Available online at http://scik.org
J. Math. Comput. Sci. 11 (2021), No. 5, 5327-5343
https://doi.org/10.28919/jmcs/6048
ISSN: 1927-5307

NEW TECHNIQUE FOR SOLVING TIME FRACTIONAL WAVE EQUATION: PYTHON

KRISHNA GHODE[1,*], KALYANRAO TAKALE[2], SHRIKISAN GAIKWAD[3]

[1]Department of Mathematics, B. K. Birla College of Arts, Science and Commerce, Kalyan, India
[2]Department of Mathematics, RNC Arts, JDB Commerce and NSC Science College, Nashik Road, Nashik, India
[3]Department of Mathematics, New Arts, Commerce and Science College, Ahmednagar, India

Abstract. In this paper, we develop the fractional order explicit finite difference scheme for time fractional wave equation. Furthermore, we prove the scheme is conditionally stable and convergent. As an application of the scheme numerical solution of the test problem is obtained by Python Programme and represented graphically.

Keywords: fractional wave equation; Caputo derivative; finite difference scheme; stability and Python.
2010 AMS Subject Classification: 35R11, 65M06, 35L05, 26A33.

1. Introduction

Fractional order partial differential equations are widely used in many areas like physics, engineering, finance, medical sciences etc.[2, 8], as they can provide an adequate and precise description of the models, which cannot be revealed by integer-order differential equations. Recently, the fractional wave equation is occurs in many studies such as acoustics, electromagnetic, seismic study etc[3, 10, 18]. It also describes the movement of strings, wires and fluid surfaces[12]. Any wave or motion can be express in terms of a sum of sine or sinusoidal waves. The traveling wave solution of the wave equation was first published by d'Alembert in 1747[7].

*Corresponding author
E-mail address: ghodekrishna@gmail.com
Received May 16, 2021

The traveling wave solutions of fractional order partial differential equations are deeply helpful to realise the mechanisms of the phenomena as well as their further application in real-world. Finite difference method is also one of the very effective method for solving fractional order partial differential equations [1, 4, 14, 15, 16, 17].

Recently, several authors have developed different numerical techniques for solving differential equations using mathematical software[5, 6, 11]. Python is a high-level interpreted programming language that has a vast standard library and a lot of modern software engineering tools. It allows fast exploration of various ideas as well as efficient implementation. It has been used for teaching and research in various branches of pure and applied mathematics[19]. Therefore, in this connection we develop the explicit finite difference scheme for time-fractional wave equation and obtain its solution using Python programme.

We organize the paper as follows: In section 2, we develop the explicit finite difference scheme for time fractional wave equation. The section 3 is devoted for stability of solution of the scheme and the convergence is proven in section 4. In section 5, we develop a python programme for the proposed scheme and numerical experiments are performed to confirm stability and convergence of the scheme.

We consider the following time-fractional wave equation with initial and boundary conditions:

$$\frac{\partial^\alpha V}{\partial t^\alpha} = C^2 \frac{\partial^2 V}{\partial x^2}, \quad 1 < \alpha \leq 2, \quad (x,t) \in \Omega = [0,L] \times [0,T] \tag{1}$$

initial conditions:

$$V(x,0) = f(x), \quad \frac{\partial}{\partial t} V(x,0) = g(x), \quad x \in [0,L] \tag{2}$$

boundary conditions:

$$V(0,t) = 0, \quad V(L,t) = 0, \quad t \in (0,T] \tag{3}$$

where $V(x,t)$ is the amplitude of wave at position x and time t, and C is the velocity of wave. Here, $\frac{\partial^\alpha V}{\partial t^\alpha}$ is Caputo time fractional derivative of order α, which is defined as follows[9]:

$$\frac{\partial^\alpha V}{\partial t^\alpha} = \frac{1}{\Gamma(m-\alpha)} \int_0^t (t-\zeta)^{m-\alpha-1} \frac{\partial^m V(x,\zeta)}{\partial \zeta^m} d\zeta$$

where m is a integer such that $m-1 < \alpha \leq m$.

2. Finite difference Scheme

Let V_i^n be the numerical approximation of $V(x,t)$ at point (ih,nk), where h and k are spatial and temporal sizes respectively. Let $x_i = ih,\ i = 0,1,2,\dots,M$ and $t_n = nk,\ n = 0,1,2,\dots,N$, where $h = \frac{L}{M}$ and $k = \frac{T}{N}$. Now, the fractional order wave equation (1) - (3) is discretized by using the second-order accurate central difference formula for the derivative term in space and the Caputo finite difference formula for the time α-order fractional derivative for each interior grid point (ih,nk). We discretized the second order space derivative by second-order accurate central difference formula as follows:

$$\left(\frac{\partial^2 V}{\partial x^2}\right)_{(x_i,t_n)} = \frac{V_{i+1}^n - 2V_i^n + V_{i-1}^n}{h^2} + O(h^2) \tag{4}$$

and Caputo time fractional derivative as follows:

$$\begin{aligned}
\left(\frac{\partial^\alpha V}{\partial t^\alpha}\right)_{(x_i,t_n)} &= \frac{1}{\Gamma(2-\alpha)}\int_0^{t_n}(t_n-\zeta)^{1-\alpha}\frac{\partial^2 V(x_i,\zeta)}{\partial \zeta^2}d\zeta \\
&= \frac{1}{\Gamma(2-\alpha)}\sum_{j=0}^{n-1}\int_{jk}^{(j+1)k}\eta^{1-\alpha}\frac{\partial^2 V(x_i,t_n-\eta)}{\partial \eta^2}d\eta \\
&= \frac{1}{\Gamma(2-\alpha)}\sum_{j=0}^{n-1}\left[\frac{V_i^{n-j+1}-2V_i^{n-j}+V_i^{n-j-1}}{k^2}+O(k^2)\right]\int_{jk}^{(j+1)k}\eta^{1-\alpha}d\eta \\
&= \frac{k^{2-\alpha}}{\Gamma(3-\alpha)}\sum_{j=0}^{n-1}\left[\frac{V_i^{n-j+1}-2V_i^{n-j}+V_i^{n-j-1}}{k^2}+O(k^2)\right]\left[(j+1)^{2-\alpha}-j^{2-\alpha}\right] \\
&= \frac{k^{2-\alpha}}{\Gamma(3-\alpha)}\sum_{j=0}^{n-1}\left[\frac{V_i^{n-j+1}-2V_i^{n-j}+V_i^{n-j-1}}{k^2}\right]\left[(j+1)^{2-\alpha}-j^{2-\alpha}\right] \\
&\quad + \frac{k^{2-\alpha}}{\Gamma(3-\alpha)}\sum_{j=0}^{n-1}\left[(j+1)^{2-\alpha}-j^{2-\alpha}\right]O(k^2) \\
&= \frac{k^{-\alpha}}{\Gamma(3-\alpha)}\sum_{j=0}^{n-1}\left[V_i^{n-j+1}-2V_i^{n-j}+V_i^{n-j-1}\right]\left[(j+1)^{2-\alpha}-j^{2-\alpha}\right] \\
&\quad + \frac{k^{2-\alpha}}{\Gamma(3-\alpha)}n^{2-\alpha}O(k^2)
\end{aligned}$$

As $nk \le T$ is finite, then above formula can be rewritten as

$$\left(\frac{\partial^\alpha V}{\partial t^\alpha}\right)_{(x_i,t_n)} = \frac{k^{-\alpha}}{\Gamma(3-\alpha)}\sum_{j=0}^{n-1}b_j\left(V_i^{n-j+1}-2V_i^{n-j}+V_i^{n-j-1}\right)+O(k^2) \tag{5}$$

where

$$b_j = (j+1)^{2-\alpha} - j^{2-\alpha}, \quad j = 0,1,2,\ldots,n-1$$

Now, putting (4) and (5) in equation (1), we obtain

$$\frac{k^{-\alpha}}{\Gamma(3-\alpha)}\sum_{j=0}^{n-1} b_j\left(V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1}\right) = C^2\left(\frac{V_{i+1}^n - 2V_i^n + V_{i-1}^n}{h^2}\right)$$

$$\sum_{j=0}^{n-1} b_j\left(V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1}\right) = \frac{\Gamma(3-\alpha)k^{\alpha}C^2}{h^2}\left(V_{i+1}^n - 2V_i^n + V_{i-1}^n\right)$$

$$V_i^{n+1} - 2V_i^n + V_i^{n-1} + \sum_{j=1}^{n-1} b_j\left(V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1}\right) = \frac{\Gamma(3-\alpha)k^{\alpha}C^2}{h^2}\left(V_{i+1}^n - 2V_i^n + V_{i-1}^n\right)$$

After simplification, we get

$$V_i^{n+1} = 2V_i^n - V_i^{n-1} + \mu\left(V_{i+1}^n - 2V_i^n + V_{i-1}^n\right) - \sum_{j=1}^{n-1} b_j\left(V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1}\right) \tag{6}$$

where $\mu = \frac{\Gamma(3-\alpha)k^{\alpha}C^2}{h^2}$.

The initial conditions are approximated as follows:

$$V(x_i,0) = f(x_i) \text{ implies } V_i^0 = f(x_i),\ i = 1,2,\ldots,M-1 \tag{7}$$

and

$$\frac{\partial}{\partial t}V(x_i,t_0) = g(x_i) \text{ implies } \frac{V_i^1 - V_i^{-1}}{2k} = g(x_i)$$

$$V_i^{-1} = V_i^1 - 2kg(x_i),\ i = 1,2,\ldots,M-1 \tag{8}$$

Also, the boundary conditions are approximated as follows:

$$V(0,t_n) = 0 \text{ implies } V_0^n = 0,\ n = 1,2,\ldots,N-1$$

and

$$V(L,t_n) = 0 \text{ implies } V_M^n = 0,\ n = 1,2,\ldots,N-1$$

Now, putting $n = 0$ in equation (6) and using equation (8), we obtain

$$V_i^1 = V_i^0 + kg(x_i) + \frac{\mu}{2}\left(V_{i+1}^0 - 2V_i^0 + V_{i-1}^0\right)$$

For $n = 1$, we have

$$V_i^2 = 2V_i^1 - V_i^0 + \mu\left(V_{i+1}^1 - 2V_i^1 + V_{i-1}^1\right)$$

and for $n = 2,3,\ldots,N-1$, we have

$$V_i^{n+1} = 2V_i^n - V_i^{n-1} + \mu\left(V_{i+1}^n - 2V_i^n + V_{i-1}^n\right) - \sum_{j=1}^{n-1} b_j\left(V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1}\right)$$

The complete discretized time-fractional wave equation with initial and boundary conditions is written as follows:

$$V_i^1 = V_i^0 + kg(x_i) + \frac{\mu}{2}\left(V_{i+1}^0 - 2V_i^0 + V_{i-1}^0\right) \tag{9}$$

$$V_i^2 = 2V_i^1 - V_i^0 + \mu\left(V_{i+1}^1 - 2V_i^1 + V_{i-1}^1\right) \tag{10}$$

$$\begin{aligned} V_i^{n+1} &= 2V_i^n - V_i^{n-1} + \mu\left(V_{i+1}^n - 2V_i^n + V_{i-1}^n\right) \\ &\quad - \sum_{j=1}^{n-1} b_j\left(V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1}\right), \text{ for } n = 2,3,\ldots N-1, \end{aligned} \tag{11}$$

initial condition:

$$V_i^0 = f(x_i),\ i = 1,2,\ldots,M-1 \tag{12}$$

boundary conditions:

$$V_0^n = 0,\ V_M^n = 0,\ n = 1,2,\ldots,N-1 \tag{13}$$

The discretized finite difference scheme (9)-(13) can be written in matrix form as follows:

$$V^1 = A_1V^0 + F \tag{14}$$

$$V^2 = 2A_1V^1 - V^0 \tag{15}$$

$$\begin{aligned} V^{n+1} &= A_2V^n + (-1+2b_1-b_2)V^{n-1} + (-b_1+2b_2-b_3)V^{n-2} + \ldots \\ &\quad + (-b_{n-2}+2b_{n-1})V^1 + (-b_{n-1})V^0, \text{ for } n = 2,3,\ldots N-1, \end{aligned} \tag{16}$$

initial condition:

$$V_i^0 = f(x_i), \quad i = 1,2,\ldots,M-1 \tag{17}$$

boundary conditions:

$$V_0^n = 0, \quad V_M^n = 0, \quad n = 1,2,\ldots,N-1 \tag{18}$$

where $V^n = \left[V_1^n, V_2^n, \ldots, V_{M-1}^n\right]^t$, $F = [kg(x_1), kg(x_2), \ldots, kg(x_{M-1})]^t$,

$$A_1 = \begin{pmatrix} 1-\mu & \frac{\mu}{2} & & & & & \\ \frac{\mu}{2} & 1-\mu & \frac{\mu}{2} & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \frac{\mu}{2} & 1-\mu & \frac{\mu}{2} & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \frac{\mu}{2} & 1-\mu & \frac{\mu}{2} \\ & & & & & \frac{\mu}{2} & 1-\mu \end{pmatrix}$$

and

$$A_2 = \begin{pmatrix} 2-2\mu-b_1 & \mu & & & & & \\ \mu & 2-2\mu-b_1 & \mu & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & \mu & 2-2\mu-b_1 & \mu & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & \mu & 2-2\mu-b_1 & \mu \\ & & & & & \mu & 2-2\mu-b_1 \end{pmatrix}$$

3. STABILITY

In this section, we discuss the stability of solution of the explicit finite difference scheme (9)-(13) develop for time-fractional wave equation (1)-(3).

Lemma 3.1. *The eigenvalues of the $N \times N$ tri-diagonal matrix*

$$\begin{pmatrix} a & b & & & & & \\ c & a & b & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & c & a & b & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & c & a & b \\ & & & & & c & a \end{pmatrix}$$

are given as

$$\lambda_s = a + 2\sqrt{bc}\cos\left(\frac{s\pi}{N+1}\right),\ s = 1,2,\ldots,N$$

where a, b and c may be real or complex[13].

Theorem 3.2. *The solution of fractional order explicit finite difference scheme (9)-(13) for time-fractional wave equation (1)-(3) is conditionally stable.*

Proof. The eigenvalues of tri-diagonal matrix A_1 are given by,

$$\begin{aligned}\lambda_s(A_1) &= 1-\mu+2\sqrt{\frac{\mu^2}{4}}\cos\left(\frac{s\pi}{M}\right),\ \text{for } s = 1,2,\ldots M-1\\ &= 1-\mu+\mu\cos\left(\frac{s\pi}{M}\right)\\ &\le 1-\mu+\mu = 1\end{aligned}$$

$$\begin{aligned}\lambda_s(A_1) &= 1-\mu+\mu\cos\left(\frac{s\pi}{M}\right),\ \text{for } s = 1,2,\ldots M-1\\ &\ge 1-\mu-\mu\\ &\ge 1-2\mu\\ &\ge -1 \ \text{ when } \ 1-2\mu \ge -1 \Rightarrow \mu \le 1\end{aligned}$$

Therefore, we have

$$|\lambda_s(A_1)| \le 1 \text{ for } 0 < \mu \le 1 \tag{19}$$

Similarly, eigenvalues of tri-diagonal matrix $2A_1$ are given by,

$$\begin{aligned}\lambda_s(2A_1) &= 2-2\mu+2\mu\cos\left(\frac{s\pi}{M}\right),\ \text{for } s = 1,2,\ldots M-1\\ &\ge 2-2\mu-2\mu\\ &\ge 2-4\mu\\ &\ge -1 \ \text{ when } \ 2-4\mu \ge -1 \Rightarrow \mu \le \frac{3}{4}\end{aligned}$$

$$\begin{aligned}\lambda_s(2A_1) &= 2-2\mu+2\mu\cos\left(\frac{s\pi}{M}\right),\ \text{for } s = 1,2,\ldots M-1\\ &= 2-4\mu\sin^2\left(\frac{s\pi}{2M}\right)\\ &\le 1\end{aligned}$$

when $2-4\mu\sin^2\left(\frac{s\pi}{2M}\right)\leq 1 \Rightarrow 1\leq 4\mu\sin^2\left(\frac{s\pi}{2M}\right)\Rightarrow 1\leq 4\mu \Rightarrow \frac{1}{4}\leq\mu$.

Therefore, we have

$$|\lambda_s(2A_1)|\leq 1 \text{ for } \frac{1}{4}\leq\mu\leq\frac{3}{4} \tag{20}$$

Now, the eigenvalues of tri-diagonal matrix A_2 are given by,

$$\begin{aligned}\lambda_s(A_2) &= 2-2\mu-b_1+2\mu\cos\left(\frac{s\pi}{M}\right), \text{ for } s=1,2,\dots M-1\\ &\geq 2-2\mu-b_1-2\mu\\ &\geq 2-4\mu-b_1\\ &\geq -1 \text{ when } 2-4\mu-b_1\geq -1\Rightarrow \mu\leq\frac{3-b_1}{4}\end{aligned}$$

$$\begin{aligned}\lambda_s(A_2) &= 2-2\mu-b_1+2\mu\cos\left(\frac{s\pi}{M}\right), \text{ for } s=1,2,\dots M-1\\ &= 2-b_1-4\mu\sin^2\left(\frac{s\pi}{2M}\right)\\ &\leq 1\end{aligned}$$

when $2-b_1-4\mu\sin^2\left(\frac{s\pi}{2M}\right)\leq 1\Rightarrow 1-b_1\leq 4\mu\sin^2\left(\frac{s\pi}{2M}\right)\Rightarrow 1-b_1\leq 4\mu\Rightarrow\frac{1-b_1}{4}\leq\mu$.

Hence, we have

$$|\lambda_s(A_2)|\leq 1 \text{ for } \frac{1-b_1}{4}\leq\mu\leq\frac{3-b_1}{4}. \tag{21}$$

Therefore, from equations (19), (20) and (21), we prove the spectral radius $\rho(A)$ of matrices $A=A_1,2A_1,A_2$, satisfies $\rho(A)\leq 1$ if

$$\max\left\{0,\frac{1}{4},\frac{1-b_1}{4}\right\}\leq\mu\leq\min\left\{1,\frac{3}{4},\frac{3-b_1}{4}\right\}$$

Hence, this proves the theorem. □

4. CONVERGENCE

In this section, we discuss the question of convergence. Let $\overline{V}_i^n$ be the exact solution of time-fractional wave equation (1)-(3) and τ_i^n be the local truncation error for $1\leq i\leq M$. The finite difference scheme (9)-(13) will become

$$\tau_i^1=\overline{V}_i^1-\overline{V}_i^0-kg(x_i)-\frac{\mu}{2}\left(\overline{V}_{i+1}^0-2\overline{V}_i^0+\overline{V}_{i-1}^0\right)=O(h^2+k^2)$$

$$\tau_i^2 = \overline{V}_i^2 - 2\overline{V}_i^1 + \overline{V}_i^0 + \mu\left(\overline{V}_{i+1}^1 - 2\overline{V}_i^1 + \overline{V}_{i-1}^1\right) = O(h^2 + k^2)$$

and for $2 \le n \le N-1$,

$$\tau_i^{n+1} = \overline{V}_i^{n+1} - 2\overline{V}_i^n + \overline{V}_i^{n-1} - \mu\left(\overline{V}_{i+1}^n - 2\overline{V}_i^n + \overline{V}_{i-1}^n\right) + \sum_{j=1}^{n-1} b_j\left(\overline{V}_i^{n-j+1} - 2\overline{V}_i^{n-j} + \overline{V}_i^{n-j-1}\right) = O(h^2+k^2)$$

Lemma 4.1. *The coefficient $b_j,\ j = 1,2,\ldots$ satisfy*

(i) $b_j > 0$

(ii) $b_j > b_{j+1}$

Theorem 4.2. *Let $\overline{V}_i^n$ be the exact solution of time-fractional wave equation (1)-(3) and V_i^n be the numerical solution of explicit finite difference scheme (9)-(13) at each mesh point (x_i, t_n). Then there exist a positive constant K independent of h and k such that*

$$\|\overline{V}_i^n - V_i^n 1\| \le K(h^2 + k^2),\ when\ \frac{1}{4} \le \mu \le \frac{3 - b_1}{4}.$$

Proof. Let e_i^n be the error at each mesh point (x_i, t_n), then

$$e_i^n = \overline{V}_i^n - V_i^n$$

Now, we denote the error vector by $e^n = (e_1^n, e_2^n, \ldots, e_{M-1}^n)$ for $1 \le n \le N$ and local truncation error vector by $\tau^n = (\tau_1^n, \tau_2^n, \ldots, \tau_{M-1}^n)$ for time level n. From equations (14)-(16), we obtain

$$e^1 = A_1 e^0 + \tau^1$$

$$e^2 = 2A_1 e^1 - e^0 + \tau^2$$

$$e^{n+1} = A_2 e^n + (-1 + 2b_1 + b_2)e^{n-1} + (-b_1 + 2b_2 - b_3)e^{n-2} + \ldots$$
$$+ (-b_{n-2} + 2b_{n-1})e^1 + (-b_{n-1})e^0 + \tau^{n+1} \qquad \text{for } n = 2, 3, \ldots N-1.$$

Using mathematical induction, we will prove that $\|e^n\|_\infty \le K(h^2 + k^2)$.

For $n = 1$, we have

$$\max_{1 \le i \le M-1} |e_i^1| \le \|e^1\|_\infty = \|A_1 e^0 + \tau^1\|_\infty \le \|\tau^1\|_\infty \le K(h^2 + k^2)$$

where K is independent of h and k. Also for $n=2$, we have

$$\begin{aligned}\max_{1\le i\le M-1}|e_i^2| \le \|e^2\|_\infty &= \|2A_1e^1 - e^0 + \tau^2\|_\infty \\ &\le \|2A_1e^1\|_\infty + \|\tau^2\|_\infty \\ &\le \|e^1\|_\infty + \|\tau^2\|_\infty \\ &\le K_1(h^2+k^2) + K_2(h^2+k^2) \\ &\le K(h^2+k^2)\end{aligned}$$

where K is independent of h and k.

Suppose that

$$\max_{1\le i\le M-1}|e_i^r| \le \|e^r\|_\infty \le K(h^2+k^2)$$

for $r\le n$ and K is independent of h and k.

Consider,

$$\begin{aligned}\max_{1\le i\le M-1}|e_i^{n+1}| \le \|e^{n+1}\|_\infty &\le \|A_2e^n + (-1+2b_1-b_2)e^{n-1} + (-b_1+2b_2-b_3)e^{n-2}+\dots \\ &\quad + (-b_{n-2}+2b_{n-1})e^1 + (-b_{n-1})e^0 + \tau^{n+1}\|_\infty \\ &\le \|A_2\|.\|e^n\|_\infty + \|(-1+2b_1-b_2)e^{n-1} + (-b_1+2b_2-b_3)e^{n-2}+\dots \\ &\quad + (-b_{n-2}+2b_{n-1})e^1 + (-b_{n-1})e^0 + \tau^{n+1}\|_\infty \\ &\le \|A_2\|.\|e^n\|_\infty + |-1+2b_1-b_2|.\|e^{n-1}\|_\infty + |-b_1+2b_2-b_3|.\|e^{n-2}\|_\infty + \\ &\quad \dots + |-b_{n-2}+2b_{n-1})|.\|e^1\|_\infty + |-b_{n-1}|.\|e^0\|_\infty + \|\tau^{n+1}\|_\infty \\ &\le (1+1-2b_1+b_2+b_1-2b_2+b_3+\dots \\ &\quad + b_{n-2}-2b_{n-1}+b_{n-1}+2(-b_{n-2}+2b_{n-1}))K_2(h^2+k^2) + K_1(h^2+k^2) \\ &\le (2-b_1+2(-b_{n-2}+2b_{n-1})+b_{n-1})K_2(h^2+k^2) + K_1(h^2+k^2) \\ &\le K(h^2+k^2)\end{aligned}$$

where K is a positive constant independent of h and k. Hence, by mathematical induction, for all

$n=1,2,\dots,N$, we have

$$\max_{1\le i\le M-1}|e_i^n| \le \|e^n\|_\infty \le K(h^2+k^2)$$

Therefore, we conclude that if

$$\frac{1}{4} \leq \mu \leq \frac{3-b_1}{4}$$

then $\|e^n\|_\infty \to 0$ as $(h,k) \to (0,0)$. Therefore, we proves that V_i^n converges to $\overline{V}_i^n$.

This completes the proof. □

5. Python Programme

We compute V_i^n at each grid point (x_i, t_n) using proposed scheme by Python. Now, the algorithm for scheme (9)-(13) as follows:

(1) Compute $V_i^0 = f(x_i), \quad i = 0, 1, 2, \ldots, M.$

(2) Compute $V_i^1, V_i^2, \quad i = 0, 1, 2, \ldots, M.$

(3) Compute V_i^{n+1}, for each $n = 2, 3, \ldots, N-1,\ i = 0, 1, 2, \ldots, M.$

Now, we develop the Python programme TFW for complete discretized scheme (9)-(13) as follows:

Inputs:

f - initial displacement

g - initial velocity

C - velocity of wave

L - spatial length

T - end time

k - temporal size

mu - μ

a - fractional order α of time derivative

t1 - time level, at which solution has to be estimate

Output of Python programme TFW is the approximate value of vector $V(x_i, t1)$.

```
import numpy as np
def TFW(f,g,C,L,T,k,mu,a,t1):
    d=(gamma(3-a)*(k)**a*C**2)/mu
    h=sqrt(d)
    N=int(round(T/k))
    M=int(round(L/h))
    t=np.linspace(0,N*k,N+1)
x=np.linspace(0,M*h,M+1)
V=np.zeros((N+1,M+1))
for i in range(0,M+1):
    V[0][i]=f(x[i])
for i in range(1,M):
    V[1][i]=V[0][i]+k*g(x[i])+0.5*mu*(V[0][i+1]-2*V[0][i]+V[0][i-1])
V[1][0]=0; V[1][M]=0
for i in range(1,M):
    V[2][i]=2*V[1][i]-V[0][i]+mu*(V[1][i+1]-2*V[1][i]+V[1][i-1])
V[2][0]=0; V[2][M]=0
for n in range(2,N):
    for i in range(1,M):
        S=0
        for j in range(1,n):
            S=S+((j+1)**(2-a)-j**(2-a))*(V[n-j+1][i]-2*V[n-j][i]+V[n-j-1][i])
        V[n+1][i]=2*V[n][i]-V[n-1][i]+mu*(V[n][i+1]-2*V[n][i]+V[n][i-1])-S
    V[n+1][0]=0; V[n+1][M]=0
return(x,V[t1])
```

Numerical Solutions:

We consider the following time-fractional wave equation:

$$\frac{\partial^\alpha V}{\partial t^\alpha} = C^2 \frac{\partial^2 V}{\partial x^2}, \quad (x,t) \in \Omega = [0,1] \times [0,1]$$

with initial conditions:

$$V(x,0) = \sin(5\pi x) + 2\sin(7\pi x), \quad \frac{\partial}{\partial t} V(x,0) = 0, \quad x \in [0,1]$$

and boundary conditions,

$$V(0,t) = 0, \quad V(1,t) = 0, \quad t \in (0,1]$$

The exact solution to this problem is

$$V(x,y) = \sin(5\pi x)\cos(5\pi Ct) + 2\sin(7\pi x)\cos(7\pi Ct)$$

Using the python programme TFW, we estimate the value of $V(x,t)$ for any time level t_n. In Table 1, we compare the exact solution and numerical solution for $\alpha = 1.99$ with parameters

$h = \frac{1}{100}$, $k = \frac{1}{150}$, $C = 1$ and $t = 1$.

TABLE 1. Comparison of exact solution and numerical solution for $\alpha = 1.99, h = \frac{1}{100}, k = \frac{1}{150}, C = 1, t = 1$

x	Exact solution	Numerical solution	Difference between the solutions
0.1	−2.6180339887	−2.2165435008	0.40149
0.2	1.9021130326	1.5695507898	0.33256
0.3	0.3819660113	0.3714258869	0.01054
0.4	−1.1755705046	−0.9700357352	0.20553
0.5	1.0	0.7689195118	0.23108
0.6	−1.1755705046	−0.9700357352	0.20553
0.7	0.3819660113	0.3714258869	0.01054
0.8	1.9021130326	1.5695507898	0.33256
0.9	−2.6180339887	−2.2165435008	0.40149
1.0	0.0	0.0	0.0

In Table 2, we compare the exact solution and numerical solution for parameters $\alpha = 2, h = \frac{1}{100}, k = \frac{1}{150}, C = 1$ and $t = 1$.

From these tables, we conclude that the proposed scheme gives accurate results and stable solutions. Using the Python programme TFW, we obtain numerical solutions of time fractional wave equation for $\alpha = 1.97, 1.98, 1.99$ and compare with exact solution with the parameters $\mu = \frac{3}{4}, k = \frac{1}{100}, C = 1$, $t = 1$ and represented graphically in Figure 1.

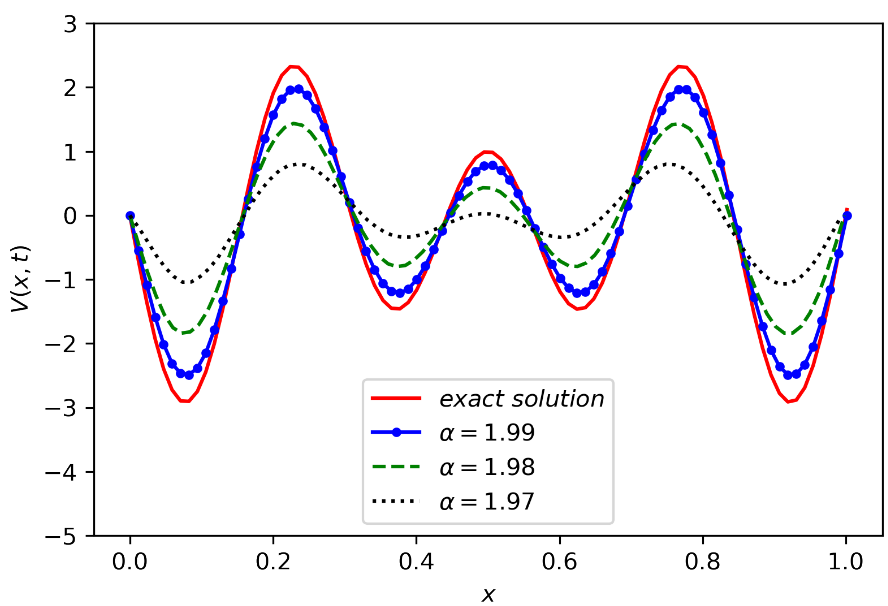
We obtain the numerical solutions for $\mu = 0.9$ and 1.0 with parameters $\alpha = 1.5, k = \frac{1}{100}, C = 1$, $t = 1$ and represented graphically using Python programme TFW in Figure 2.

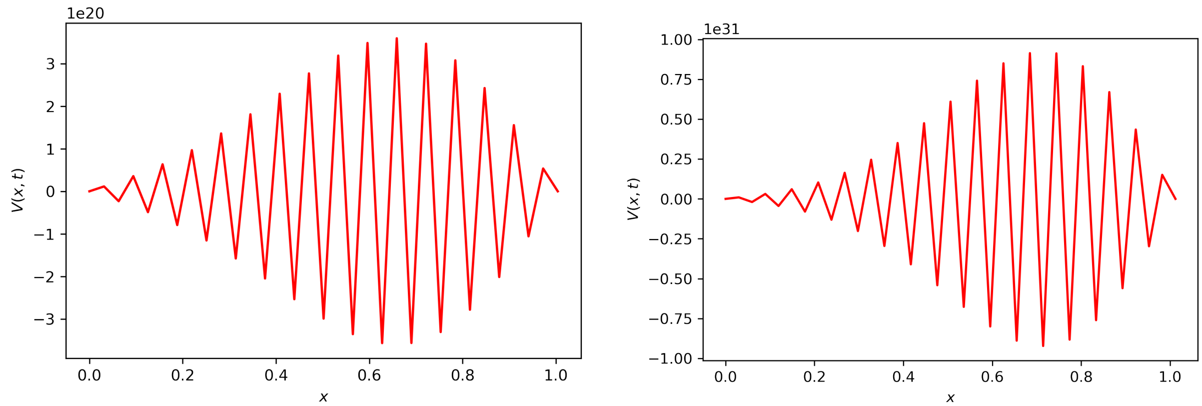
We observe that $\mu > 0.75$ then approximate solution obtained by the develop scheme using Python programme TFW is unstable.

TABLE 2. Comparison of exact and numerical solution for $\alpha = 2, h = \frac{1}{100}, k = \frac{1}{150}, C = 1, t = 1$

x	Exact solution	Numerical solution	Relative error $e_i^n = \|\overline{V}_i^n - V_i^n\|$
0.1	−2.6180339887	−2.6175015948	0.000532
0.2	1.9021130326	1.9015345652	0.000578
0.3	0.3819660113	0.3821136465	0.000147
0.4	−1.1755705046	−1.1752129921	0.000357
0.5	1.0	0.9994320835	0.000567
0.6	−1.1755705046	−1.1752129921	0.000357
0.7	0.3819660113	0.3821136465	0.000147
0.8	1.9021130326	1.9015345652	0.000578
0.9	−2.6180339887	−2.6175015948	0.000532
1.0	0.0	0.0	0.0

FIGURE 1. The nature of approximate solutions for the parameters $\mu = \frac{3}{4}, k = \frac{1}{100}, t = 1, C = 1$





(A) $\mu = 0.9$ (B) $\mu = 1.0$

FIGURE 2. The nature of approximate solutions for the parameters $\alpha = 1.5, t = 1, C = 1, k = \frac{1}{100}$

6. Conclusions

(i) We successfully develop the fractional order finite difference scheme for time fractional wave equation.

(ii) Theoretically, we proved that the developed scheme is conditionally stable and bound of stability is $\frac{1}{4} \leq \mu \leq \frac{3-b_1}{4}$, where $0 \leq b_1 \leq 1$.

(iii) The convergence of the scheme is theoretically proved by using maximum norm method.

(iv) We successfully develop a python programme for time fractional wave equation.

(v) Analysis shows that the finite difference scheme is numerically stable and the results are compatible with our theoretical analysis.

(vi) We observe that Python is very powerful tool, which allows for the convenient computation of finite difference schemes for solving fractional order differential equations. The numerical solution of test problem is obtained successfully by Python programme and represented graphically.

Conflict of Interests

The author(s) declare that there is no conflict of interests.

REFERENCES

[1] D. Dhaigude, K. Takale, A weighted average finite difference scheme for one dimensional pennes bioheat equation, Proc. Neural Parallel Sci. Comput. 4 (2010), 124-128.

[2] R. Hilfer, Applications of Fractional Calculus in Physics, World Scientific, Singapore, 2000.

[3] S. Holm, S.P. NÃ¤sholm, F. Prieur, R. Sinkus, Deriving fractional acoustic wave equations from mechanical and thermal constitutive equations, Computers Math. Appl. 66 (2013), 621–629.

[4] S. Jogdand, K. Takale, A. Jagtap, Finite difference approximation for space fractional soil moisture diffusion equation and its application, IOSR J. Math. 8 (2013), 01–08.

[5] H.P. Langtangen, S. Linge, Finite Difference Computing with PDEs: A Modern Software Approach, volume 16 of Texts in Computational Science and Engineering. Springer International Publishing, Cham, 2017.

[6] J. Li, Y.-T. Chen, Computational partial differential equations using MATLAB, CRC Press, Boca Raton, 2020.

[7] R.B. Lindsay, Acoustics: historical and philosophical development, Benchmark papers in acoustics, Dowden, Hutchinson and Ross, Stroudsburg, PA, 1973.

[8] Y. Luchko, Fractional wave equation and damped waves, J. Math. Phys. 54 (2013), 031505.

[9] I. Podlubny, Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications, Number v. 198 in Mathematics in science and engineering, Academic Press, San Diego, 1999.

[10] A. Pskhu, S. Rekhviashvili, Fractional Diffusion–Wave Equation with Application in Electrodynamics, Mathematics, 8 (2020), 2086.

[11] K.A. Seeler, Finite Difference Methods and MATLAB, In System Dynamics, pages 467–517, Springer New York, New York, NY, 2014.

[12] J. Singh, D. Kumar, D. Baleanu, S. Rathore, On the local fractional wave equation in fractal strings, Math. Meth. Appl. Sci. 42 (2019), 1588–1595.

[13] G.D. Smith, Numerical Solution of Partial Differential Equations: Finite Difference Methods, Clarendon Press, Oxford, 1985.

[14] Kalyanrao Takale. Fractional order finite difference scheme for space fractional boussinesq's equation and its application. International Journal of Mathematical Sciences and Applications, 3, 01 2013.

[15] K.C. Takale, V.R. Nikam, S.R. Kulkarni, Fractional order finite difference scheme for space fractional diffusion equation, Eng. Autom. Probl. 3 (2014), 120–124.

[16] K. Takale, M. Datar, S. Kulkarni, Approximate solution of space fractional partial differential equations and its applications, Int. J. Eng. Res. Computer Sci. Eng. 5 (2018), 216–220.

[17] K. Takale, D. Dhaigude, V. Nikam, Douglas higher order finite difference scheme for one dimensional pennes bioheat equation, Int. J. Adv. Eng. Appl. 06 (2011).

[18] Y. Wang, Y. Ning, Y. Wang, Fractional Time Derivative Seismic Wave Equation Modeling for Natural Gas Hydrate, Energies, 13 (2020), 5901.

[19] M. Ziółkowski, L. Stepień, M.R. Stepień, A. Gola, Applications of Python programs in solving of equations based on selected numerical methods, Sci. Issues Jan Dlugosz Univ. Czestochowa, Mathematics, 22 (2017), 31–45.