline{a_n}$ and a_{Λ}^{Tm} that are in Λ, a_{Λ}^{n} and a_{Λ}^{Tm}. These are used in the system. Next, we are looking for vector correction a^{Cr} where $a^{n+1} = a^{Tm} + a^{Cr}$. Now, what we want is to solve the system on Λ at the small scale resolution V_{j-1}.

Consider the fractional Adam's method for this case

$$\left(I - \frac{h^{\beta}}{\Gamma(\beta+2)}\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right)_{\Lambda}\begin{bmatrix} a^{n+1} \\ b^{n+1} \end{bmatrix}_{\Lambda} = \begin{bmatrix} \overline{a_n} \\ \overline{b_n} \end{bmatrix}_{\Lambda}. \tag{28}$$

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

$$\begin{aligned}\left(I - \tfrac{h^{\beta}}{\Gamma(\beta+2)}\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right)_{\Lambda}\left(\begin{bmatrix} a^{Cr} \\ b^{n+1} \end{bmatrix}_{\Lambda} + \begin{bmatrix} a^{Tm} \\ 0 \end{bmatrix}_{\Lambda}\right) \\ = \begin{bmatrix} \overline{a_n} \\ \overline{b_n} \end{bmatrix}_{\Lambda}\end{aligned} \tag{29}$$

then

$$\begin{aligned}\left(I - \frac{h^{\beta}}{\Gamma(\beta+2)}\begin{bmatrix} A & B \\ C & D \end{bmatrix}\right)_{\Lambda}\begin{bmatrix} a^{C_r} \\ b^{n+1} \end{bmatrix}_{\Lambda} \\ = \begin{bmatrix} 0 \\ \widetilde{b_n} + \frac{h^{\beta}}{\Gamma(\beta+2)} C\, a^{Tm} \end{bmatrix}_{\Lambda}.\end{aligned} \tag{30}$$

We get $\begin{bmatrix} a^{n+1} \\ b^{n+1} \end{bmatrix}_{\Lambda}$ by solving the above system.

The last step is to construct the vector $\begin{bmatrix} a^{n+1} \\ b^{n+1} \end{bmatrix}_{\Omega}$ from vectors $\begin{bmatrix} a^{Tm} \\ 0 \end{bmatrix}_{\Omega}$ and $\begin{bmatrix} a^{n+1} \\ b^{n+1} \end{bmatrix}_{\Lambda}$. In the subdomain Λ the vector $\begin{bmatrix} a^{n+1} \\ b^{n+1} \end{bmatrix}_{\Lambda}$ is a better approximate solution than the vector $\begin{bmatrix} a^{Tm} \\ 0 \end{bmatrix}_{\Lambda}$ for the system (25). So to increase the accuracy of the approximated vector $\begin{bmatrix} a^{Tm} \\ 0 \end{bmatrix}_{\Omega}$ we must replace 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.

$$\begin{bmatrix} a^{Tm} \\ \\ \text{———} \\ \\ 0 \end{bmatrix} = \begin{bmatrix} a_1^{Tm} \\ \vdots \\ a_\Lambda^{Tm} \\ \vdots \\ a_N^{Tm} \\ \text{———} \\ 0 \\ \vdots \\ 0_\Lambda \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \longleftrightarrow \\ and \\ \longleftrightarrow \end{matrix} \begin{bmatrix} a_\Lambda^{n+1} \\ \\ \\ b_\Lambda^{n+1} \end{bmatrix}$$

$$\begin{bmatrix} a_\Lambda^{n+1} \\ \\ \\ b_\Lambda^{n+1} \end{bmatrix} \overset{REPLACE}{\longrightarrow} \begin{bmatrix} a_1^{Tm} \\ \vdots \\ a_\Lambda^{Tm} \\ \vdots \\ a_N^{Tm} \\ \text{———} \\ 0 \\ \vdots \\ 0_\Lambda \\ \vdots \\ 0 \end{bmatrix} \equiv \begin{bmatrix} a^{n+1} \\ \\ \text{———} \\ \\ b^{n+1} \end{bmatrix}.$$

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 $[a_r, \ldots, a_{s+1}]^T$.

Algorithm 1 Proposed Method

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

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

Step 3: Constructing matrix M by using (17).

Step 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, D.

Step 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.

Step 6: for $n = 0$ to k.

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

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

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

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

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

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

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

Step 14: end for.

VI. Stability

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

Lemma 1: (Discrete Gronwall Lemma) If $\{y_k\}$, $\{f_k\}$ and $\{g_k\}$ are nonnegative sequences and

$$y_n \le f_n + \sum_{0 \le k \le n} g_k y_k\,, \quad \text{for } n \ge 0 \tag{31}$$

then

$$y_n \le f_n + \sum_{0 \le k \le n} f_k g_k \exp\left(\sum_{k \le i \le n} g_i\right), \quad for\ n \ge 0. \tag{32}$$

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

$$y_n \le f_n \exp\left(\sum_{0 \le i \le n} g_i\right), \quad for\ n \ge 0. \tag{33}$$

Theorem 1: The approximation scheme (23) is stable.

Proof: Let $u_n = [u\,(x_0, t_n)\,, u\,(x_1, t_n)\,, \ldots, u\,(x_{2N}, t_n)]$ denote the exact solution and $\widetilde{U}_n$ denote approximate solution of it. From approximation scheme (23) and $\widetilde{U}_n = P_j \times IFWT \begin{bmatrix} a^n \\ b^n \end{bmatrix}$ we have

$$\begin{aligned} \widetilde{U}_{n+1} = {} & \widetilde{U}_0 \\ & + \frac{h^\beta}{\Gamma(\beta+2)} \Bigg\{ \sum_{k=0}^{n} c_{k,n+1} \sum_{r=1}^{S} F_r M^{\alpha_r} \\ & + \operatorname{diag}\left(FWT \times P_j^{-1} \times \widetilde{U}_k\right) M^{\beta_r} \\ & \times FWT \times P_j^{-1} \times \widetilde{U}_k + \sum_{r=1}^{S} F_r M^{\alpha_r} \\ & \times \operatorname{diag}\left(FWT \times P_j^{-1} \times \widetilde{U}_{n+1}^P\right) \\ & \times M^{\beta_r} FWT \times P_j^{-1} \times \widetilde{U}_{n+1} \Bigg\} \end{aligned} \tag{34}$$

where

$$\begin{aligned} \widetilde{U}_{n+1}^P = {} & \widetilde{U}_0^P \\ & + \tfrac{h^\beta}{\Gamma(\beta+1)} \left\{ \sum_{k=0}^{n} b_{k,n+1} \sum_{r=1}^{S} F_r M^{\alpha_r} \right. \\ & \times \operatorname{diag}\left(FWT \times P_j^{-1} \times \widetilde{U}_k\right) \\ & \times M^{\beta_r} FWT \times P_j^{-1} \times \widetilde{U}_k \end{aligned}$$

and $b_{k,\,n+1} = (n+1-k)^\beta - (n-k)^\beta$.

There is no loss of generality in assuming that u_n and its approximation are bounded in the domain $\left\|\widetilde{U}_n\right\| \leq M^*$. Also we can choose h enough small that makes

$$\begin{aligned}&\tfrac{h^\beta}{\Gamma(\beta+2)} \sum_{r=1}^{S} \|F_r\| \|M^{\alpha_r}\| \|FWT\| \left\|P_j^{-1}\right\| M^* \left\|M^{\beta_r}\right\| \\ &\|FWT\| \left\|P_j^{-1}\right\| \leq \tfrac{1}{2} < 1\end{aligned} \tag{35}$$

this guarantees nonsingularity of the matrix

$$\begin{aligned}I - &\frac{h^\beta}{\Gamma(\beta+2)} \sum_{r=1}^{S} F_r \times M^{\alpha_r} \\ &\times \operatorname{diag}\left(FWT \times P_j^{-1} \times \widetilde{U}_{n+1}^P\right) \times M^{\beta_r} \\ &\times FWT \times P_j^{-1}.\end{aligned} \tag{36}$$

Then we have

$$\begin{aligned}\widetilde{U}_{n+1} = &\Bigg(I - \frac{h^\beta}{\Gamma(\beta+2)} \sum_{r=1}^{S} F_r \times M^{\alpha_r} \\ &\times \operatorname{diag}\left(FWT \times P_j^{-1} \times \widetilde{U}_{n+1}^P\right) \\ &\times M^{\beta_r} \times FWT \times P_j^{-1}\Bigg)^{-1} \widetilde{U}_0 \\ &+ \frac{h^\beta}{\Gamma(\beta+2)} \Bigg(I - \frac{h^\beta}{\Gamma(\beta+2)} \sum_{r=1}^{S} F_r \times M^{\alpha_r} \\ &\times \operatorname{diag}\left(FWT \times P_j^{-1} \times \widetilde{U}_{n+1}^P\right) \times M^{\beta_r} \\ &\times FWT \times P_j^{-1}\Bigg)^{-1} \\ &\times \Bigg\{\sum_{k=0}^{n} c_{k,n+1} \sum_{r=1}^{S} F_r \times M^{\alpha_r} \\ &\times \operatorname{diag}\left(FWT \times P_j^{-1} \times \widetilde{U}_k\right) \times M^{\beta_r} \\ &\times FWT \times P_j^{-1} \widetilde{U}_k\Bigg\}.\end{aligned} \tag{37}$$

Next using the fact that

$$\begin{aligned}&\Bigg\|\Bigg(I - \tfrac{h^\beta}{\Gamma(\beta+2)} \sum_{r=1}^{S} F_r \times M^{\alpha_r} \\ &\times \operatorname{diag}\left(FWT \times P_j^{-1} \times \widetilde{U}_{n+1}^P\right) \\ &\times M^{\beta_r} \times FWT \times P_j^{-1}\Bigg)^{-1}\Bigg\| \\ &\leq \frac{1}{1 - \left\| \begin{array}{l} \frac{h^\beta}{\Gamma(\beta+2)} \sum_{r=1}^{S} F_r \times M^{\alpha_r} \\ \times\, diag\left(FWT \times P_j^{-1} \times \widetilde{U}_{n+1}^P\right) \\ \times M^{\beta_r} \times FWT \times P_j^{-1} \end{array} \right\|} \leq 2.\end{aligned} \tag{38}$$

Therefore, equation (37) yields

$$\left\|\widetilde{U}_{n+1}\right\| \leq 2\left\|\widetilde{U}_0\right\| + \frac{\left(\frac{T}{N}\right)^\beta}{\Gamma(\beta+2)} \sum_{k=0}^{n} c_{k,n+1} \left\|\widetilde{U}_k\right\|. \tag{39}$$

By applying Gronwall's inequality, we obtain

$$\left\|\widetilde{U}_{n+1}\right\| \leq 2\left\|\widetilde{U}_0\right\| \exp\left(\frac{T^\beta}{\Gamma(\beta+2)} \sum_{k=0}^{n} \frac{c_{k,n+1}}{N^\beta}\right) \tag{40}$$

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

$$\left\|\widetilde{U}_{n+1}\right\| \leq 2\left\|\widetilde{U}_0\right\| \exp\left(\frac{2T^\beta}{\Gamma(\beta+2)}\right) \tag{41}$$

this completes the proof of stability. ■

VII. Numerical Example

In this section we implement the presented method to solve two examples, and also we compare the exact solution with the approximate solution.

Example 1: We consider the following nonlinear time space fractional Burger equation:

$$D_t^\alpha u = -u D_x^\beta u + \nu u_{xx}$$

with the initial condition and boundary condition

$$\begin{cases} u(x,0) = \coth(5x-10), & 0 < x < 4 \\ u(0,t) = u(4,t) = 0, & 0 \leq t \leq 1. \end{cases}$$

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

Example 2: We consider the following fractional nonlinear Klein-Gordon differential equation:

$$D_t^\beta u = u_{xx} - u + u^3$$

with initial condition and boundary condition are as follows:

$$\begin{cases} u(x,0) = \cos(x), & 1 < x < 7 \\ u(1,t) = u(7,t) = 0, & 0 \leq t \leq 1. \end{cases}$$

The example was solved by the presented multi-scale method with V_{-4} on Ω and V_{-5} on Λ. The Table III shows that the accuracy can be improved by enlarging subdomain.

TABLE I 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 $t = 0.5$

(a) error $\alpha = 0.9,\ \beta = 0.8$				(b) error $\alpha = 0.7,\ \beta = 0.8$			
x	V_{-3}/V_{-4}	V_{-4}/V_{-5}	V_{-5}/V_{-6}	x	V_{-3}/V_{-4}	V_{-4}/V_{-5}	V_{-5}/V_{-6}
1	4.86851E-5	1.20725E-5	3.01086E-6	1	3.87951E-5	9.5696E-6	2.38312E-6
1.5	-6.27884E-5	-1.78008E-5	-4.54069E-6	1.5	-3.15994E-5	-8.706E-6	-2.21822E-6
2	-0.00048152	-0.000117335	-2.86878E-5	2	-7.38939E-5	-2.2020E-5	-5.64321E-6
2.5	0.003519481	0.000616091	0.000135349	2.5	-0.000228084	-9.5180E-5	-2.53185E-5
3	0.000154392	4.97541E-5	1.25426E-5	3	0.004762792	-0.0008628	-0.00039248
3.5	7.77682E-6	1.98912E-6	4.99819E-7	3.5	-3.36494E-5	1.52567E-5	4.19946E-6

TABLE II 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 $t = 0.5$

(a) error $\alpha = 0.9,\ \beta = 0.8$				(b) error $\alpha = 0.7,\ \beta = 0.8$			
x	V_{-6}/V_{-7}	V_{-7}/V_{-8}	$solution on V_{-8}$	x	V_{-6}/V_{-7}	V_{-7}/V_{-8}	$solution on V_{-8}$
1	7.52109E-7	1.87972E-7	0.01563038	1	5.95109E-7	1.48723E-7	0.017861225
1.5	-1.13743E-6	-2.84176E-7	0.126210355	1.5	-5.56031E-7	-1.38975E-7	0.08964552
2	-7.09146E-6	-1.76352E-6	0.428300053	2	-1.41355E-6	-3.52853E-7	0.225663084
2.5	3.2228E-5	7.92191E-6	0.471766274	2.5	-6.38202E-6	-1.59423E-6	0.388315038
3	3.13962E-6	7.8496E-7	0.020295168	3	-0.00011126	-2.87553E-5	0.250095232
3.5	1.25102E-7	3.12235E-8	0.001442839	3.5	1.07135E-6	2.69138E-7	0.002252919

TABLE III The Errors are the Difference Between the Multi-Scale Results and the Results Obtained Using V_{-5}, with $h = 0.01$ at $t = 0.5$

x	$error \Lambda = [3\ 5]$	$error \Lambda = [4\ 6]$	$solution\ on V_{-5}$
1.25	-0.000132	-7.99128E-5	-0.022484
2.25	-0.0003292	**-2.97362E-6**	-0.0542849
3.25	**1.60129E-6**	**1.22943E-6**	0.0224848
4.25	**3.87375E-6**	**2.97362E-6**	0.0542849
5.25	-0.0001361	**-1.22943E-6**	-0.0224848
6.25	-0.00032	-0.00019	-0.05428

The points in the subdomain Λ are dislayed by bold font

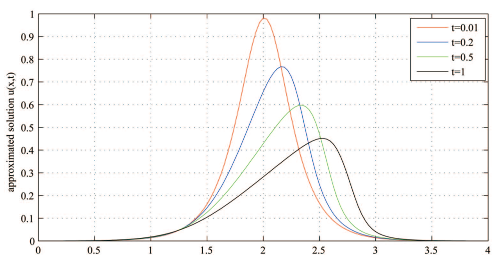

Fig. 1. This figure is the approximated solution of the presented method in different times with $\alpha = 0.9$, $h = 0.01$, $\beta = 0.8$.

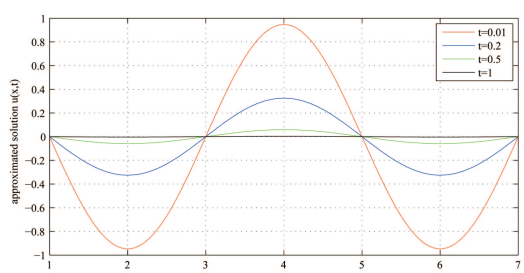

Fig. 2. This figure is the approximated solution of the presented method in different times with $\alpha = 0.9$, $h = 0.01$.

VIII. Results

In this work a practical approach for solving nonlinear 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, also wavelets are employed in such a way that satisfy the boundary conditions exactly.

References

[1] Li X X. Numerical solution of fractional differential equations using cubic B-spline wavelet collocation method. *Communications in Nonlinear Science and Numerical Simulation*, 2012, **17**(10): 3934−3946

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

[3] Podlubny I. *Fractional Differential Equations*. San Diego, CA: Academic Press, 1999.

[4] Li X K, Han X H, Wang X P. Numerical modeling of viscoelastic flows using equal low-order finite elements. *Computer Methods in Applied Mechanics and Engineering*, 2010, **199**(9−12): 570−581

[5] Gorenflo R, Mainardi F, Scalas E, Raberto M. Fractional calculus and continuous-time finance. III: the diffusion limit. *Mathematical Finance*. Konstanz, Germany: Birkhäuser, 2001. 171−180

[6] Raberto M, Scalas E, Mainardi F. Waiting-times and returns in high-frequency financial data: an empirical study. *Physica A: Statistical Mechanics and its Applications*, 2002, **314**(1−4): 749−755

[7] Sabatelli L, Keating S, Dudley J, Richmond P. Waiting time distributions in financial markets. *The European Physical Journal B - Condensed Matter and Complex Systems*, 2002, **27**(2): 273−275

[8] Baeumer B, Benson D A, Meerschaert M M, Wheatcraft S W. Subordinated advection-dispersion equation for contaminant transport. *Water Resources Research*, 2001, **37**(6): 1543−1550

[9] Benson D A, Schumer R, Meerschaert M M, Wheatcraft S W. Fractional dispersion, Lévy motions, and the MADE tracer tests. *Transport in Porous Media*, 2001, **42**(1): 211−240

[10] Benson D A, Wheatcraft S W, Meerschaert M M. Application of a fractional advection-dispersion equation. *Water Resources Research*, 2000, **36**(6): 1403−1412

[11] Benson D A, Wheatcraft S W, Meerschaert M M. The fractional-order governing equation of Lévy motion. *Water Resources Research*, 2000, **36**(6): 1413−1424

[12] Schumer R, Benson D A, Meerschaert M M, Baeumer B. Multiscaling fractional advection-dispersion equations and their solutions. *Water Resources Research*, 2003, **39**(1): 1022

[13] Tadjeran C, Benson D, Meerschaert M M. Fractional radial flow and its application to field data. *Water Resources Research*, to be published

[14] Machado J A T. Discrete-time fractional-order controllers. *Fractional Calculus & Applied Analysis*, 2001, **4**(1): 47−66

[15] Fix G J, Roof J P. Least squares finite element solution of a fractional order two-point boundary value problem. *Computers & Mathematics with Applications*, 2004, **48**(7−8): 1017−1033

[16] Shen S, Liu F, Anh V, Turner I. The fundamental solution and numerical solution of the Riesz fractional advection-dispersion equation. *IMA Journal of Applied Mathematics*, 2008, **73**(6): 850−872

[17] Meerschaert M M, Tadjeran C. Finite difference approximations for fractional advection dispersion flow equations. *Journal of Computational and Applied Mathematics*, 2004, **172**(1): 65−77

[18] Richtmyer R D, Morton K W. *Difference Methods for Initial-Value Problems* (Second edition). Malabar, Florida: Krieger Publishing, 1994.

[19] Li C P, Zeng F H. *Numerical Methods for Fractional Calculus*. New York: Chapman and Hall, 2015.

[20] Guo B L, Pu X K, Huang F H. *Fractional Partial Differential Equations and their Numerical Solutions*. Singapore: World Scientific Publishing Co., 2015.

[21] Luchko Y F, Srivastava H M. The exact solution of certain differential equations of fractional order by using operational calculus. *Computers and Mathematics with Applications*, 1995, **29**(8): 73−85

[22] Samko S G, Kilbas A A, Marichev O I. *Fractional Integrals and Derivatives: Theory and Applications*. Switzerland: Gordon and Breach Science Publishers, 1993.

[23] Kilbas A A, Srivastava H M, Trujillo J J. *Theory and Applications of Fractional Differential Equations*. Amsterdam: Elsevier, 2006.

[24] Odibat Z M, Momani S. Application of variational iteration method to nonlinear differential equations of fractional order. *International Journal of Nonlinear Sciences and Numerical Simulation*, 2006, **7**(1): 27−34

[25] Momani S, Odibat Z. Numerical comparison of methods for solving linear differential equations of fractional order. *Chaos, Solitons & Fractals*, 2007, **31**(5): 1248−1255

[26] Momani S. Non-perturbative analytical solutions of the space- and time-fractional Burgers equations. *Chaos, Solitons & Fractals*, 2006, **28**(4): 930−937

[27] Dahmen W, Kurdila A, Oswald P. *Multiscale Wavelet Methods for Partial Differential Equations, Volume 6*. San Diego: Academic Press, 1997.

[28] McLaren D A. Sequential and Localized Implicit Wavelet based Solvers for Stiff Partial Differential Equations [Ph. D. dissertation], University of Ottawa, Ottawa, ON, 2012

[29] Hesthaven J S, Jameson L M. A wavelet optimized adaptive multi-domain method. *Journal of Computational Physics*, 1998, **145**(1): 280−296

[30] Mehra M, Kevlahan N K R. An adaptive multilevel wavelet solver for elliptic equations on an optimal spherical geodesic grid. *SIAM Journal on Scientific Computing*, 2008, **30**(6): 3073−3086

[31] Müller S, Stiriba Y. A multilevel finite volume method with multiscale-based grid adaptation for steady compressible flows. *Journal of Computational and Applied Mathematics*, 2009, **277**(2): 223−233

[32] Vasilyev O V, Paolucci S. A dynamically adaptive multilevel wavelet collocation method for solving partial differential equations in a finite domain. *Journal of Computational Physics*, 1996, **125**(2): 498−512

[33] Aminikhah H, Tahmasebi M, Roozbahani M M. Numerical solution for the time space-fractional partial differential equation by using wavelet multi-scale method. *U. P. B Scientific Bulletin-Series A-Applied Mathematics and Physics*, to be published

[34] Blatter C. *Wavelets: A Primer*. Naticks, Massachusetts: A K Peters/CRC Press, 2002.

[35] Goswami J C, Chan A K. *Fundamentals of Wavelets: Theory, Algorithms, and Applications*. New York: John Wiley and Sons, 1999.

[36] Mallat S. *A Wavelet Tour of Signal Processing* (Second edition). New York: Academic Press, 1999.

[37] Sweldens W. The Construction and Application of Wavelets in Numerical Analysis [Ph. D. dissertation], KU Leuven, Leuven, 1995

[38] Diethelm K, Ford N J, Freed A D. Detailed error analysis for a fractional Adams method. *Numerical Algorithms*, 2004, **36**(1): 31−52

Hossein Aminikhah was born in Iran, in 1979. He received a Ph. D. degree in Applied Mathematics (Numerical Analysis) from the University of Guilan, Rasht, Iran, in 2008, where he is currently Associate Professor in the Department of Applied Mathematics. His research interests include numerical methods for functional differential equations and numerical linear algebra.

Mahdieh Tahmasebi was born in Tehran, Iran, in 1980. She received her B. Sc. in Applied Mathematics from Shahid Beheshti University. She finished her M. Sc. and Ph. D. in Sharif University of Technology in Pure Mathematics. She is a member of Applied Mathematics Department of Tarbiat Modares University as an assistant professor from 2012 to now. Her research interests includes stochastic differential equation, Malliavin calculus, stochastic modelling, stochastic control theory, mathematical finance and numerical solution of SDEs.

Mahmoud Mohammadi Roozbahani is now a Ph. D. degree student in the Applied Mathematics Department of the University of Guilan. His interests and research areas include numerical analysis.