Advances and Applications in Mathematical Sciences
Volume 22, Issue 2, December 2022, Pages 503-526

PYTHON: POWERFUL TOOL FOR SOLVING SPACE-TIME FRACTIONAL TRAVELING WAVE EQUATION

KRISHNA GHODE, KALYANRAO TAKALE, SHRIKISAN GAIKWAD and KIRANKUMAR BONDAR

Department of Mathematics
B. K. Birla College of Arts
Science and Commerce, Kalyan
Thane-421301, (M.S.), India

Department of Mathematics
RNC Arts, JDB Commerce and
NSC Science College, Nashik Road
Nashik-422101, (M. S.), India
E-mail: kalyanraotakale1@gmail.com

Department of Mathematics
New Atrs, Commerce and Science College
Ahmadnagar-414001, (M. S.), India
E-mail: sbgmathsnagar@gmail.com

Department of Mathematics
Government Vidarbha Institute
of Science and Humanities
Amravati 444604, (M.S.), India
E-mail: klbondar75@rediffmail.com

Abstract

The aim of this paper is to investigate the solution of space-time fractional traveling wave equation by Crank-Nicolson finite difference method using Python Programme. Also, we prove the scheme is unconditionally stable and convergent. Furthermore, we develop the Python programme for the proposed scheme and estimate the error. Finally, we obtain the numerical solutions of some test problems and simulated graphically by a Python programme.

2020 Mathematics Subject Classification: 35L05, 26A33, 65M06.

Keywords: Fractional traveling wave equation, Caputo derivative, Grunwald-Letnikov derivative, stability, Python etc.

*Corresponding author; E-mail: ghodekrishna@gmail.com

Received September 17, 2021; Accepted January 1, 2022

1. Introduction

In recent years, fractional differential equations are becoming a significant implement in the analysis and modeling of scientific problems in a broad array of fields such as physics, engineering, biology, finance, economics and earthquakes study etc. [2, 6, 9, 11, 10, 17]. There has been increasing attention in the description of physical and chemical processes using equations involving fractional derivatives and integrals. The study of fractional differential equations has been a highly focused in recent years. But most of the fractional differential equations do not have exact solutions. Traveling wave analysis is the most significant approach to study linear and non-linear partial differential equations. This study leads to various types of solutions such as soliton solutions, periodic solutions, kink solutions, cuspons solutions, compacton solutions, peakon solutions etc. [18]. The traveling wave solutions of fractional order partial differential equations are useful to analyses the mechanisms of phenomena as well as further application in various fields. Though, finding traveling wave solutions is not a straightforward task at all, therefore researchers are preferring finite difference schemes.

The finite difference approximations for derivatives are one of the simplest and the efficient method to solve fractional order partial differential equations [1, 3, 14, 13, 16]. Therefore, in this connection we develop the Crank-Nicolson finite difference scheme for space time fractional traveling wave equation and obtain its solution using Python programme. Recently, many researchers have shifted from compiled languages to interpreted problem solving environments, such as MATLAB, Maple, Octave, R etc. [5, 12, 15]. The Python is now rising as a potentially competitive replacement to MATLAB, Octave, and other similar environments [4, 7]. The popularity of Python is because of simple and clean syntax of the commands, incorporation of simulation and visualization, interactive execution of commands with instantaneous feedback and lots of built-in functions available and work efficiently on arrays in compiled code. Now a days, researchers are using Python due to its simplicity, wealth of available support and the NumPy package, which provides contiguous multi-dimensional array structures with a large library of array operations.

The plan of the paper is as follows: In section 2, the Crank-Nicolson finite difference scheme is developed for space-time fractional traveling wave equation. The section 3 is devoted for stability of the scheme and the question of convergence is proved in section 4. The last section is devoted for Python programme and numerical solution of the space-time fractional traveling wave equations.

We consider the following space-time fractional traveling wave equation

$$\frac{\partial^\alpha V}{\partial t^\alpha} = C^2 \frac{\partial^\beta V}{\partial x^\beta}, \; t \in [0, T], \; x \in [0, L], \; 1 < \alpha \le 2, \; 1 < \beta \le 2 \tag{1.1}$$

subject to the initial conditions:

$$V(x, 0) = f(x), \; \frac{\partial}{\partial t} V(x, 0) = g(x), \; x \in [0, L] \tag{1.2}$$

and boundary conditions:

$$V(0, t) = 0, \; V(L, t) = 0, \; t > 0 \tag{1.3}$$

where $V(x, t)$ is the displacement of wave at position x and time t, and C is the velocity of wave. The Caputo time fractional derivative of order α is defined as follows [8],

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

where m is a integer such that $m - 1 < \alpha \le m$. The right shifted Grunwald-Letnikov space fractional derivative of order β is defined as follows [8],

$$\frac{\partial^\alpha V}{\partial t^\alpha} = \lim_{M \to \infty} \frac{1}{h^\beta} \sum_{l=0}^{M} \frac{\Gamma(l-\beta)}{\Gamma(-\beta)\Gamma(l+1)} V(x-(l-1)h, t).$$

2. Finite Difference Scheme

In this section, we discretized the space-time fractional traveling wave equation (1.1)-(1.3) using Crank-Nicolson finite difference scheme. Let $x_i = ih, i = 0, 1, 2, \ldots, M$ and $t_n = nk, n = 0, 1, 2, \ldots, N$, where $h = \frac{L}{M}$

and $k = \frac{T}{M}$. 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. We discretized the Caputo time fractional derivative as follows:

$$\left(\frac{\partial^\alpha V}{\partial t^\alpha}\right)_{(x_i,\, t_{n+1})} = \frac{1}{\Gamma(2-\alpha)}\int_0^{t_{n+1}} (t_{n+1} - \zeta)^{1-\alpha}\, \frac{\partial^2 V(x_i,\, \zeta)}{\partial \zeta^2}\, d\zeta$$

$$= \frac{1}{\Gamma(2-\alpha)}\sum_{j=0}^{n}\int_{jk}^{(j+1)k} \eta^{1-\alpha}\, \frac{\partial^2 V(x_i,\, t_{n+1} - \eta)}{\partial \eta^2}\, d\eta$$

$$= \frac{1}{\Gamma(2-\alpha)}\sum_{j=0}^{n}\left(\frac{V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1}}{k^2} + O(k)\right)$$

$$\times \int_{jk}^{(j+1)k} \eta^{1-\alpha} d\eta$$

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

$$\times((j+1)^{2-\alpha} - j^{2-\alpha})$$

$$= \frac{k^{-\alpha}}{\Gamma(3-\alpha)}\sum_{j=0}^{n}(V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1})((j+1)^{2-\alpha} - j^{2-\alpha})$$

$$+ \frac{k^{2-\alpha}}{\Gamma(3-\alpha)} O(k)\sum_{j=0}^{n}((j+1)^{2-\alpha} - j^{2-\alpha})$$

$$= \frac{k^{-\alpha}}{\Gamma(3-\alpha)}\sum_{j=0}^{n}(V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1})((j+1)^{2-\alpha} - j^{2-\alpha})$$

$$+ \frac{k^{2-\alpha}}{\Gamma(3-\alpha)}(n+1)^{2-\alpha} O(k)$$

As $(n+1)k$ is finite, then above formula can be rewritten as

$$\left(\frac{\partial^{\alpha} V}{\partial t^{\alpha}}\right)_{(x_i,\, t_{n+1})} =$$

$$\frac{k^{-\alpha}}{\Gamma(3-\alpha)} \sum_{j=0}^{n} (V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1})((j+1)^{2-\alpha} - j^{2-\alpha}) + O(k) \tag{2.1}$$

where

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

We use the right shifted Grunwald formula to discretized the space fractional derivative as follows [8]:

$$\frac{\partial^{\beta} V}{\partial t^{\beta}} = \frac{1}{h^{\beta}} \sum_{l=0}^{i+1} w_l V_{i-l+1}^{n} + O(h) \tag{2.2}$$

where

$$w_l = \frac{\Gamma}{\Gamma(-\beta)\Gamma(l+1)},\ l = 0, 1, 2, \ldots, M.$$

which called the normalized Grunwald weights. They can be found by the recursive formula:

$$w_0 = 1,\ w_l = w_{l-1}\left(1 - \frac{\beta+1}{l}\right)$$

Now, putting (2.1) and (2.2) in equation (1.1), we obtain the Crank-Nicloson type numerical approximation of space-time fractional traveling wave equation (1.1) as follows:

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

$$\left[\sum_{l=0}^{i+1} w_l V_{i-l+1}^{n+1} + \sum_{l=0}^{i+1} w_l V_{i-l+1}^{n}\right]$$

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

We simplify the above equation and obtain

$$V_i^{n+1} - 2V_i^n + V_i^{n-1} + \sum_{j=1}^{n} b_j (V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1})$$

$$= r \left[\sum_{l=0}^{i+1} w_l V_{i-l+1}^{n+1} + \sum_{l=0}^{i+1} w_l V_{i-l+1}^{n} \right]$$

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

$$V_i^{n+1} - r \sum_{j=1}^{i+1} w_l V_{i-l+1}^{n+1} = 2V_i^n + V_i^{n-1} - \sum_{j=1}^{n} b_j (V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1})$$

$$+ r \sum_{l=0}^{i+1} w_l V_{i-l+1}^{n} \tag{2.3}$$

The initial conditions are approximated as follows:

$$V(x_i, 0) = f(x_i) \text{ implies } V_i^0 = f(x_i), \; i = 1, 2, \ldots, M-1 \tag{2.4}$$

and

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

$$V_i^{-1} = V_i^1 - 2kg(x_i), \; i = 1, 2, \ldots, M-1. \tag{2.5}$$

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

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

We put $n = 0$ in equation (2.3) and using equation (2.5), we obtain

$$V_i^1 - \frac{r}{2}\sum_{l=0}^{i+1} w_l V_{i-l+1}^1 = V_i^0 + kg(x_i) + \frac{r}{2}\sum_{l=0}^{i+1} w_l V_{i-l+1}^0$$

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

$$V_i^{n+1} - r\sum_{l=0}^{i+1} w_l V_{i-l+1}^{n+1} = 2V_i^n - V_i^{n-1} - \sum_{j=1}^{n-1} b_j (V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1})$$

$$- 2b_n (V_i^1 - V_i^0 + kg(x_i)) + r\sum_{l=0}^{i+1} w_l V_{i-l+1}^n$$

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

$$V_i^1 - \frac{r}{2}\sum_{l=0}^{i+1} w_l V_{i-l+1}^1 = V_i^0 + kg(x_i) + \frac{r}{2}\sum_{l=0}^{i+1} w_l V_{i-l+1}^0 \tag{2.6}$$

for $n = 0$,

$$V_i^{n+1} - r\sum_{l=0}^{i+1} w_l V_{i-l+1}^{n+1} = 2V_i^n - V_i^{n-1} - \sum_{j=1}^{n-1} b_j (V_i^{n-j+1} - 2V_i^{n-j} + V_i^{n-j-1})$$

$$- 2b_n (V_i^1 - V_i^0 + kg(x_i)) + r\sum_{l=0}^{i+1} w_l V_{i-l+1}^n, \tag{2.7}$$

initial condition:

$$V_i^0 = f(x_i),\ i = 1, 2, \ldots, M-1 \tag{2.8}$$

boundary conditions:

$$V_i^n = 0,\ V_M^n = 0,\ n = 1, 2, \ldots, N-1 \tag{2.9}$$

The discretized finite difference scheme (2.6)-(2.9) can be written in matrix form as follows:

for $n = 0$,

$$AV^1 = (2I - A)V^0 + F_1 \tag{2.10}$$

for $n \geq 1$,

$$(2A - I)V^{n+1} = ((4 - b_1)I - 2A)V^n + \sum_{j=1}^{n-1}(2b_j - b_{j-1} - b_{j+1})V_i^{n-j}$$

$$- b_n V^1 + (2b_n - b_{n-1})V^0 + 2b_n kg(x_i)I, \tag{2.11}$$

initial condition:

$$V_i^0 = f(x_i),\ i = 1, 2, \ldots, M - 1 \tag{2.12}$$

boundary conditions:

$$V_i^n = 0,\ V_M^n = 0,\ n = 1, 2, \ldots, N - 1 \tag{2.13}$$

where $V^n = [V_1^n, V_2^n, \ldots, V_{M-1}^n]^T$, $F = [kg(x_1), kg(x_2), \ldots, kg(x_{M-1})]^T$, I is an identity matrix of order $(n - 1) \times (n - 1)$ and matrix A is defined as follows

$$A = (a_{ij})_{(M-1)\times(M-1)} = \begin{cases} 1 - \frac{r}{2}w_1, & j = i \\ -\frac{r}{2}w_0, & j = i + 1 \\ -\frac{r}{2}w_{i-j+1}, & j \leq i - 1 \\ 0, & j \geq i + 2 \end{cases}$$

Lemma 2.1. *The coefficient* b_j, $j = 1, 2, \ldots$ *satisfy*

(i) $b_j > 0$

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

Lemma 2.2. *Grunwald-Letnikov coefficients* w_l *satisfy*:

(i) $w_0 = 1$, $w_1 = -\beta$, $w_2 = \frac{\beta(\beta - 1)}{2}$

(ii) $1 \geq w_2 \geq w_3 \geq \ldots \geq 0$

(iii) $\sum_{l=0}^{n} w_l < 0,\ n \geq 1.$

3. Stability

Let $\overline{V}_i^n$ and V_i^n are exact and approximate solutions of the equation (1.1)-(1.3) respectively and ε_i^n be the error at each mesh point (x_i, t_n), then

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

From equations (2.6)-(2.7), we obtain for $n = 0$,

$$\varepsilon_i^1 - \frac{r}{2}\sum_{l=0}^{i+1} w_l \varepsilon_{i-l+1}^1 = \varepsilon_i^0 + \frac{r}{2}\sum_{l=0}^{i+1} w_l \varepsilon_{i-l+1}^0 \tag{3.1}$$

for $n \geq 1$,

$$\varepsilon_i^{n+1} - r\sum_{l=0}^{i+1} w_l \varepsilon_{i-l+1}^{n+1} = 2\varepsilon_i^n - \varepsilon_i^{n-1} - \sum_{j=1}^{n-1} b_j(\varepsilon_i^{n-j+1} - 2\varepsilon_i^{n-j} + \varepsilon_i^{n-j-1})$$

$$- 2b_n(\varepsilon_i^1 - \varepsilon_i^0) + r\sum_{l=0}^{i+1} w_l \varepsilon_{i-l+1}^n. \tag{3.2}$$

Theorem 3.1. *The solution of Crank-Nicolson finite difference scheme given by* (2.6)-(2.9) *developed for equation* (1.1)-(1.3) *is unconditionally stable*.

Proof. We denote the error vector by $E^n = (\varepsilon_1^n, \varepsilon_2^n, \ldots, \varepsilon_{M-1}^n)^T$ for $0 \leq n \leq N.$ Also, we assume that

$$|\, \varepsilon^n \,| = \max_{1 \leq i \leq M-1} |\, \varepsilon_i^n \,| = \|\, E^n \,\|_\infty, \text{ for } n = 0, 1, 2, \ldots, N.$$

Using mathematical induction, we will prove that $\|\, E^n \,\|_\infty \leq K_1 \|\, E^0 \,\|_\infty$, for $n = 0, 1, 2, \ldots, N,$ where K_1 is a positive number independent of h and k. Now, using Lemma (2.2) and equation (3.1), we obtain

$$|\, \varepsilon_i^1 \,| \leq |\, \varepsilon_i^1 \,| - \frac{r}{2}\sum_{l=0}^{i+1} w_l |\, \varepsilon_{i-l+1}^1 \,|$$

$$\le | \varepsilon_i^1 - \frac{r}{2}\sum_{l=0}^{i+1} w_l \varepsilon_{i-l+1}^1 |$$

$$\le | \varepsilon_i^0 - \frac{r}{2}\sum_{l=0}^{i+1} w_l \varepsilon_{i-l+1}^0 |$$

$$| \varepsilon_i^0 | + \frac{r}{2}\sum_{l=0}^{i+1} w_l | \varepsilon_{i-l+1}^1 | \le \left(1 + \frac{r}{2}\sum_{l=0}^{i+1} w_l\right) | \varepsilon^0 | \le K_1 | \varepsilon^0 |$$

$$\| E^1 \|_\infty \le K_1 \| E^0 \|_\infty$$

Suppose that

$$\| E^q \|_\infty \le K_1 \| E^0 \|_\infty,$$

for $q \le n$ and K_1 is independent of h and k.

Using Lemma (2.2), we have $2 - b_1 > 0$, $b_{j-1} - 2b_j > 0$, $2b_n - b_{n-1} > 0$. Consider,

$$| \varepsilon_i^{n+1} | \le | \varepsilon_i^{n+1} | - r\sum_{l=0}^{i+1} w_l | \varepsilon_{i-l+1}^{n+1} |$$

$$\le | \varepsilon_i^{n+1} - r\sum_{l=0}^{i+1} w_l \varepsilon_{i-l+1}^{n+1} |$$

$$\le | 2\varepsilon_i^n - \varepsilon_i^{n-1} - \sum_{l=0}^{n-1} b_j (\varepsilon_i^{n-j+1} - 2\varepsilon_i^{n-j} + \varepsilon_i^{n-j-1}) - 2b_n(\varepsilon_i^1 - \varepsilon_i^0) + r\sum_{l=0}^{i+1} w_l \varepsilon_{i-l+1}^n |$$

$$\le | (2 - b_1)\varepsilon_i^n + \sum_{l=0}^{n-1} (2b_j - b_{j-1} - b_{j+1})\varepsilon_i^{n-j} - b_n \varepsilon_i^1 + (2b_n - b_{n-1})\varepsilon_i^0$$

$$+ r\sum_{l=0}^{i+1} w_l \varepsilon_{i-l+1}^n |$$

$$\leq (2 - b_1)|\,\varepsilon_i^n\,| + \sum_{j=1}^{n-1}(b_{j-1} + b_{j+1} - 2b_j)|\,\varepsilon_i^{n-j}\,| + b_n|\,\varepsilon_i^1\,| + (2b_n - b_{n-1})|\,\varepsilon_i^0\,|$$

$$+ r\sum_{l=0}^{i+1} w_l|\,\varepsilon_{i-l+1}^n\,|$$

$$\leq \left(2 - b_1 + \sum_{j=1}^{n-1}(b_{j-1} + b_{j+1} - 2b_j) + b_n + 2b_n - b_{n-1} + r\sum_{l=0}^{i+1} w_l\right)|\,\varepsilon^0\,|$$

$$\leq \left(2(1 - b_1) + 2(2b_n - b_{n-1}) + r\sum_{l=0}^{i+1} w_l\right)|\,\varepsilon^0\,| \leq K_1|\,\varepsilon^0\,|$$

Therefore, $\| E^{n+1} \|_\infty \leq K_1 \| E^0 \|_\infty$, where K_1 is a positive constant independent of h and k. Hence, by mathematical induction, for all $n = 1, 2, \ldots, N$, we have

$$\| E^n \|_\infty \leq K_1 \| E^0 \|_\infty$$

This completes the proof. □

4. Convergence

In this section, we discuss the question of convergence. Let $\overline{V}_i^n$ be the exact solution of space-time fractional traveling wave equation (1.1)-(1.3) and τ_i^n be the local truncation error for $1 \leq i \leq M$. Then, from (2.6)-(2.9), we have

$$\tau_i^1 = \overline{V}_i^1 - \frac{r}{2}\sum_{l=0}^{i+1} w_l \overline{V}_{i-l+1}^1 - \overline{V}_i^0 - kg(x_i) - \frac{r}{2}\sum_{l=0}^{i+1} w_l \overline{V}_{i-l+1}^0 = O(h + k) \tag{4.1}$$

and for $1 \leq n \leq N - 1$,

$$\tau_i^{n+1} = \overline{V}_i^{n+1} - r\sum_{l=0}^{i+1} w_l \overline{V}_{i-l+1}^{n+1} - 2\overline{V}_i^n + \overline{V}_i^{n-1} + \sum_{j=1}^{n-1} b_j(\overline{V}_i^{n-j+1} - 2\overline{V}_i^{n-j} + \overline{V}_i^{n-j-1})$$

$$+2b_n(\overline{V}_i^1 - \overline{V}_i^0 - kg(x_i)) - r\sum_{l=0}^{i+1} w_l \overline{V}_{i-l+1}^n = O(h+k) \tag{4.2}$$

Theorem 4.1. *Let* $\overline{V}_i^n$ *be the exact solution of* (1.1)-(1.3) *and* $\overline{V}_i^n$ *be the numerical solution of finite difference scheme* (2.6)-(2.9) *at each mesh point* (x_i, t_n). *Then there exist a positive constant* K_2 *independent of h and k such that*

$$\| \overline{V}_i^n - V_i^n \| \leq K_2(h+k), 1 \leq n \leq N.$$

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 $e^n = (e_1^n, e_2^n, \ldots, e_{M-1}^n)^T$ for $1 \leq n \leq N$ and local truncation error vector by $\tau^n = (\tau_1^n, \tau_2^n, \ldots, \tau_{M-1}^n)^T$ for time level n. From equations (4.1)-(4.2), we get

$$e_i^1 - \frac{r}{2}\sum_{l=0}^{i+1} w_l e_{i-l+1}^1 = e_i^0 + \frac{r}{2}\sum_{l=0}^{i+1} w_l e_{i-l+1}^0 + \tau_i^1 \tag{4.3}$$

for $n \geq 1$,

$$e_i^{n+1} - r\sum_{l=0}^{i+1} w_l e_{i-l+1}^{n+1} = 2e_i^n - e_i^{n-1} - \sum_{j=1}^{n-1} b_j(e_i^{n-j+1} - 2e_i^{n-j} + e_i^{n-j-1})$$

$$+2b_n(e_i^1 - e_i^0) + r\sum_{l=0}^{i+1} w_l e_{i-l+1}^n + \tau_i^{n+1}. \tag{4.4}$$

Using mathematical induction, we will prove that $\| e^n \|_\infty \leq K_2(h+k)$. For $n = 1$, we have

$$| e_i^1 | \leq | e_i^1 | - \frac{r}{2}\sum_{l=0}^{i+1} w_l | e_{i-l+1}^1 |$$

$$\leq | e_i^1 - \frac{r}{2}\sum_{l=0}^{i+1} w_l e_{i-l+1}^1 |$$

$$\leq | e_i^0 - \frac{r}{2}\sum_{l=0}^{i+1} w_l e_{i-l+1}^0 + \tau_i^1 | \leq \left[1 + \frac{r}{2}\sum_{l=0}^{i+1} w_l\right] | e_i^0 | + | \tau_i^1 | \leq | \tau_i^1 | \leq K_2(h+k)$$

Therefore, $\| e^1 \|_\infty \leq K_2(h+k)$, where K_2 is independent of h and k. Suppose that

$$\| e^q \|_\infty \leq K_2(h+k),$$

for $q \leq n$ and K_2 is independent of h and k. Consider,

$$| e_i^{n+1} | \leq | e_i^{n+1} | - r\sum_{l=0}^{i+1} w_l | e_{i-l+1}^{n+1} |$$

$$\leq | e_i^{n+1} - r\sum_{l=0}^{i+1} w_l e_{i-l+1}^{n+1} |$$

$$\leq | 2e_i^n - e_i^{n-1} - \sum_{l=0}^{n-1} b_j (e_i^{n-j+1} - 2e_i^{n-j} + e_i^{n-j-1}) - 2b_n(e_i^1 - e_i^0)$$

$$+ r\sum_{l=0}^{i+1} w_l e_{i-l+1}^n + \tau_i^{n+1} |$$

$$\leq | (2-b_1)e_i^n + \sum_{l=0}^{n-1} (2b_j - b_{j-1} - b_{j+1}) e_i^{n-j} - b_n e_i^1 + (2b_n - b_{n-1}) e_i^0$$

$$+ r\sum_{l=0}^{i+1} w_l e_{i-l+1}^n + \tau_i^{n+1} |$$

$$\leq (2-b_1) | e_i^n | + \sum_{j=1}^{n-1} (b_{j-1} + b_{j+1} - 2b_j) | e_i^{n-j} | + b_n | e_i^1 | + r\sum_{l=0}^{i+1} w_l | e_{i-l+1}^n | + | \tau_i^{n+1} |$$

$$\leq \left(2 - b_1 + \sum_{j=1}^{n-1} (b_{j-1} + b_{j+1} - 2b_j) + b_n + r\sum_{l=0}^{i+1} w_l \right) K_2'(h+k) + | \tau_i^{n+1} |$$

$$\leq \left(2(1 - b_1) + 2(2b_n - b_{n-1}) + r\sum_{l=0}^{i+1} w_l \right) K_2'(h+k) + | \tau_i^{n+1} | \leq K_2(h+k)$$

Therefore, $\| e^{n+1} \|_\infty \leq K_2(h+k)$.

Hence, by mathematical induction, for all $n = 1, 2, \ldots, N,$ we have

$$\| e^n \|_\infty \leq K_2(h+k)$$

This completes the proof. □

5. Python Programme

In this section, we develop the Python programme-CN for Crank-Nicolson finite difference scheme (2.6)-(2.9) to solve space-time fractional traveling wave equation (1.1)-(1.3) numerically. We compute $\overline{V}_i^n$ at each grid point (x_i, t_n) using the scheme (2.6)-(2.9). The algorithm is given below:

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

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

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

Now, we develop the Python programme-CN for complete discretized scheme (2.6)-(2.9) as follows:

Inputs:

f - initial displacement

g - initial velocity

C - velocity of wave

L - spatial length

T - end time

h - space step size

k - temporal step size

a - fractional order α of time derivative

b - fractional order β of space derivative

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

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

```
import math

import numpy as np

import scipy.linalg

def CN(f,g,C,T,L,a,b,h,k,t1):

r=(C**2*math.gamma(3-a)*k**a)/(2*h**b)

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])

A1 = np.zeros((M-1, M-1))

A2 = np.zeros((M-1, M-1))

b1 = np.zeros(M-1)

b2 = np.zeros(M-1)

w = np.zeros(M+1)

w[0]=1

for l in range(1,M+1):
```

```
w[l]=w[l-1]*(1-((1+b)/l))
for i in range(0,M-1):
A1[i][i]=1-(r/2)*w[1]
for i in range(0,M-2):
A1[i][i+1]=-(r/2)*w[0]
for i in range(1,M-1):
for j in range(0,i):
A1[i][j]=-(r/2)*w[i-j+1]
for i in range(1,M):
s=0
for l in range(0,i+2):
s=s+w[l]*V[0][i-l+1]
b1[i-1]=V[0][i]+k*g(x[i])+(r/2)*s
V[1][1:M]=scipy.linalg.solve(A1, b1)
V[1][0]=0;V[1][M]=0
for i in range(0,M-1):
A2[i][i]=1-r*w[1]
for i in range(0,M-2):
A2[i][i+1]=-r*w[0]
for i in range(1,M-1):
for j in range(0,i):
A2[i][j]=-r*w[i-j+1]
for n in range(1,N):
for i in range(1,M):
s1,s2=0,0
for j in range(1,n):
```

```
s1=s1+((j+1)**(2-a)-j**(2-a))*(V[n-j+1][i]-2*V[n-j][i]+V[n-j-1][i])
s1=s1+2*((n+1)**(2-a)-(n)**(2-a))*(V[1][i]-V[0][i]-k*g(x[i]))
for l in range(0,i+2):
s2=s2+w[l]*V[n][i-l+1]
b2[i-1]=2*V[n][i]-V[n-1][i]-s1+r*s2
V[n+1][1:M]=scipy.linalg.solve(A2, b2)
V[n+1][0]=0;V[n+1][M]=0
t1=int(t1/k)
return(x,V[t1])
```

Numerical experiment 1. We consider the following space-time fractional traveling wave equation:

$$\frac{\partial^{\alpha} V}{\partial t^{\alpha}} = C^2 \frac{\partial^{\beta} V}{\partial x^{\beta}}, \ (x, t) \in \Omega = [0, 1] \times [0, 1] \tag{5.1}$$

with initial conditions:

$$V(x, 0) = \sin(2\pi x), \ \frac{\partial}{\partial t} V(x, 0) = 0, \ x \in [0, 1] \tag{5.2}$$

and boundary conditions,

$$V(0, t) = 0, \ V(1, t) = 0, \ t \in (0, 1] \tag{5.3}$$

The exact solution to this problem for $\alpha = 2$, $\beta = 2$ and $C = 1$ as follows:

$$V(x, t) = \sin(2\pi x) \cos(2\pi t)$$

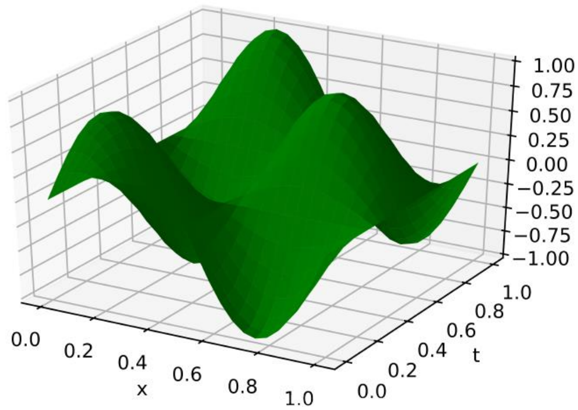

Figure 1. Periodic solution of traveling wave equation.

Using the python programme-CN, we estimate the value of $V(x, t)$ for any time level t_n. Let $\varepsilon(h, k)$ be the maximum error between exact and numerical solutions with temporal and spatial grid sizes k and h respectively. The temporal and spatial order of convergence are computed using

$$\text{temporal order } = \log_2\left(\frac{\varepsilon(h, 2k)}{\varepsilon(h, k)}\right), \text{ spatial order } = \log_2\left(\frac{\varepsilon(2h, k)}{\varepsilon(h, k)}\right).$$

In Table 1, we obtain the maximum error and order of convergence in temporal direction at time $t = 1$ with $h = 2^{-10}$.

Table 1. Maximum errors and temporal orders of convergence at $t = 1$, $h = 2^{-10}$.

k	Maximum error	Order
2^{-5}	0.264489	–
2^{-6}	0.142758	0.89
2^{-7}	0.074186	0.94
2^{-8}	0.037816	0.97
2^{-9}	0.019091	0.98
2^{-10}	0.009592	0.99

In Table 2, we obtain the maximum error and order of convergence in spatial direction at

Table 2. Maximum errors and spatial orders of convergence at $x = 0.9999$, $k = 2^{-10}$.

h	Maximum error	Order
2^{-2}	0.999371	–
2^{-3}	0.706478	0.50
2^{-4}	0.382055	0.88
2^{-5}	0.194462	0.97
2^{-6}	0.097388	0.99
2^{-7}	0.048439	1.00

$x = 0.9999$ with $k = 2^{-10}$.

From Table 1 and 2, we observe that the proposed finite difference scheme is first-order accurate in temporal as well as spatial direction. The order of convergence obtained in the numerical results agreed to the theoretical analysis. In Figure 2, we compare the exact and numerical solutions obtained by the Crank-Nicolson scheme and observe that the numerical solution is enormously agreed with the exact solution.

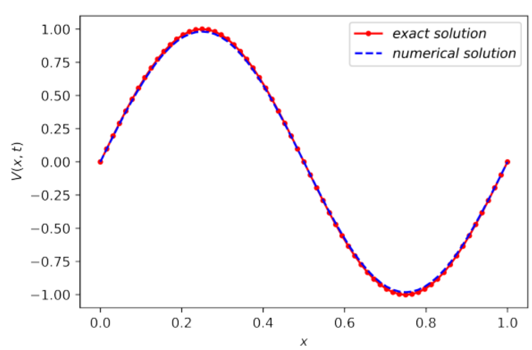

Figure 2. Comparison between exact and the numerical solutions with the parameters $h = 2^{-6}$, $k = 2^{-9}$, $t = 1$, $C = 1$.

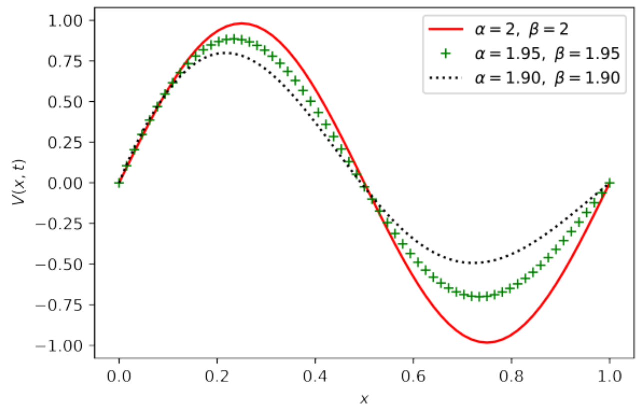

Figure 3. Comparison of the numerical solutions with the parameters $h = 2^{-6}$, $k = 2^{-9}$, $t = 1$, $C = 1$.

From Figure 3, we observed that the obtained solutions are stable and sufficiently approximate to the exact solutions and therefore, we conclude that the proposed scheme gives accurate results and stable solutions. Hence, Python is a powerful tool to obtain the numerical solutions of space-time fractional traveling wave equation.

Numerical experiment 2. We consider the following space-time fractional traveling wave equation:

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

subject to initial conditions:

$$V(x, 0) = 0, \ \frac{\partial}{\partial t} V(x, 0) = 2\pi C \sin(2\pi x), \ x \in (0, 1]$$

and boundary conditions,

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

The exact solution to this problem for $\alpha = 2, \beta = 2$ is $V(x, t) = \sin(2\pi x)\sin(2C\pi t)$. In Table 3 and 4, we obtain the order of convergence in temporal and spatial directions respectively.

Table 3. Maximum errors and temporal orders of convergence at $t = 0.75$, $h = 2^{-8}$.

k	Maximum error	Order
2^{-6}	0.106548	–
2^{-7}	0.055491	0.94
2^{-8}	0.028306	0.97
2^{-9}	0.014285	0.98
2^{-10}	0.007166	0.99
2^{-11}	0.003580	1.00

Table 4. Maximum errors and spatial orders of convergence at $x = 0.9999$, $k = 2^{-10}$.

h	Maximum error	Order
2^{-2}	1.107705	–
2^{-3}	0.723289	0.61
2^{-4}	0.383607	0.91
2^{-5}	0.194308	0.98
2^{-6}	0.097193	0.99
2^{-7}	0.048326	1.00

From these tables, it can be seen that the proposed finite difference scheme is first-order accurate in temporal as well as spatial direction.

In Figure 4, we obtain the numerical solutions using proposed finite difference scheme for different values of t for $\alpha = 1.9$, $\beta = 1.8$.

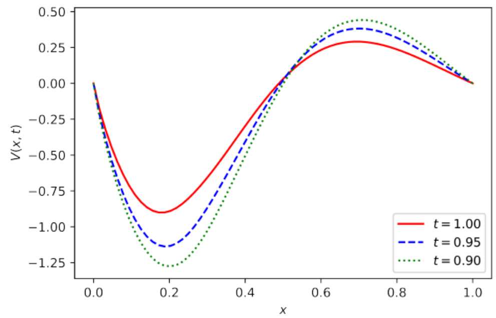

Figure 4. Behavior of the numerical solutions with the parameters $\alpha = 1.9$, $\beta = 1.8$, $h = 2^{-6}$, $k = 2^{-9}$, $C = 1$.

From Figure 4, we observe that solutions obtained by proposed scheme are stable and converges appropriately to the solution obtained at $t = 1$. In Figure 5, we obtain the numerical solutions for different values of α and β at $t = 0.7$.

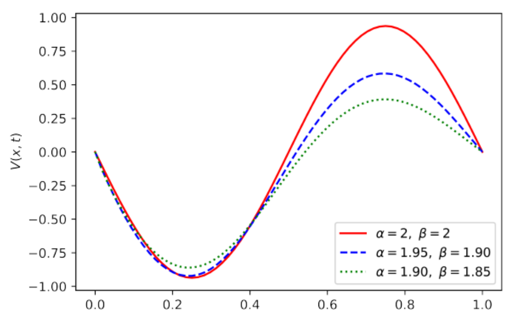

Figure 5. Behavior of the numerical solutions with the parameters $h = 2^{-6}$, $k = 2^{-9}$, $C = 1$, $t = 0.7$.

We observe that solutions obtained by proposed scheme are converges to the solution obtained for $\alpha = 2$, $\beta = 2$.

6. Conclusions

(i) We develop the Crank-Nicolson finite difference scheme for space-time fractional traveling wave equation.

(ii) Furthermore, we proved that the developed scheme is unconditionally stable and convergent.

(iii) We successfully develop a python programme for space-time fractional traveling wave equation and obtain the numerical solutions of the test problems and estimate the error.

(iv) Also, we found that the finite difference scheme is numerically stable and the results are compatible with our theoretical analysis.

(v) Finally, we conclude that Python is a powerful tool for obtaining the numerical solutions of space-time fractional traveling wave equation because the numerical results are very close to the exact solutions.

References

[1] Elham Afshari, Behnam Sepehrian and Ali Mohamad Nazari, Finite difference method for solving the space-time fractional wave equation in the Caputo form, Fractional Differential Calculus 1 (2015), 55-63.

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

[3] Suryakant Jogdand, Sharvari Kulkarni and Kalyanrao Takale, On fractional order crank-nicolson finite difference scheme for space-time fractional convection diffusion equation, International Journal of Research and Analytical Reviews 6(1) (2019), 50-55.

[4] Hans Petter Langtangen and Svein Linge, Finite Difference Computing with PDEs: A Modern Software Approach, volume 16 of Texts in Computational Science and Engineering, Springer International Publishing, Cham, 2017.

[5] Jichun Li and Yi-Tung Chen, Computational partial differential equations using MATLAB. 2020. OCLC: 1127853350.

[6] Yuri Luchko, Fractional wave equation and damped waves, Journal of Mathematical Physics 54(3) (2013), 031505.

[7] Sam Morley, Applying math with python: practical recipes for solving computational math problems using Python programming and its libraries, Packt Publishing, Birmingham Mumbai, 2020.

[8] Igor 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.

[9] Arsen Pskhu and Sergo Rekhviashvili, Fractional Diffusion {Wave Equation with Application in Electrodynamics, Mathematics 8(11) (2020) 20-86.

[10] Fathalla A. Rihan, Numerical Modeling of Fractional-Order Biological Systems, Abstract and Applied Analysis 1 (2013), 1-11.

[11] Enrico Scalas, Rudolf Goreno and Francesco Mainardi, Fractional calculus and continuous-time finance, Physica A: Statistical Mechanics and its Applications 284(1-4) (2000), 376-384.

[12] Karl A. Seeler, Finite Difference Methods and MATLAB, In System Dynamics, Springer New York, New York, NY (2014), 467-517.

[13] G. W. Shrimangale, T. L. Holambe and K. C. Takale, Fractional order crank-nicolson finite difference scheme for time-space fractional radon diffusion equation in charcoal medium, International Journal of Research and Analytical Reviews 6(1) (2019), 831-836.

[14] G. D. Smith, Numerical solution of partial differential equations: finite difference methods, Oxford applied mathematics and computing science series, Clarendon Press; Oxford University Press, Oxford [Oxfordshire] : New York, 3rd ed. edition, 1985.

[15] Roelof J. Stroeker and Johan F. Kaashoek, Discovering Mathematics with Maple, Birkhauser Basel, Basel, 1999.

[16] K. C. Takale, V. R. Nikam and S. R. Kulkarni, Fractional order finite difference scheme for space fractional diffusion equation, Engineering and Automation Problems 3 (2014), 120-124.

[17] Yanfei Wang, Yaxin Ning and Yibo Wang, Fractional Time Derivative Seismic Wave Equation Modeling for Natural Gas Hydrate, Energies 13(22) (2020), 5901.

[18] Abdul-Majid Wazwaz, Partial differential equations and solitary waves theory, Nonlinear physical science, Higher Education Press Springer, Beijing Berlin, 2009.