POLİTEKNİK DERGİSİ

JOURNAL of POLYTECHNIC

ISSN: 1302-0900 (PRINT), ISSN: 2147-9429 (ONLINE)
URL: http://dergipark.gov.tr/politeknik

A new perspective on the numerical solution for fractional Klein Gordon

Kesirli mertebeden Klein Gordon denkleminin nümerik çözümleri üzerine yeni bir bakış açısı

***Yazar(lar) (Author(s))**: Berat KARAAGAC[1], Yusuf UCAR[2], N. Murat YAGMURLU[3], Alaattin ESEN[4]*

ORCID[1]: 0000-0002-6020-3243

ORCID[2]: 0000-0003-1469-5002

ORCID[3]: 0000-0003-1593-0254

ORCID[4]: 0000-0002-7927-5941

Bu makaleye şu şekilde atıfta bulunabilirsiniz(To cite to this article): Karaagac B., Ucar Y, Yagmurlu N. M. ve Esen A., "A new perspective on the numerical solution for fractional Klein Gordon", ***Politeknik Dergisi***, 22(2): 443-451, (2019).

Erişim linki (To link to this article): http://dergipark.gov.tr/politeknik/archive

DOI: **10.2339/politeknik.428986**

Kesirli Mertebeden Klein Gordon Denkleminin Nümerik Çözümleri Üzerine Yeni Bir Bakış Açısı

Araştırma Makalesi / Research Article

Berat KARAAGAC[1], Yusuf UCAR[2], N. Murat YAGMURLU[2], Alaattin ESEN[2]

[1]Faculty of Education, Departmant of Mathematics Education, Adıyaman University, Turkey
[2]Faculty of Art and Science, Departmant of Mathematics, Inonu University, Turkey

(Geliş/Received :06.02.2018; Kabul/Accepted :27.05.2018)

ÖZ

Sunulan çalışmada, zamana göre kesirli mertebeden türevli lineer olmayan Klein Gordon denklemini çözmek için yeni bir nümerik şema sunuldu. Kesirli mertebeden denklemin yaklaşık çözümleri kübik B-spline kollokasyon sonlu eleman yöntemi ve L2 algortimasına dayanmaktadır. Denklemde verilen kesirli türev ise Caputo anlamında ele alınmıştır. Yöntemler kullanılarak, kesirli mertebeden diferansiyel denklem bilgisayar kodlamasına elverişli cebirsel denklem sistemine dönüştürülür. Daha sonra, amaçlanan yöntemin güvenilirliğini ve etkisini göstermek amacı ile iki model problem ele alındı ve hata normları hesaplandı. Yeni hesaplanan hata normları saysısal çözümlerin tam çözümlerle uyum içinde olduğunu göstermektedir.

Anahtar Kelimeler: Sonlu eleman yöntemi, kollokasyon, kesirli mertebeden Klein Gordon denklemi, Caputo türevi.

A New Perspective on The Numerical Solution for Fractional Klein Gordon Equation

ABSTRACT

In the present manuscript, a new numerical scheme is presented for solving the time fractional nonlinear Klein-Gordon equation. The approximate solutions of the fractional equation are based on cubic B-spline collocation finite element method and L2 algorithm. The fractional derivative in the given equation is handled in terms of Caputo sense. Using the methods, fractional differential equation is converted into algebraic equation system that are appropriate for computer coding. Then, two model problems are considered and their error norms are calculated to demonstrate the reliability and efficiency of the proposed method. The newly calculated error norms show that numerical results are in a good agreement with the exact solutions.

Keywords: Finite element method, collocation, Fractional Klein Gordon equation, Caputo derivative.

1. INTRODUCTION

Fractional differential equations own a deep history and also rich theory. Its past is as long as classical calculus and up to date since 1695. Over the years, many mathematician and physicist have been attracted by fractional calculus because of its wide application areas, longterm memory and chaotic behaviour such as physics, biology, finance, fluid dynamics, engineering etc. The development and obtaining numerical and exact solutions of the equations, containing fractional derivative and integral, have gained great and significant importance. So, various methods have been investigated for this purpose. Among others,some of them are [1, 2, 3, 4, 5, 6, 7].

In this study, we are going to concern with obtaining numerical solutions of time fractional Klein Gordon equation in terms of Caputo sense derivative which is one of the fundamental equations seen in fractional calculus.

The mathematical expression of the equation is given as

$$D_t^{\alpha}u(x,t)+au_{xx}(x,t)+bu(x,t)+cu^{\beta}(x,t)u(x,t)=f(x,t) \quad (1)$$

subject to the following initial and boundary conditions

$$u(x,0)=u_0(x),\quad u_t(x,0)=u_1(x)$$
$$u(0,t)=h_0(x),\quad u(1,t)=h_1(x) \quad (2)$$

where $D_t^{\alpha}(.)$ symbolizes α_{th} order fractional derivative according to time variable and the range of α is $(1,2]$. $f(x,t)$ is a known forced term and in addition to these terms a,b,c and β are real constants and also c can seen as a variable coefficients in some examples. For $\alpha=2$, we get the classical Klein Gordon equation which appears in classical relativistic and quantum mechanics and analysing of wave propagation in linear dispersive media. Additionally, The fractional Klein–Gordon equation has many application in nonlocal quantum field theory and stochastic quantization of nonlocal fields [8]

The equation has been solved by several authors using different methods and techniques. Among others, Nagy [7] has solved the problem using a method consisting of

*Sorumlu Yazar (Corresponding Author)
e-posta : bkaraagac@adiyaman.edu.tr

expanding the required approximate solution as the elements of Sinc functions along the space direction and shifted Chebyshev polynomials of the second kind for the time variable. Kheri *et. al.* [9] have solved inhomogenous fractional Klein-Gordon equation by the method of separating variables and applied the method for three boundary conditions.

Mohebbi *et al.* [10] have applied a high-order difference scheme for the solution of some time fractional partial differential equations including linear time fractional Klein-Gordon and dissipative Klein-Gordon equations. Lyu and Vong [11] have considered difference schemes for nonlinear timefractional Klein-Gordon type equations. Khader *et al.* [12] have implemented the Chebyshev spectral method for solving the non-linear fractional Klein-Gordon equation and considered the fractional derivative in the Caputo sense. Alqahtani [13] has implemented the spectral collocation method with the help of the Legendre polynomials for solving the non-linear Fractional (Caputo sense) Klein-Gordon Equation. For all that, recent developments in computational methods are lead to improving new numerical methods for solving fractional or ordinary order partial differential equations. One can receive more information about newly research in Refs [14,15,16,17,] and therein.

The manuscript consists of four parts. The first part presents an introduction to the model problem and some research papers on it. The second one covers application of cubic B-spline collocation method to the problem and obtaining numerical formulation. Two different examples of time fractional Klein Gordon equation and their numerical results are considered in the third part for different values of constants and forced term. The last one is conclusion

2. Application of collocation cubic B-spline FEM method to the time fractional Klein Gordon equation

In this part of paper, we are going to obtain numerical solution for the fractional Klein Gordon equation with the help of finite element formulation and cubic spline basis. At first, we discretize the problem in time for fractional derivative with the help of $L2$ finite difference approximation. We consider t_j as the grid points for time and Δt are grid size. So time discretization for $t_j = t_0 + j\Delta t\left(j = 0,1,2,\cdots,n\right)$ is obtain as following [18]

$$D_t^{\alpha} f\left(t\right) = \frac{\left(\Delta t\right)^{-\alpha}}{\Gamma\left(3-\alpha\right)} \sum_{k=0}^{n} w^{(1-\alpha)} \left[f\left(t_{n+1-k}\right) - 2 f\left(t_{n-k}\right) + f\left(t_{n-1-k}\right) \right] \tag{3}$$

where $w^{(1-\alpha)} = \left(k+1\right)^{(1-\alpha)} - k^{(1-\alpha)}$. $\Gamma(.)$ symbolizes Euler Gamma function and n is the time step as taken $n = \frac{t_{final}}{\Delta t}$. Before construction of numerical scheme, let us divide the interval $\left[0,1\right]$ into N subinterval using $\left\{x_i\right\}_{i=0}^{N}$ nodal points such that

$$0 = x_0 < x_1 < \cdots < x_{N-1} < x_N = 1$$

and symbolize each element as a typical element with step size $h = \left[x_i, x_{i+1}\right]$. Our aim is to develop an approximate solution for $u\left(x,t\right)$ as it will be $U_N\left(x,t\right)$ and are used to be linear combination of cubic B-spline basis $\left(\phi_i\left(x\right)\right)$ and time dependent element shape functions $\left(\delta\left(t\right)\right)$ in the form

$$U_N\left(x,t\right) = \sum_{m=-1}^{N+1} \delta_m\left(t\right)\phi_m\left(x\right). \tag{4}$$

In order to define all spline basis in same typical element $\left[x_i, x_{i+1}\right]$ and create a systematic procedure for numerical scheme, it is apparent the cubic B-spline basis required a local transformation coordinate instead of global coordinates. So we should use transformation $\xi = x - x_i$ $0 \le \xi \le 1$. After the transformation, cubic B-spline basis functions are defined as [19] follows

$$\begin{aligned}
&\phi_{m-1} = \frac{\left(h-\xi\right)^3}{h^3},\\
&\phi_{m=} \frac{h^3 + 3h^2\left(h-\xi\right) + 3h\left(h-\xi\right)^2 - 3\left(h-\xi\right)^3}{h^3},\\
&\phi_{m+1=} \frac{h^3 + 3h^2\xi + 3h\xi^2 - 3\xi^3}{h^3},\\
&\phi_{m+2=} \frac{\xi^3}{h^3}.
\end{aligned} \tag{5}$$

Moreover the approximate solution can be written in terms of the basis given in (5) as

$$U_N\left(x,t\right) = \sum_{i=m-1}^{m+2} \delta\left(t\right)\phi_i\left(\xi\right). \tag{6}$$

The nodal values of U_N and $U_N^{'}$ at the points x_i are derived using (2) as

$$\begin{aligned}
&U_N = \delta_{m-1}(t) + 4\delta_m(t) + \delta_{m+1}(t),\\
&U_N^{'} = \frac{3}{h}(\delta_{m+1}(t) - \delta_{m-1}(t),)\\
&U_N^{''} = \frac{6}{h^2}(\delta_{m-1}(t) - 2\delta_m(t) + \delta_{m+1}(t)).
\end{aligned} \tag{7}$$

First all of , we will start to progress with a simple linearization choosing $zm = u_N$. Then substituting (2) into Eq.(1) and using (7), we have

$$\begin{aligned}
&D_t^{\alpha}(\delta_{m-1}\left(t\right) + 4\delta_m\left(t\right) + \delta_{m+1}\left(t\right))\\
&+ \frac{6a}{h^2}\left(\delta_{m-1}\left(t\right) - 2\delta_m\left(t\right) + \delta_{m+1}\left(t\right)\right)\\
&+ \left(b + czm^{\beta}\right)(\delta_{m-1}\left(t\right) + 4\delta_m\left(t\right) + \delta_{m+1}\left(t\right)) = f\left(x,t\right).
\end{aligned} \tag{8}$$

For

going on to obtain numerical scheme, time dependent element shape functions $\delta(t)$ s' are discretized using $L2$ algorithm given in (1.3), forward difference and crank-Nicolson formula as

$$D_t^{\alpha}\delta(t) = \frac{(\Delta t)^{-\alpha}}{\Gamma(3-\alpha)}\sum_{k=0}^{n} w^{(1-\alpha)}\left[\delta^{n+1-k}(t) - 2\delta^{n-k}(t) + f\,\delta^{n-1-k}(t)\right] \quad (9)$$

$$\delta(t) = \frac{\delta^{n+1}(t)+\delta^{n}(t)}{2}, \quad \dot{\delta}(t) = \frac{\delta^{n+1}(t)-\delta^{n}(t)}{\Delta t}.$$

At the end, after some calculation and simplification we we get a algebraic equation system consisting $(n+1)_{th}$ time step unknown $\delta^{n+1}(t)$ parameters and n_{th} time step known $\delta^{n}(t)$ parameters following form

$$\begin{aligned}
&\delta_{m-1}^{n+1}(t)\left(1+\frac{3aS}{h^2}+\frac{(b+czm^{\beta})S}{2}\right)+\delta_{m}^{n+1}(t)\left(4-\frac{6aS}{h^2}+2(b+czm^{\beta})S\right)\\
&+\delta_{m+1}^{n+1}(t)\left(1+\frac{3as}{h^2}+\frac{(b+czm^{\beta})S}{2}\right)=\delta_{m-1}^{n}(t)\left(2-\frac{3aS}{h^2}-\frac{(b+czm^{\beta})S}{2}\right)\\
&+\delta_{m}^{n}(t)\left(8+\frac{6aS}{h^2}-2(b+czm^{\beta})S\right)+\delta_{m+1}^{n}(t)\left(2-\frac{3aS}{h^2}-\frac{(b+czm^{\beta})S}{2}\right)\\
&-\left(\delta_{m-1}^{n-1}(t)+4\delta_{m}^{n-1}(t)+\delta_{m+1}^{n-1}(t)\right)\\
&-\sum_{k=1}^{n} w^{(1-\alpha)}\left\{\left(\delta_{m-1}^{n+1-k}(t)-2\delta_{m-1}^{n-k}(t)+\delta_{m-1}^{n-1-k}(t)\right)\right.\\
&+4\left(\delta_{m}^{n+1-k}(t)-2\delta_{m}^{n-k}(t)+\delta_{m}^{n-1-k}(t)\right)\\
&\left.+\delta_{m+1}^{n+1-k}(t)-2\delta_{m+1}^{n-k}(t)+\delta_{m+1}^{n-1-k}(t)\right\}+Sf(x,t)
\end{aligned} \quad (10)$$

Where $S=\Gamma(3-\alpha)(\Delta t)^{\alpha}$ and $zm=\delta_{m-1}(t)+4\delta_{m}(t)+\delta_{m+1}(t)$. Now we get a system consisting of $(N+1)$ equations and $(N+3)$ unknown variables. Eq. (10) is valid for only interior nodal points so to obtain unique solution one must apply boundary conditions given in (2) to numerical scheme. For this purpose, we employ $u(0,t)=\delta_{-1}(t)+4\delta_{0}(t)+\delta_{1}(t)=h_0(0)$ for $m=0$ and $u(1,t)=\delta_{N-1}(t)+4\delta_{N}(t)+\delta_{N+1}(t)=h_1(1)$ for $m=N$. So eliminating $\delta_{-1}(t)$ and $\delta_{N+1}(t)$ from the system we get $(N+1)\times(N+1)$ system of equations. At the last , we have a iterative system. Now, we need an initial vector for begining iteration, so one can obtain $\delta^{n}(0)$ parameters easily by using initial conditions as $u(x_m,0)=U_N(x_m,0)=u_0(x_m)$. If this is written clearly, we get

$$\begin{aligned}
&\delta_{-1}(0)+4\delta_{0}(0)+\delta_{1}(0)=u_0(x_0)\\
&\delta_{0}(0)+4\delta_{1}(0)+\delta_{2}(0)=u_0(x_1)\\
&\vdots\\
&\delta_{N-2}(0)+4\delta_{N-1}(0)+\delta_{N}(0)=u_0(x_{N-1})\\
&\delta_{N-1}(0)+4\delta_{N}(0)+\delta_{N+1}(0)=u_0(x_N)
\end{aligned} \quad (11)$$

as seen from the Eq. (11), there exist $(N+1)$ equations and $(N+3)$ unknown variables. $\delta_{-1}(0)$ and $\delta_{N+1}(0)$ parameters can be eliminated using $u(0,0)=U_N(0,0)=\delta_{-1}(0)+4\delta_0(0)+\delta_1(0)$ and $u(1,0)=U_N(1,0)=\delta_{N-1}(0)+4\delta_N(0)+\delta_{N+1}(0)$. this system can now be solved with any algorithm and iteration can be started.

3. Numerical Tests for Time Fractional Klein Gordon Equation

In the third section of the manuscript, we are going to demonstrate efficiency and applicability of numerical method using two test problems. For two examples, since the exact solutions of the examples are known, we are going to calculate error norms L_2 and L_∞ using the definition given as below;

$$L_2 = \|u-U_N\|_2 = \sqrt{h\sum_{j=0}^{N}\left|u_j-(U_N)_j\right|^2},$$

$$L_\infty = \|u-U_N\|_\infty = \max_{0\le j\le N}\left|u_j-(U_N)_j\right|,$$

where u and U_N represent exact and numerical solutions, respectively. And the order of convergence is calculated with the following formula;

$$order = \frac{Log\left(\frac{(L_\infty)_{old}}{(L_\infty)new}\right)}{Log\left(\frac{h_{old}}{h_{new}}\right)}.$$ Thus, we are able to compare exact ones with numerical ones.

3.1. Example 1:

In the following first numerical experiment, we have taken time fractional Klein-Gordon equation with the values of the coefficients as $a=-1, b=0, c=1, \beta=1$ for $0\le x\le 1$ and $t_{final}=1$. So we can rewrite the equation given in (1) with the forced term, as follows

$$\begin{aligned}
D_t^{\alpha}u-u_{xx}+u^2 &= \frac{\Gamma(5/2)}{\Gamma(5/2-\alpha)}(1-x)^{5/2}t^{(3/2-\alpha)}\\
&-\frac{15}{4}(1-x)^{1/2}t^{3/2}+(1-x)^5t^3
\end{aligned} \quad (12)$$

and with non-zero right boundary condition

$$\begin{aligned}
&u(x,0)=0, \quad u_t(x,0)=0\\
&u(0,t)=t^{3/2}, \quad u(1,t)=0
\end{aligned} \quad (13)$$

exact solution of the example is given as

$$u(x,t)=(1-x)^{5/2}t^{3/2}. \quad (14)$$

First of all, the results are calculated for various space step (h) and time step (Δt) size for $\alpha=1.3$. Then the error norms L_2 and L_∞ are presented in Tables 1 and 2, respectively. In addition, error norms and orders are

reported in Table 3 for $\alpha = 1.7$ and $\Delta t = 0.00001$. It is seen from Tables 1, 2 and 3, when the number of time step size are the same, to increase number of collocation points lead to a decrease in the error norms. Additionally, for the collocation finite element method, time step sizes as important as collocation points. So one can see, decreasing of time step sizes results decreasing in the error norms.

Table 1. A representation of the $L_2 \times 10^3$ norm for various values of Δt and h for $\alpha = 1.3$.

Δt	0.01	0.005	0.001	0.0005	0.0001	0.00005	0.00001
h							
0.25	5.461326	4.731241	4.162181	4.092810	4.037694	4.030666	4.022183
0.125	2.504028	1.724493	1.123959	1.052339	0.996010	0.989010	0.982798
0.1	2.166491	1.375058	0.762329	0.689667	0.632882	0.625871	0.619889
0.05	1.744344	0.937212	0.297653	0.220406	0.161032	0.153902	0.148171
0.025	1.648449	0.838967	0.192712	0.112499	0.049669	0.042201	0.036389
0.0125	1.625804	0.815966	0.168899	0.088131	0.023748	0.015858	0.009760
0.01	1.623192	0.813319	0.166198	0.085399	0.020871	0.012899	0.006696

Table 2. A representation of the $L_\infty \times 10^3$ norm for various values of Δt and h for $\alpha = 1.3$

Δt	0.01	0.005	0.001	0.0005	0.0001	0.00005	0.00001
h							
0.25	7.700108	6.972540	6.394705	6.323344	6.267288	6.260548	6.252998
0.125	3.245500	2.246167	1.569895	1.494217	1.456844	1.452790	1.449505
0.1	2.861628	1.789100	1.034280	0.970713	0.920066	0.913785	0.908673
0.05	2.382455	1.266362	0.385704	0.286827	0.224010	0.217614	0.212481
0.025	2.276125	1.154699	0.259479	0.148555	0.064919	0.056718	0.051215
0.0125	2.250264	1.128478	0.232098	0.120229	0.031186	0.020518	0.013229
0.01	2.247337	1.125460	0.229031	0.117111	0.027784	0.016821	0.008900

Table 3. The error norms and orders for various h values for $\Delta t = 0.0001, 0.00001$ for $\alpha = 1.7$

Δt	0.0001			0.00001		
h	$L_2 \times 10^3$	$L_\infty \times 10^3$	$order(L_\infty)$	$L_2 \times 10^3$	$L_\infty \times 10^3$	$order(L_\infty)$
0.25	3.531801	5.748447	–	3.734496	6.148587	––
0.125	0.874420	1.372834	2.06602	0.898053	1.334451	2.20401
0.1	0.561053	0.845099	2.17429	0.567403	0.818875	2.18847
0.05	0.156823	0.217845	1.95582	0.138186	0.194232	2.07586
0.025	0.063585	0.083314	1.38667	0.035906	0.048160	2.01187
0.0125	0.042662	0.058626	0.507018	0.011576	0.015588	1.6274
0.01	0.040366	0.056028	0.203128	0.008835	0.012140	1.120034

Table 4. A comparison of absolute errors for $\Delta t = 0.00001, \alpha = 1.5, 1.9$ and $N = 10$.

	$\alpha = 1.5$		$\alpha = 1.9$	
x	[7] ($m = n = 6$)	Present method ($N = 10$)	[7] ($m = n = 6$)	Present method ($N = 10$)
0.1	1.9004×10^{-3}	1.065601×10^{-4}	1.7145×10^{-3}	2.3961723×10^{-3}
0.2	2.0752×10^{-3}	2.479456×10^{-4}	8.3897×10^{-4}	2.7676947×10^{-3}
0.3	2.0682×10^{-3}	3.840470×10^{-4}	5.9801×10^{-5}	1.265934×10^{-4}
0.4	1.8787×10^{-3}	5.107724×10^{-4}	4.2370×10^{-4}	2.8074307×10^{-3}
0.5	1.6102×10^{-3}	6.424084×10^{-4}	7.4383×10^{-4}	1.5788475×10^{-3}
0.6	1.4483×10^{-3}	7.617908×10^{-4}	8.5920×10^{-4}	6.606390×10^{-4}
0.7	1.5545×10^{-3}	8.574570×10^{-4}	5.6969×10^{-4}	1.4651589×10^{-3}
0.8	1.6959×10^{-3}	9.138149×10^{-4}	8.9237×10^{-4}	8.686224×10^{-4}
0.9	1.4757×10^{-3}	8.828905×10^{-4}	7.4789×10^{-5}	1.982566×10^{-4}
L_2		6.18386×10^{-4}		1.648307×10^{-3}
L_∞		9.13815×10^{-4}		2.807431×10^{-3}

Tables 4, 5 and 6 compare absolute errors between three different methods. Tables 4 is consist of a comparisons between [7] and present method for $\Delta t = 0.00001, \alpha = 1.5, 1.9$ and $N = 10, 20$, respectively. other one is between [7], VRM and present method for $\Delta t = 0.00001, \alpha = 1.4, 1.6$ and $N = 10, 20$. Finally, we tabulated some comparisons of absolute errors and relative errors for $\Delta t = 0.00001, \alpha = 1.6, 1.8$ and N = 20 in Table 6 and 7. We can conclude that newly obtained results are more convergent for the $\alpha = 1.3$, 1.5, and 1.6 and agree with for $\alpha = 1.9$ when partition number is chosen as $N = 10$. Also applied method has more converges results for all α values when partition number is chosen as $N > 10$

Table 5. A comparison of absolute errors for $\Delta t = 0.00001, \alpha = 1.5, 1.9$ and N = 20

	$\alpha = 1.5$		$\alpha = 1.9$	
x	[7] ($m = n = 9$)	Present method (N = 20)	[7] ($m = n = 9$)	Present method (N = 20)
0.1	8.7105×10^{-4}	2.29860×10^{-5}	4.3675×10^{-4}	3.547459×10^{-4}
0.2	6.7781×10^{-4}	6.14772×10^{-5}	9.8359×10^{-5}	8.937207×10^{-4}
0.3	6.2089×10^{-4}	9.13921×10^{-5}	4.8897×10^{-4}	4.322902×10^{-4}
0.4	5.7015×10^{-4}	1.228082×10^{-4}	7.6534×10^{-4}	4.290707×10^{-4}
0.5	5.1476×10^{-4}	1.538698×10^{-4}	9.3043×10^{-4}	2.275352×10^{-4}
0.6	4.8948×10^{-4}	1.811331×10^{-4}	9.4248×10^{-4}	3.7590×10^{-7}
0.7	5.1671×10^{-4}	2.023530×10^{-4}	7.5585×10^{-4}	1.286623×10^{-4}
0.8	5.3919×10^{-4}	2.136602×10^{-4}	4.2006×10^{-5}	2.716377×10^{-4}
0.9	6.0660×10^{-4}	2.038887×10^{-4}	5.4848×10^{-5}	2.267357×10^{-4}
L_2		1.478850×10^{-4}		4.50187×10^{-4}
L_∞		2.13660×10^{-4}		9.10231×10^{-4}

Table 6. A comparision of absolute errors for $\Delta t = 0.00001, \alpha = 1.4, 1.6$ and $N = 10, 20$

α	(x,t)			$N = 10$	$N = 20$
	$(0.1, 0.1)$	9.2852×10^{-3}	8.4385×10^{-4}	4.5158×10^{-6}	7.484×10^{-7}
	$(0.2, 0.2)$	2.2201×10^{-2}	1.1433×10^{-3}	5.4722×10^{-6}	1.2260×10^{-6}
	$(0.3, 0.3)$	3.5651×10^{-2}	5.3780×10^{-4}	1.75791×10^{-5}	4.5398×10^{-6}
1.4	$(0.4, 0.4)$	4.9628×10^{-2}	1.5545×10^{-4}	6.16598×10^{-5}	1.52821×10^{-5}
	$(0.5, 0.5)$	6.4449×10^{-2}	5.3227×10^{-4}	1.487415×10^{-4}	3.61098×10^{-5}
	$(0.6, 0.6)$	7.9514×10^{-2}	1.3268×10^{-3}	2.808198×10^{-4}	6.71903×10^{-5}
	$(0.7, 0.7)$	9.1443×10^{-2}	1.9159×10^{-3}	4.465890×10^{-4}	1.056595×10^{-4}
	$(0.8, 0.8)$	8.7942×10^{-2}	2.0414×10^{-3}	6.226345×10^{-4}	1.457116×10^{-6}
	$(0.9, 0.9)$	9.2321×10^{-4}	1.8996×10^{-3}	7.445275×10^{-4}	1.719402×10^{-4}
	$(0.1, 0.1)$	4.1518×10^{-3}	1.1685×10^{-3}	6.93424×10^{-5}	1.50576×10^{-5}
	$(0.2, 0.2)$	1.0319×10^{-2}	2.5887×10^{-3}	4.24945×10^{-5}	8.8257×10^{-6}
	$(0.3, 0.3)$	1.7757×10^{-2}	2.8863×10^{-3}	3.96546×10^{-5}	8.6607×10^{-6}
1.6	$(0.4, 0.4)$	2.6987×10^{-2}	2.3912×10^{-3}	5.38836×10^{-5}	1.26185×10^{-5}
	$(0.5, 0.5)$	3.8327×10^{-2}	1.7692×10^{-3}	1.316628×10^{-4}	3.09618×10^{-5}
	$(0.6, 0.6)$	5.0993×10^{-2}	1.4174×10^{-3}	2.538026×10^{-4}	6.06844×10^{-5}
	$(0.7, 0.7)$	6.1379×10^{-2}	1.4334×10^{-3}	4.144158×10^{-4}	9.79853×10^{-5}
	$(0.8, 0.8)$	5.6577×10^{-2}	1.6653×10^{-3}	5.952975×10^{-4}	1.395405×10^{-4}
	$(0.9, 0.9)$	3.8618×10^{-2}	1.7449×10^{-3}	7.282169×10^{-4}	1.685685×10^{-4}

Table 7. A comparision of absolute errors for $\Delta t = 0.00001, \alpha = 1.6$ and $N = 20$.

$\alpha = 1.6$					
		[7] (m=9)		**Present Method**	
t	x	Absolute error	Relative error	Absolute error	Relative error
0.4	0.4	9.3726×10^{-4}	1.3286×10^{-2}	1.26185×10^{-5}	1.789×10^{-4}
	0.6	9.4592×10^{-4}	3.6950×10^{-2}	3.02423×10^{-5}	1.1813×10^{-3}
	0.8	6.5448×10^{-4}	1.4462×10^{-1}	4.52139×10^{-5}	9.9910×10^{-3}
0.8	0.4	1.7359×10^{-4}	8.6999×10^{-4}	7.66008×10^{-5}	3.839×10^{-4}
	0.6	1.2080×10^{-4}	1.6683×10^{-3}	1.075323×10^{-4}	1.4851×10^{-3}
	0.8	2.4657×10^{-4}	1.9263×10^{-2}	1.395405×10^{-4}	1.09016×10^{-2}

Table 8. A comparision of absolute errors for $\Delta t = 0.00001, \alpha = 1.8$ and $N = 20$

$\alpha = 1.8$					
		[7] (m=9)		**Present Method**	
t	x	Absolute error	Relative error	Absolute error	Relative error
0.4	0.4	3.4329×10^{-3}	4.8663×10^{-2}	1.22000031×10^{-2}	17.29385×10^{-2}
	0.6	3.4391×10^{-3}	1.3434×10^{-1}	7.7639662×10^{-3}	30.32794×10^{-2}
	0.8	2.1582×10^{-3}	4.7690×10^{-1}	2.8061378×10^{-3}	62.00712×10^{-2}
0.8	0.4	6.9075×10^{-4}	3.4618×10^{-3}	1.22414817×10^{-2}	6.13509×10^{-2}
	0.6	6.5491×10^{-4}	9.0448×10^{-3}	1.22552370×10^{-2}	16.92529×10^{-2}
	0.8	5.7248×10^{-4}	4.4725×10^{-2}	8.1618789×10^{-3}	63.76431×10^{-2}

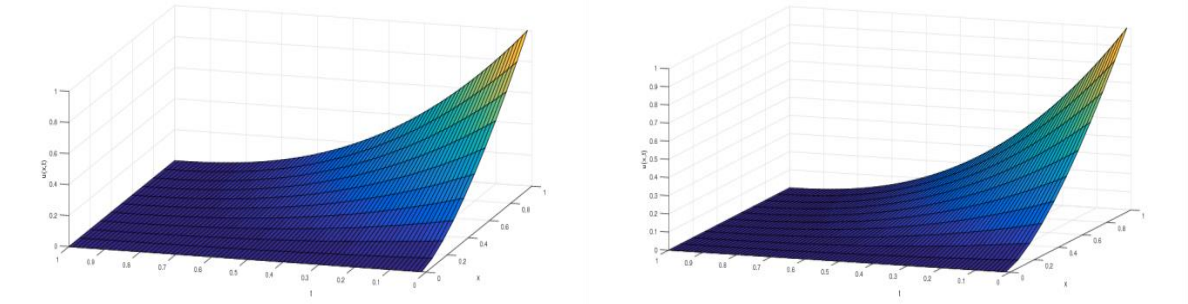

Figure 1. The numerical solutions of Time Fractional Klein Gordon equation for $\alpha = 1.3$ and 1.5.

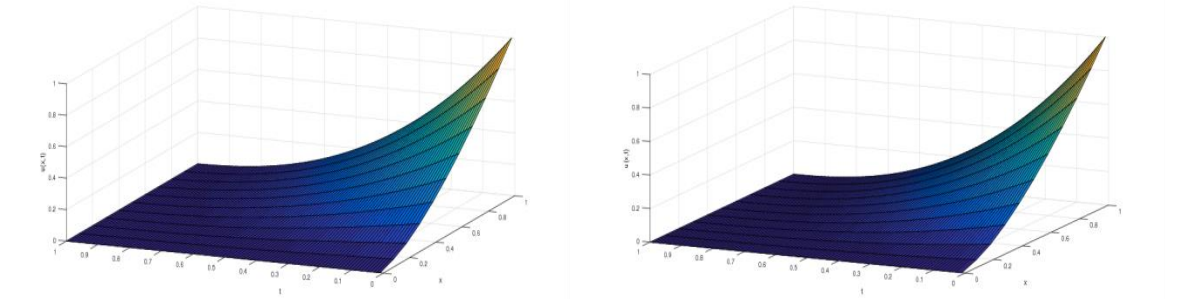

Figure 2. The numerical solutions of Time Fractional Klein Gordon equation for $\alpha = 1.8$ and 1.9.

3.2 Example 2:

In our second example, weassume that $a = -1, b = 0, c = 2.5e^x, \beta = 3/2$ and the forced term is

$$f(x,t) = \frac{\Gamma(3+\alpha)}{2} x^3 (1-x)^3 t^2 + \left(30x^4 - 60x^3 + 36x^2 - 6x\right) t^{2+\alpha} + 2.5e^x x^{4.5} (1-x)^{4.5} t^{3+1.5\alpha}$$

with the boundary conditions and the exact solution is

$$u(x,0)=0,\quad u_t(x,0)=0$$
$$u(0,t)=0,\quad u(1,t)=0$$
$$u(x,t)=x^3(1-x)^3 t^{2+\alpha}$$

In the considering numerical experiment, we present a series of table according to error norms L_2 and L_∞. Table 6- 9 consist of error norms results for various values of time step Δt. and space step h for $\alpha=1.3$ and 1.5, respectively. It can be seen from the tables, the most convergent results are found for the values $h=0.01$ and $\Delta t=0.00001$.

Numerical results have same behaviour as first example i.e decreasing in time and space step size ends up with decreasing in the error norms. At the end of this part, numerical simulations of example 2 are depicted for the various choosing of α parameters in Figures 1 and 2.

Table 9. A representation of the $L_2\times10^3$ norm for various values of Δt and h for $\alpha=1.3$.

Δt	0.01	0.005	0.001	0.0005	0.0001	0.00005	0.00001
h							
0.25	0.916627	0.916267	0.916166	0.916172	0.916182	0.916184	0.942900
0.125	0.271518	0.271585	0.271579	0.271572	0.271566	0.275610	0.275609
0.1	0.175499	0.175609	0.175612	0.175604	0.175596	0.175594	0.187200
0.05	0.044173	0.044332	0.044348	0.044338	0.044327	0.045241	0.062025
0.025	0.010973	0.011106	0.011124	0.011113	0.030162	0.011342	0.011340
0.0125	0.002817	0.002784	0.002801	0.002789	0.002777	0.002837	0.002835
0.01	0.001940	0.001788	0.001803	0.175604	0.001778	0.001817	0.032412

Table 10. A representation of the $L_\infty\times10^3$ norm for various values of Δt and h for $\alpha=1.3$.

Δt	0.01	0.005	0.001	0.0005	0.0001	0.00005	0.00001
h							
0.25	1.178927	1.177854	1.177550	1.177568	1.177597	1.177603	1.236302
0.125	0.416998	0.416427	0.416247	0.416252	0.416265	0.419592	0.419595
0.1	0.256970	0.256507	0.256359	0.256363	0.256373	0.256375	0.282494
0.05	0.062948	0.062482	0.062332	0.062336	0.062346	0.063598	0.111092
0.025	0.016361	0.015783	0.015599	0.015604	0.053928	0.016170	0.016159
0.0125	0.004637	0.004063	0.003995	0.003950	0.003910	0.004070	0.004059
0.01	0.003248	0.002660	0.002603	0.002558	0.002500	0.002612	0.051111

Table 11. A representation of the $L_2\times10^3$ norm for various values of Δt and h for $\alpha=1.5$.

Δt	0.01	0.005	0.001	0.0005	0.0001	0.00005	0.00001
h							
0.25	0.933111	0.933327	0.933555	0.916172	0.916182	0.916184	0.916186
0.125	0.281918	0.281729	0.281544	0.271572	0.271566	0.271565	0.271564
0.1	0.187800	0.187536	0.187273	0.175604	0.175596	0.175594	0.175593
0.05	0.063206	0.062689	0.062171	0.044338	0.044327	0.044325	0.044323
0.025	0.038371	0.037691	0.036999	0.011113	0.011102	0.011100	0.011098
0.0125	0.034415	0.033709	0.032989	0.002789	0.002777	0.002775	0.002773
0.01	0.002590	0.001788	0.032617	0.001791	0.001778	0.001776	0.001775

Table 12. A representation of the $L_\infty\times10^3$ norm for various values of Δt and h for $\alpha=1.5$.

Δt	0.01	0.005	0.001	0.0005	0.0001	0.00005	0.00001
h							
0.25	1.213214	1.213791	1.214409	1.177568	1.177597	1.177603	1.177608
0.125	0.406493	0.406761	0.407052	0.416252	0.416265	0.416267	0.416270
0.1	0.284914	0.283863	0.282795	0.256363	0.256373	0.256375	0.256377
0.05	0.113572	0.112493	0.111400	0.062336	0.062346	0.062348	0.062349
0.025	0.066852	0.065765	0.064665	0.015604	0.015616	0.015619	0.015621
0.0125	0.055051	0.053957	0.052853	0.003950	0.003910	0.003912	0.003914
0.01	0.004670	0.002660	0.051422	0.002558	0.002500	0.002501	0.002503

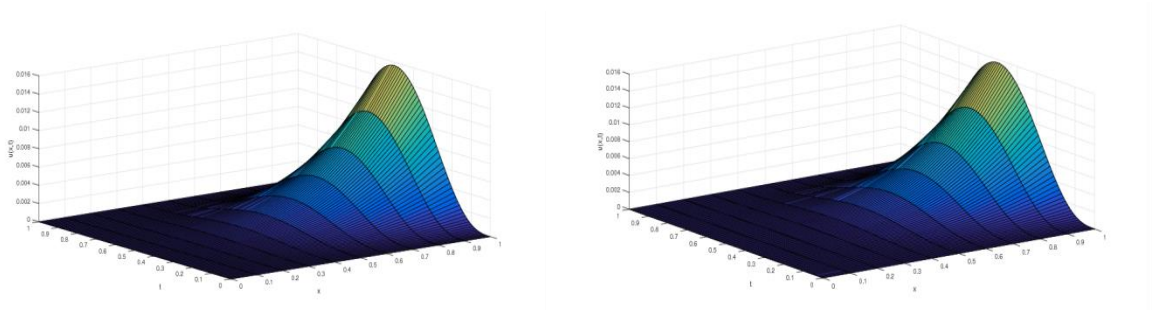

Figure 3. The numerical solutions of Time Fractional Klein Gordon equation given in example 2 for $\alpha = 1.3$ and 1.5.

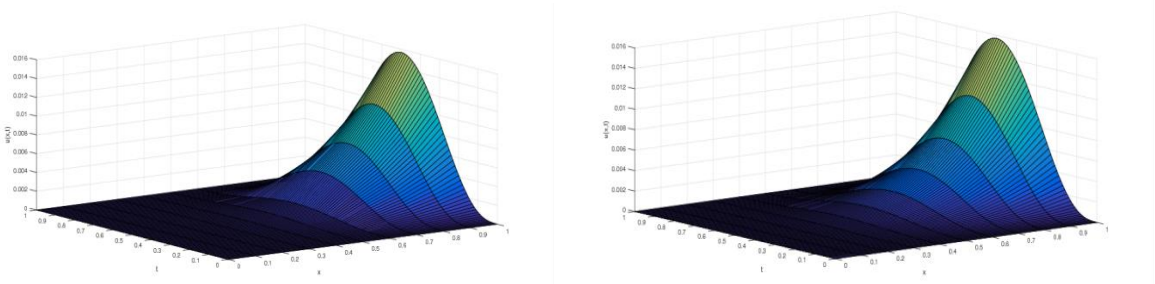

Figure 4. The numerical solutions of Time Fractional Klein Gordon equation given in example 2 for $\alpha = 1.8$ and 1.9.

4. CONCLUSION

In conclusion, this study has introduced to obtain numerical approximations using the finite element method. We have started with the application of the method to the time fractional Klein Gordon equation. Then, the problem has converted into ordinary differential equation system with the help of cubic B-spline basis and element shape parameters. Using mathematical coding, the newly obtained numerical scheme is solved with an iterative method uses an initial vector to purpose of generate approximate solution for two test problem. Additionally, the newly numerical results with the calculation have shown that has a reasonable agreement with exact ones. After all the numerical experiments examined in this paper, we can conclude that finite element collocation method and L2 algorithm can be a useful tool for obtaining numerical solutions of wide variety problems on fractional order partial differential equations.

REFERENCES

[1] Kumara D., Seadawy A. R. and Joardare A. K., "Modified Kudryashov method via new exact solutions for some conformable fractional differential equations arising in mathematical biology", ***Chinese Journal of Physics***, 56: 75–85, (2018).

[2] Kumara D., Seadawy A. R. and Joardare A. K., "Modified Kudryashov method via new exact solutions for some conformable fractional differential equations arising in mathematical biology", ***Chinese Journal of Physics***, 56: 75–85, (2018).

[3] Sohail A., Maqbool K. and Ellahi R**.,** "Stability analysis for fractional-order partial differential equations by means of space spectral time Adams-Bashforth Moulton method", ***Numerical solution for Partial Differential Equations***, 34: 19–29, (2018).

[4] Singh J., Kumar D., Hammouch Z. And Atangana A.," A fractional epidemiological model for computer viruses pertaining to a new fractional derivative", ***Applied Mathematics and Computation***, 316: 504–515,(2018).

[5] Tasbozan O. and Esen A.," Quadratic B-Spline Galerkin Method for Numerical Solutions of Fractional Telegraph Equations" ***Bulletin of Mathematical Sciences and Applications***, 18: 23-39, (2017).

[6] Bulut F., Oruç, Ö. And Esen A.," Numerical Solutions of Fractional System of Partial Differential Equations By Haar Wavelets" ***Computer Modeling in Engineering & Sciences,*** 108 :263-284,(2015).

[7] Nagy A. M., "Numerical solution of time fractional nonlinear Klein–Gordon equation using Sinc–Chebyshev collocation method", ***Applied Mathematics and Computation***, 310:139-148,(2017).

[8] Lim S. C. and S. V. Muniandy. "Stochastic quantization of nonlocal fields." ***Physics Letters A*** 324: 396-405(2004).

[9] Kheiri H., Shahi S. and Mojaver A., "Analytical solutions for the fractional Klein-Gordon equation", ***Computational Methods for Differential Equations***, 2:99-114,(2014).

[10] Mohebbi A., Abbaszadeh M. and Dehghan M., "High-Order Difference Scheme for the Solution of Linear Time Fractional Klein–Gordon Equations, Numerical solution for Partial Differential Equations"***Numerical solution for Partial Differential Equations,*** 30: 1234-1253.(2014).

[11] Lyu P. and Vong S., "A linearized second-order scheme for nonlinear time fractional Klein-Gordon type equations", ***Numerical Algorithms,*** 78:485-511,(2018).

[12] Khader M. M., Swetlam N. H . and Mahdy A. M. S., "The Chebyshev Collection Method for Solving Fractional Order Klein-Gordon Equation", ***Wseas Trans. Math***, 13: 31-38,(2014).

[13] Alqahtani R. T., "Approximate Solution of Non-Linear Fractional Klein-Gordon Equation Using Spectral Collocation Method", ***Applied Mathematic*s**, 6: 2175-2181,(2015).

[14] Li C., and Chen A., "Numerical methods for fractional partial differential equations." ***International Journal of Computer Mathematics*** 95: 1048-1099, (2018).

[15] Ara A., Khan N. A., "Wavelets optimization method for evaluation of fractional partial differential equations: an application to financial modelling." ***Advances in Difference Equations*** 2018.1:8. (2018)

[16] Liu Y. , Khan M., and Yan Y.,"Fourier spectral methods for stochastic space fractional partial differential equations driven by special additive noises." ***Journal of Computational Analysis and Applications*** 24: 290-309, (2018).

[17] Baseri, A., Abbasbandy S., and Babolian E., "A collocation method for fractional diffusion equation in a long time with Chebyshev functions." ***Applied Mathematics and Computation*** 322:55-65, (2018)

[18] Yuste S. B**.,** "Weighted average finite difference methods for fractional diffusion equations", **Journal of Computational Physics,** 216:264-274 ,(2006).

[19] Prenter P. M., "***Splines and Variational Methods"***, Wiley, (1975)