Appl. Math. Inf. Sci. **16**, No. 2, 331-339 (2022)

Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.18576/amis/160221

A High Order Accurate Numerical Solution of the Klein-Gordon Equation

Dursun Irk[1,*], *Emre Kirli*[2] *and Melis Zorsahin Gorgulu*[1]

[1]Department of Mathematics and Computer Science, Eskisehir Osmangazi University, 26040, Eskisehir, Turkey
[2]Department of Mathematics, Istanbul Bilgi University, 34440, Istanbul, Turkey

Received: 18 Jun. 2020, Revised: 30 Sep. 2021, Accepted: 20 Dec. 2021
Published online: 1 Mar. 2022

Abstract: In this paper, numerical solution of the nonlinear Klein-Gordon equation is obtained by using the cubic B-spline Galerkin method for space discretization and the finite difference method which is of order four for time discretization. Accuracy of the method is presented by computing the maximum error norm. Robustness of the suggested method is shown by studying some classical test problems.

Keywords: Cubic B-spline functions, Galerkin method, Klein-Gordon equation, finite difference method

1 Introduction

The Klein-Gordon equation (KGE) which has an important historic role in the formulation of relativistic quantum mechanics [1] is of the form:

$$u_{tt} + \alpha u_{xx} + G(u) = f(x,t) \tag{1}$$

with the following conditions

$$\begin{aligned} u(x,0) &= f_0(x), && (2)\\ u_t(x,0) &= f_1(x), \quad a \leq x \leq b, && (3)\\ u(a,t) &= g_0(t), \quad u(b,t) = g_1(t), && (4)\\ u_x(a,t) &= g_2(t), \quad u_x(b,t) = g_3(t), \quad t \geq 0 && (5) \end{aligned}$$

where $G(u)$ is the nonlinear force. By choosing the function $G(u)$ as $\beta u + \gamma u^k$, $k = 2,3$, the nonlinear KGE is given as

$$u_{tt} + \alpha u_{xx} + \beta u + \gamma u^k = f(x,t) \tag{6}$$

where β and γ are constants. The Eq. (6) is named as the quadratic nonlinear KGE for $k = 2$ and the cubic nonlinear KGE for $k = 3$. The KGE is known as the relativistic version of the Schrödinger equation and named after Oskar Klein and Walter Gordon [1]. The KGE arises in various physical events like as the interaction of solitons in a collisionless plasma, the motion of rigid pendula attached a stretched wire and examining the nonlinear waves, etc. So, the numerical solutions of KGE have been investigated using various numerical methods such as pseudospectral, decomposition, finite difference, finite element, He's variational iteration, radial basis function approximation, cubic B-spline collocation, differential quadrature, meshless, multiquadric Quasi-interpolation, Haar wavelet, exponential cubic B-spline collocation and Galerkin methods by many researchers such as Li and Guo [2], Duncan [3], Kaya and El-Sayed [4], Khalifa and Elgamal [5], Shakeri and Dehghan [6], Dehghan and Shokri [7], Rashidinia et al. [8], Bao and Dong [9], Verma et al.[10], Hussain et al. [11], Sarboland and Aminataei [12], Shira et al. [13], Ersoy et al. [14], Yang [15], Selvitopi and Yazici [16].

This paper's purpose is to present a numerical method to get the numerical solution of the nonlinear KGE by applying the Galerkin finite element method based on cubic B-spline functions for the space discretization of the KGE and a finite difference method which is of order four for the time discretization of the KGE. Using the fourth order finite difference method in time discretization of the KGE, it is aimed to increase the accuracy of the proposed numerical method.

The organization of this paper is as follows. First, the time and space discretizations of the KGE is described in Section 2. Then, three examples are given to investigate the efficiency of the proposed method, and a comparison

* Corresponding author e-mail: dirk@ogu.edu.tr

with the existed studies is made in Section 3. Finally, the conclusion is given in Section 4.

2 The Numerical Method

By choosing the term $u_t(x,t)$ in the KGE (6) is equal to $v(x,t)$, the nonlinear partial differential equation (1) can be rewritten as a system of partial differential equations

$$u_t = v, \tag{7}$$
$$v_t = -\alpha u_{xx} - \beta u - \gamma u^k + f(x,t) \tag{8}$$

with the boundary and initial conditions:

$$u(a,t) = g_0(t),\; u(b,t) = g_1(t),$$
$$v(a,t) = \frac{\partial g_0}{\partial t}(t),\; v(b,t) = \frac{\partial g_1}{\partial t}(t), \tag{9}$$
$$u_x(a,t) = g_2(t),\; u_x(b,t) = g_3(t),$$
$$v_x(a,t) = \frac{\partial g_2}{\partial t}(t),\; v_x(b,t) = \frac{\partial g_3}{\partial t}(t), \tag{10}$$
$$u(x,0) = f_0(x),\; v(x,0) = \frac{\partial f_0}{\partial t}(x). \tag{11}$$

Consider $\Omega = [a,b] \times [0,T]$ be smooth region with the grid points (x_m,t_n), where

$$x_m = a + mh,\; m = 0,1,2,\dots,N,\; t_n = n\Delta t,\; n = 0,1,2,\dots,$$

h and Δt are mesh size in the space and time direction respectively.

2.1 The Finite Difference Method

For the time discretization of the Eqs. (7) and (8) the following finite difference approximation has been employed:

$$u^{n+1} = u^n + \theta_1 u_t^{n+1} + \theta_2 u_t^n + \theta_3 u_{tt}^{n+1} + \theta_4 u_{tt}^n \tag{12}$$

and

$$v^{n+1} = v^n + \theta_1 v_t^{n+1} + \theta_2 v_t^n + \theta_3 v_{tt}^{n+1} + \theta_4 v_{tt}^n \tag{13}$$

where θ_1, θ_2, θ_3 and θ_4 are unknown parameters which will be defined later. By using the Eq. (7) into the Eq. (12), taking partial derivative with respect to t in the both sides of the Eq. (7) and substituting the Eq. (8) in related term, the following form is obtained:

$$\begin{aligned} u^{n+1} &= u^n + \theta_1 v^{n+1} + \theta_2 v^n + \theta_3(-\alpha u_{xx}^{n+1} - \beta u^{n+1} \\ &\quad -\gamma(u^{n+1})^k + f(x,t_{n+1})) \\ &\quad +\theta_4(-\alpha u_{xx}^n - \beta u^n - \gamma(u^n)^k + f(x,t_n)). \end{aligned} \tag{14}$$

After required simplifications, Eq. (14) is rewritten as

$$\begin{aligned} &u^{n+1}(1+\theta_3\beta+\theta_3\gamma(u^{n+1})^{k-1}) + \theta_3\alpha u_{xx}^{n+1} - \theta_1 v^{n+1} \\ &= u^n(1-\theta_4\beta-\theta_4\gamma(u^n)^{k-1}) - \theta_4\alpha u_{xx}^n + \theta_2 v^n \\ &\quad +\theta_4 f(x,t_n) + \theta_3 f(x,t_{n+1}). \end{aligned} \tag{15}$$

Also, by substituting the Eq. (8) into the Eq. (13), the following form is obtained:

$$\begin{aligned} &v^{n+1} - \theta_1(-\alpha u_{xx}^{n+1} - \beta u^{n+1} - \gamma(u^{n+1})^k + f(x,t_{n+1})) \\ &\quad -\theta_3 v_{tt}^{n+1} = v^n + \theta_2(-\alpha u_{xx}^n - \beta u^n - \gamma(u^n)^k \\ &\quad + f(x,t_n)) + \theta_4 v_{tt}^n \end{aligned} \tag{16}$$

and taking partial derivative with respect to t in the both sides of Eq. (8), we get

$$\begin{aligned} v_{tt} &= -\alpha(u_{xx})_t - \beta u_t - \gamma(u_t)^k + f_t(x,t) \\ &= -\alpha(u_t)_{xx} - \beta v - \gamma k(u^{k-1})u_t + f_t(x,t) \\ &= -\alpha v_{xx} - \beta v - \gamma k(u^{k-1})v + f_t(x,t). \end{aligned} \tag{17}$$

By substituting Eq. (17) into Eq. (16), we have

$$\begin{aligned} &u^{n+1}(\theta_1\beta + \theta_1\gamma(u^{n+1})^{k-1}) + \theta_1\alpha u_{xx}^{n+1} + \theta_3\alpha v_{xx}^{n+1} \\ &+v^{n+1}(1+\theta_3\beta+\theta_3\gamma k(u^{n+1})^{k-1}) \\ &= u^n(-\theta_2\beta - \theta_2\gamma(u^n)^{k-1}) - \theta_2\alpha u_{xx}^n + v^n(1-\theta_4\beta \\ &\quad -\theta_4\gamma k(u^n)^{k-1}) - \theta_4\alpha v_{xx}^n + \theta_1 f(x,t_{n+1}) \\ &\quad +\theta_2 f(x,t_n) + \theta_3 f_t(x,t_{n+1}) + \theta_4 f_t(x,t_n). \end{aligned} \tag{18}$$

Lemma 1. *Suppose $u,v,f \in C^6(\Omega)$ and $\theta_1 = \theta_2 = \frac{\Delta t}{2}$ and $\theta_3 = -\theta_4 = -\frac{(\Delta t)^2}{12}$. Then, the numerical scheme (15) and (18) are consistent and fourth order accurate in time for the norm $\|\cdot\|_\infty$.*

Proof. Using the $\theta_1 = \theta_2 = \frac{\Delta t}{2}$ and $\theta_3 = -\theta_4 = -\frac{(\Delta t)^2}{12}$ in (15) and (18), the truncation errors E_1 of (15) and E_2 of (18) are obtained for $k=2$ as

$$\begin{aligned} E_1(u,v,f) = \Big[&(\frac{\alpha\gamma}{180}v_{xx}(x,\tau) - \frac{\gamma}{360}f_t(x,\tau))u(x,\tau) \\ &+(\frac{\alpha\gamma}{90}u_{xx}(x,\tau) - \frac{\gamma}{120}f(x,\tau) \\ &+\frac{\gamma^2}{72}u^2(x,\tau) + \frac{\beta^2}{720} + \frac{\beta\gamma}{72}u(x,\tau))v(x,\tau) \\ &+\frac{\alpha\gamma}{180}u_x(x,\tau)v_x(x,\tau) + \frac{\alpha\beta}{360}v_{xx}(x,\tau) \\ &+\frac{\alpha^2}{720}v_{xxxx}(x,\tau) - \frac{\beta}{720}f_t(x,\tau) \\ &-\frac{\alpha}{720}f_{xxt}(x,\tau) + \frac{1}{720}f_{ttt}(x,\tau)\Big]\Delta t^5 + \dots, \end{aligned}$$

$$
\begin{aligned}
E_2(u,v,f) = \Big[& \big(-\frac{\alpha\gamma^2}{45}u_x^2(x,\tau) - \frac{\beta^3}{720} + \frac{\alpha\gamma}{180}f_{xx}(x,\tau) \\
& - \frac{\alpha^2\gamma}{120}u_{xxxx}(x,\tau) - \frac{\gamma}{360}f_{tt}(x,\tau) \\
& + \frac{\beta\gamma}{45}f(x,\tau)\big)u(x,\tau) + \big(\frac{\gamma^2}{36}u(x,\tau) \\
& + \frac{\beta\gamma}{72}\big)v^2(x,\tau) - \frac{\gamma}{90}f_t(x,\tau)v(x,\tau) \\
& - \frac{\gamma}{120}f^2(x,\tau) + \big(-\frac{\gamma^3}{72}u^2(x,\tau) - \frac{\beta\gamma^2}{36}u(x,\tau) \\
& + \frac{\gamma^2}{45}f(x,\tau) - \frac{11\beta^2\gamma}{720}\big)u^2(x,\tau) \\
& + \big(-\frac{\alpha\beta\gamma}{90}u_x(x,\tau) + \frac{\alpha\gamma}{180}f_x(x,\tau) \\
& - \frac{\alpha^2\gamma}{60}u_{xxx}(x,\tau)\big)u_x(x,\tau) + \big(\frac{7\alpha\gamma}{360}f(x,\tau) \\
& - \frac{13\alpha\beta\gamma}{360}u(x,\tau) - \frac{13\alpha\gamma^2}{360}u^2(x,\tau) \\
& - \frac{\alpha\beta^2}{240} - \frac{7\alpha^2\gamma}{360}u_{xx}(x,\tau)\big)u_{xx}(x,\tau) \\
& - \frac{\alpha^2\beta}{240}u_{xxxx}(x,\tau) - \frac{\alpha^3}{720}u_{xxxxxx}(x,\tau) \\
& + \frac{\alpha\gamma}{180}v_x^2(x,\tau) + \frac{\alpha\gamma}{60}v(x,\tau)v_{xx}(x,\tau) \\
& + \frac{\beta^2}{720}f(x,\tau) + \frac{\alpha\beta}{360}f_{xx}(x,\tau) \\
& + \frac{\alpha^2}{720}f_{xxxx}(x,\tau) - \frac{\alpha}{720}f_{xxtt}(x,\tau) \\
& - \frac{\beta}{720}f_{tt}(x,\tau) + \frac{1}{45}f_{tttt}(x,\tau)\Big]\Delta t^5 + \dots
\end{aligned}
$$

and for $k=3$ as

$$
\begin{aligned}
E_1(u,v,f) = \Big[& \big(-\frac{\gamma}{40}f(x,\tau) + \frac{\alpha\gamma}{30}u_{xx}(x,\tau)\big)u(x,\tau) \\
& + \big(-\frac{\gamma}{240}f_t(x,\tau) + \frac{\alpha\gamma}{120}v_{xx}(x,\tau)\big)u^2(x,\tau) \\
& + \big(\frac{3\gamma^2}{80}u^4(x,\tau) + \frac{\beta\gamma}{30}u^2(x,\tau) \\
& + \frac{\alpha\gamma}{120}u_x^2(x,\tau) + \frac{\beta^2}{720}\big)v(x,\tau) - \frac{\gamma}{120}v^3(x,\tau) \\
& + \frac{\alpha\beta}{360}v_{xx}(x,\tau) + \frac{\alpha^2}{720}v_{xxxx}(x,\tau) \\
& + \frac{\alpha\gamma}{60}u(x,\tau)u_x(x,\tau)v_x(x,\tau) - \frac{\beta}{720}f_t(x,\tau) \\
& + \frac{1}{720}f_{ttt}(x,\tau) - \frac{\alpha\beta}{720}f_{xxt}(x,\tau)\Big]\Delta t^5 + \dots,
\end{aligned}
$$

$$
\begin{aligned}
E_2(u,v,f) = \Big[& \big(-\frac{\gamma}{40}f^2(x,\tau) + \frac{\alpha\gamma}{60}f_x(x,\tau)u_x(x,\tau) \\
& - \frac{\beta^3}{720} - \frac{\alpha^2\gamma}{20}u_x(x,\tau)u_{xxx}(x,\tau) + \frac{\alpha\gamma}{60}v_x^2(x,\tau) \\
& - \frac{\alpha\beta\gamma}{24}u_x^2(x,\tau)\big)u(x,\tau) + \big(\frac{\alpha\gamma}{120}f_{xx}(x,\tau) \\
& - \frac{\alpha^2\gamma}{80}u_{xxxx}(x,\tau) + \frac{7\beta\gamma}{120}f(x,\tau) \\
& - \frac{\gamma}{240}f_{tt}(x,\tau)\big)u^2(x,\tau) + \big(-\frac{13\alpha\gamma^2}{120}u_x^2(x,\tau) \\
& - \frac{5\beta^2\gamma}{144}\big)u^3(x,\tau) + \frac{\alpha\gamma}{120}f(x,\tau)u_x^2(x,\tau) \\
& + \frac{\gamma^2}{16}f(x,\tau)u^4(x,\tau) - \frac{17\beta\gamma^2}{240}u^5(x,\tau) \\
& - \frac{3\gamma^3}{80}u^7(x,\tau) + \big(\frac{\alpha\gamma}{30}u_x(x,\tau)v_x(x,\tau) \\
& - \frac{\gamma}{30}f_t(x,\tau)u(x,\tau) + \frac{\alpha\gamma}{20}u(x,\tau)v_{xx}(x,\tau)\big)v(x,\tau) \\
& + \big(\frac{7\gamma^2}{40}u^3(x,\tau) + \frac{11\beta\gamma}{120}u(x,\tau) \\
& - \frac{\gamma}{20}f(x,\tau)\big)v^2(x,\tau) + \big(-\frac{7\alpha^2\gamma}{120}u_x^2(x,\tau) \\
& + \frac{7\alpha\gamma}{120}v^2(x,\tau) - \frac{23\alpha\gamma^2}{240}u^4(x,\tau) \\
& + \frac{7\alpha\gamma}{120}f(x,\tau)u(x,\tau) - \frac{\alpha\beta^2}{240} - \frac{\alpha\beta\gamma}{12}u^2(x,\tau) \\
& - \frac{7\alpha^2\gamma}{120}u(x,\tau)u_{xx}(x,\tau)\big)u_{xx}(x,\tau) \\
& - \frac{\alpha^2\beta}{240}u_{xxxx}(x,\tau) - \frac{\alpha^3}{720}u_{xxxxxx}(x,\tau) \\
& + \frac{\beta^2}{720}f(x,\tau) + \frac{\alpha\beta}{360}f_{xx}(x,\tau) \\
& + \frac{\alpha^2}{720}f_{xxxx}(x,\tau) - \frac{\alpha}{720}f_{xxtt}(x,\tau) \\
& - \frac{\beta}{720}f_{tt}(x,\tau) + \frac{1}{45}f_{tttt}(x,\tau)\Big]\Delta t^5 + \dots .
\end{aligned}
$$

where $\tau \in (t_n, t_{n+1})$. Hence, we have:

$$\|E_1(u,v,f)\|_\infty \le \Delta t^5 \sup_{(x,\xi)\in\Omega} |\varepsilon_1(x,\xi)| \text{ and}$$

$$\|E_2(u,v,f)\|_\infty \le \Delta t^5 \sup_{(x,\xi)\in\Omega} |\varepsilon_2(x,\xi)|$$

where $\varepsilon_i(x,\xi)$, $i=1,2$ denote the coefficients of the Δt^5 in $E_i(u(x,\xi), v(x,\xi), f(x,\xi))$. It follows that the proposed numerical scheme (15) and (18) are consistent and have order four in time.

Note that by choosing the parameters in Eqs. (12) and (13) as $\theta_1 = \theta_2 = \frac{\Delta t}{2}$ and $\theta_3 = -\theta_4 = 0$, we have Crank-Nicolson method which is of order two in time and by choosing the parameters in Eqs. (12) and (13) as $\theta_1 = \theta_2 = \frac{\Delta t}{2}$ and $\theta_3 = -\theta_4 = -\frac{(\Delta t)^2}{12}$, we have high order accurate which is of order four in time. Therefore our presented method for time disretization is a high order method.

2.2 The Finite Element Method

For space discretization of the KGE, the Galerkin finite element method has been applied to the Eqs. (15) and (18) as

$$\begin{aligned}&\int_a^b w(x)[u^{n+1}(1+\theta_3\beta+\theta_3\gamma(u^{k-1})^{n+1})+\theta_3\alpha u_{xx}^{n+1}\\&-\theta_1 v^{n+1}]dx=\int_a^b w(x)[u^n(1-\theta_4\beta-\theta_4\gamma(u^{k-1})^n)\\&-\theta_4\alpha u_{xx}^n+\theta_2 v^n+\theta_4 f(x,t_n)+\theta_3 f(x,t_{n+1})]dx,\end{aligned}\tag{19}$$

$$\begin{aligned}&\int_a^b w(x)[u^{n+1}(\theta_1\beta+\theta_1\gamma(u^{k-1})^{n+1})+\theta_1\alpha u_{xx}^{n+1}\\&+\theta_3\alpha v_{xx}^{n+1}+v^{n+1}(1+\theta_3\beta+\theta_3\gamma k(u^{k-1})^{n+1})]dx\\&=\int_a^b w(x)[u^n(-\theta_2\beta-\theta_2\gamma(u^{k-1})^n)-\theta_2\alpha u_{xx}^n\\&+v^n(1-\theta_4\beta-\theta_4\gamma k(u^{k-1})^n)-\theta_4\alpha v_{xx}^n]dx+\\&\int_a^b w(x)[\theta_1 f(x,t_{n+1})+\theta_2 f(x,t_n)+\theta_3 f_t(x,t_{n+1})\\&+\theta_4 f_t(x,t_n)]dx\end{aligned}\tag{20}$$

where w is a weight function.

The approximate solutions $U_N(x,t)$ and $V_N(x,t)$ are taken in terms of the cubic B-spline functions Q_m as

$$\begin{aligned}U_N(x,t)&=\sum_{m=-1}^{N+1}\delta_m(t)Q_m(x),\\V_N(x,t)&=\sum_{m=-1}^{N+1}\sigma_m(t)Q_m(x)\end{aligned}\tag{21}$$

where δ_m and σ_m, $m=-1,0,1,...,N+1$ are unknowns that are time dependent parameters to be determined by the discretized form of Eq. (1) and the cubic B-spline function for $m=-1,0,...,N+1$ is defined as follows:

$$Q_m(x)=\frac{1}{h^3}\begin{cases}(x-x_{m-2})^3, & x_{m-2}\le x<x_{m-1},\\ h^3+3h^2(x-x_{m-1}) & \\ +3h(x-x_{m-1})^2 & x_{m-1}\le x<x_m,\\ -3(x-x_{m-1})^3, & \\ h^3+3h^2(x_{m+1}-x) & \\ +3h(x_{m+1}-x)^2 & x_m\le x<x_{m+1},\\ -3(x_{m+1}-x)^3, & \\ (x_{m+2}-x)^3, & x_{m+1}\le x<x_{m+2},\\ 0, & \text{otherwise}\end{cases}\tag{22}$$

where the set of cubic B-spline $\{Q_{-1},Q_0,\dots,Q_{N+1}\}$ generates a basis over the solution domain $[a,b]$. The approximation functions, first and second derivatives of those over the element $[x_m,x_{m+1}]$ are given by using the cubic B-spline function as

$$\begin{aligned}U_N(x_m,t)&=\delta_{m-1}+4\delta_m+\delta_{m+1},\\U_N'(x_m,t)&=-\frac{3}{h}\delta_{m-1}+\frac{3}{h}\delta_{m+1},\\U_N''(x_m)&=\frac{6}{h^2}(\delta_{m-1}-2\delta_m+\delta_{m+1}),\\V_N(x_m,t)&=\sigma_{m-1}+4\sigma_m+\sigma_{m+1},\\V_N'(x_m,t)&=-\frac{3}{h}\sigma_{m-1}+\frac{3}{h}\sigma_{m+1},\\V_N''(x_m)&=\frac{6}{h^2}(\sigma_{m-1}-2\sigma_m-\sigma_{m+1}).\end{aligned}\tag{23}$$

By choosing the weight function w as cubic B-spline function, the Eqs. (19) and (20) are rewritten by using the Eq. (23) as

$$\begin{aligned}&\sum_{j=m-1}^{m+2}\Bigg\{(1+\theta_3\beta)\int_{x_m}^{x_{m+1}}Q_iQ_jdx\\&+\theta_3\gamma\int_{x_m}^{x_{m+1}}Q_i\left(\sum_{r=m-1}^{m+2}Q_r\delta_r^{n+1}\right)^{k-1}Q_jdx\\&+\theta_3\alpha\int_{x_m}^{x_{m+1}}Q_iQ_j''dx\Bigg\}\delta_j^{n+1}-\sum_{j=m-1}^{m+2}\left\{\theta_1\int_{x_m}^{x_{m+1}}Q_iQ_jdx\right\}\sigma_j^{n+1}\\&-\sum_{j=m-1}^{m+2}\Bigg\{(1-\theta_4\beta\int_{x_m}^{x_{m+1}}Q_iQ_jdx\\&-\theta_4\gamma\int_{x_m}^{x_{m+1}}Q_i\left(\sum_{r=m-1}^{m+2}Q_r\delta_r^n\right)^{k-1}Q_jdx\\&-\theta_4\alpha\int_{x_m}^{x_{m+1}}Q_iQ_j''dx\Bigg\}\delta_j^n-\sum_{j=m-1}^{m+2}\left\{\theta_2\int_{x_m}^{x_{m+1}}Q_iQ_jdx\right\}\sigma_j^n\\&-\int_{x_m}^{x_{m+1}}Q_i(\theta_4 f(x,t_n)+\theta_3 f(x,t_{n+1}))dx,\end{aligned}\tag{24}$$

$$\begin{aligned}&\sum_{j=m-1}^{m+2}\Bigg\{\theta_1\beta\int_{x_m}^{x_{m+1}}Q_iQ_jdx\\&+\theta_1\gamma\int_{x_m}^{x_{m+1}}Q_i\left(\sum_{r=m-1}^{m+2}Q_r\delta_r^{n+1}\right)^{k-1}Q_jdx\\&+\theta_1\alpha\int_{x_m}^{x_{m+1}}Q_iQ_j''dx\Bigg\}\delta_j^{n+1}\\&+\sum_{j=m-1}^{m+2}\Bigg\{(1+\theta_3\beta)\int_{x_m}^{x_{m+1}}Q_iQ_jdx\\&+\theta_3\gamma k\int_{x_m}^{x_{m+1}}Q_i\left(\sum_{r=m-1}^{m+2}Q_r\delta_r^{n+1}\right)^{k-1}Q_jdx\\&+\theta_3\alpha\int_{x_m}^{x_{m+1}}Q_iQ_j''dx\Bigg\}\sigma_j^{n+1}\\&-\sum_{j=m-1}^{m+2}\Bigg\{-\theta_2\beta\int_{x_m}^{x_{m+1}}Q_iQ_jdx\end{aligned}\tag{25}$$

$$
\begin{aligned}
&-\theta_2\gamma \int_{x_m}^{x_{m+1}} Q_i\left(\sum_{r=m-1}^{m+2} Q_r\delta_r^n\right)^{k-1} Q_j dx \\
&\left.-\theta_2\alpha \int_{x_m}^{x_{m+1}} Q_iQ_j^{''}dx\right\}\delta_j^n \\
&-\sum_{j=m-1}^{m+2}\left\{(1-\theta_4\beta)\int_{x_m}^{x_{m+1}} Q_iQ_jdx\right. \\
&-\theta_4\gamma k \int_{x_m}^{x_{m+1}} Q_i\left(\sum_{r=m-1}^{m+2} Q_r\delta_r^n\right)^{k-1} Q_j dx \\
&\left.-\theta_4\alpha \int_{x_m}^{x_{m+1}} Q_iQ_j^{''}dx\right\}\sigma_j^n \\
&-\int_{x_m}^{x_{m+1}} Q_i\left[\theta_1 f(x,t_{n+1})+\theta_2 f(x,t_n)\right. \\
&\left.+\theta_3 f_t(x,t_{n+1})+\theta_4 f_t(x,t_n)\right]dx
\end{aligned}
$$

where $i=m-1,m,m+1,m+2$. Denoting the integrals in (24) and (25) by

$$
\begin{aligned}
A_{ij}^e &= \int_{x_m}^{x_{m+1}} Q_iQ_jdx, \\
B_{ij}^e(\delta^{n+1}) &= \int_{x_m}^{x_{m+1}} Q_i\left(\sum_{r=m-1}^{m+2} Q_r\delta_r^{n+1}\right)^{k-1} Q_jdx, \\
C_{ij}^e &= \int_{x_m}^{x_{m+1}} Q_iQ_j^{''}dx, \\
D_i^e &= \int_{x_m}^{x_{m+1}} Q_i(\theta_4 f(x,t_n)+\theta_3 f(x,t_{n+1}))dx, \\
E_i^e &= \int_{x_m}^{x_{m+1}} Q_i[[\theta_1 f(x,t_{n+1})+\theta_2 f(x,t_n) \\
&\quad +\theta_3 f_t(x,t_{n+1})+\theta_4 f_t(x,t_n)]dx, \\
i &= m-1,m,m+1,m+2; \\
j &= m-1,m,m+1,m+2
\end{aligned}
$$

and collecting the element matrices over all elements $[x_m,x_{m+1}]$, we get the following matrix forms of the (24) and (25):

$$
\begin{aligned}
&[(1+\theta_3\beta)\mathbf{A}+\theta_3\gamma\mathbf{B}((\boldsymbol{\delta})^{n+1})+\theta_3\alpha\mathbf{C}]\,(\boldsymbol{\delta})^{n+1} \\
&-\theta_1\mathbf{A}(\boldsymbol{\sigma})^{n+1}=[(1-\theta_4\beta)\mathbf{A}-\theta_4\gamma\mathbf{B}((\boldsymbol{\delta})^{n}) \\
&-\theta_4\alpha\mathbf{C}]\,(\boldsymbol{\delta})^{n}+\theta_2\mathbf{A}(\boldsymbol{\sigma})^{n}+\mathbf{D},
\end{aligned} \tag{26}
$$

$$
\begin{aligned}
&\left[\theta_1\beta\mathbf{A}+\theta_1\gamma\mathbf{B}(\boldsymbol{\delta}^{n+1})+\theta_1\alpha\mathbf{C}\right]\boldsymbol{\delta}^{n+1} \\
&+\left[(1+\theta_3\beta)\mathbf{A}+\theta_3\gamma k\mathbf{B}(\boldsymbol{\delta}^{n+1})+\theta_3\alpha\mathbf{C}\right]\boldsymbol{\sigma}^{n+1} \\
&=\left[-\theta_2\beta\mathbf{A}-\theta_2\gamma\mathbf{B}(\boldsymbol{\delta}^{n})-\theta_2\alpha\mathbf{C}\right]\boldsymbol{\delta}^{n} \\
&+\left[(1-\theta_4)\mathbf{A}-\theta_4\gamma k\mathbf{B}(\boldsymbol{\delta}^{n})-\theta_4\alpha\mathbf{C}\right]\boldsymbol{\sigma}^{n}+\mathbf{E}
\end{aligned} \tag{27}
$$

where $\boldsymbol{\delta}^{n+1} = (\delta_{-1}^{n+1},...,\delta_{N+1}^{n+1})^T$ and $\boldsymbol{\sigma}^{n+1} = (\sigma_{-1}^{n+1},...,\sigma_{N+1}^{n+1})^T$. The set of equations consists of $2N+6$ equations with $2N+6$ unknown parameters. Before starting the iteration procedure, boundary conditions must be adapted into the system. For this purpose, we delete first and last equations from the systems (26) and (27), and eliminate the terms δ_{-1},σ_{-1} and $\delta_{N+1},\sigma_{N+1}$ from the remaining systems (26) and (27) by using boundary conditions (9). So, we obtain a new matrix system with the dimension $(2N+2)\times(2N+2)$.

To start evolution of the vector of initial parameters $(\delta_{-1}^0,\sigma_{-1}^0,\delta_0^0,\sigma_0^0,\ldots,\delta_{N-1}^0,\sigma_{N-1}^0,\delta_N^0,\sigma_N^0)$, it must be determined by using the boundary (10) and initial (11) conditions. Once the initial parameters $\boldsymbol{\delta}^0$ and $\boldsymbol{\sigma}^0$ are determined, we can start the iteration of the system to find the unknown parameters $\boldsymbol{\delta}^n$ and $\boldsymbol{\sigma}^n$ at time $t_n = n\Delta t$. Thus the approximate solution U_N and V_N (19) can be determined by using these values.

3 Numerical Experiments

In this section, we apply proposed method to four numerical examples of nonlinear KGE. The accuracy of our presented method is tested by employing maximum error norm L_∞

$$
L_\infty = \|u-U_N\|_\infty = \max_j \left|u_j - U_j\right|. \tag{28}
$$

3.1 First example

Let choose $\alpha=-1$, $\beta=0$, $\gamma=1$ and $k=2$ for the nonlinear KGE and the space domain $[0,1]$. The initial conditions and boundary conditions are given as follows

$u(x,0)=0,\ \ u_t(x,0)=0,$

$u(0,t)=0,\ \ u(1,t)=t^3$

and $f(x,t)=6xt(x^2-t^2)+x^6t^6$. For these parameters

$$
u(x,t)=x^3t^3
$$

is an exact solution of the nonlinear KGE. Using the formulation in (28), Table 1 is prepared for the various space steps as $h=0.02$, $h=0.04$, $h=0.05$ and the time step as $\Delta t=0.0001$ at various times. In the same table to compare the solutions, the results of the maximum error norm L_∞ obtained by Dehghan [7], Rashidinia [8] and Sarboland [12] are included. The results show that the accuracy of present method is considerable good than the accuracy of the other methods. The simulation of the numerical solutions at different times up to $t=5$ is shown in Fig. 1 and the absolute error propagation of the proposed method is seen in Fig. 2 at time $t=5$.

3.2 Second example

Let choose $\alpha=-1$, $\beta=0$, $\gamma=1$ and $k=2$ for the nonlinear KGE and the space domain $[-1,1]$. The initial conditions and boundary conditions are given as follows

$u(x,0)=x,\ \ u_t(x,0)=0,$

$u(-1,t)=-\cos(t),\ \ u(1,t)=\cos(t)$

Table 1: The error norms for first example with $\Delta t = 0.0001$ at various times.

Method	h	$t=1$	$t=2$
present	0.02	3.241×10^{-14}	2.093×10^{-13}
present	0.04	2.753×10^{-14}	3.606×10^{-13}
present	0.05	4.185×10^{-14}	2.344×10^{-13}
[7]	0.02	1.101×10^{-5}	1.649×10^{-4}
[8]	0.04	6.414×10^{-13}	7.451×10^{-12}
[12]	0.05	7.795×10^{-6}	1.230×10^{-4}
		$t=3$	$t=4$
present	0.02	2.321×10^{-12}	3.872×10^{-12}
present	0.04	8.166×10^{-13}	1.397×10^{-12}
present	0.05	1.417×10^{-12}	2.629×10^{-12}
[7]	0.02	5.972×10^{-4}	1.826×10^{-3}
[8]	0.04	1.624×10^{-11}	2.007×10^{-11}
[12]	0.05	5.301×10^{-4}	1.860×10^{-3}
		$t=5$	
present	0.02	5.215×10^{-12}	
present	0.04	8.242×10^{-12}	
present	0.05	6.949×10^{-12}	
[7]	0.02	3.691×10^{-3}	
[8]	0.04	2.540×10^{-11}	
[12]	0.05	3.519×10^{-3}	

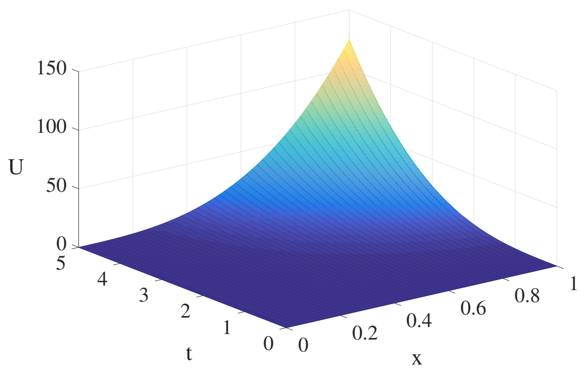

Fig. 1: Solutions up to $t = 5$ with $\Delta t = 0.0001$, $h = 0.02$.

and $f(x,t) = -x\cos(t) + x^2\cos^2(t)$. For these parameters

$$u(x,t) = x\cos(t)$$

is an exact solution of the nonlinear KGE. Maximum error norm L_∞ is presented in the Table 2 for $h = 0.2$, $h = 0.02$ and $\Delta t = 0.0001$ at various times as $t = 1,3,5,7,10$. The values of the maximum error norm L_∞ are compared with the results obtained by [7,12]. We observe that results obtained by using proposed method are more accurate than results obtained by the others. By the Fig. 3, it can be seen the solution profiles at various times for the proposed numerical method. The absolute error of the proposed method is given in Fig. 4 at $t = 10$.

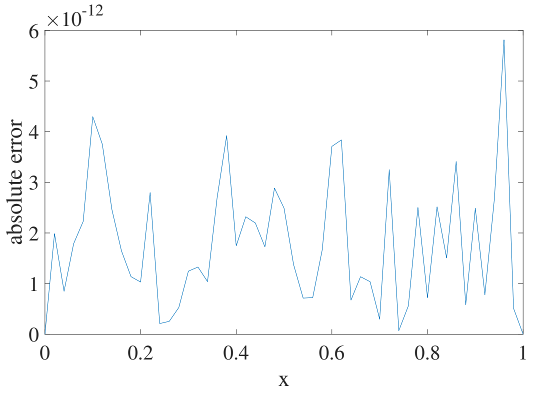

Fig. 2: Absolute error of the method in $t = 5$ with $\Delta t = 0.0001$, $h = 0.02$.

Table 2: The error norms for second example with $\Delta t = 0.0001$ at various times.

Method	h	$t=1$	$t=3$
present	0.02	1.234×10^{-13}	1.257×10^{-13}
present	0.2	4.569×10^{-14}	2.037×10^{-13}
[7]	0.02	1.254×10^{-5}	1.555×10^{-5}
[12]	0.2	1.259×10^{-5}	1.542×10^{-5}
		$t=5$	$t=7$
present	0.02	1.387×10^{-13}	2.716×10^{-13}
present	0.2	1.739×10^{-13}	1.693×10^{-13}
[7]	0.02	3.379×10^{-5}	3.775×10^{-5}
[12]	0.2	3.362×10^{-5}	3.741×10^{-5}
		$t=10$	
present	0.02	1.652×10^{-13}	
present	0.2	2.094×10^{-13}	
[7]	0.02	1.309×10^{-5}	

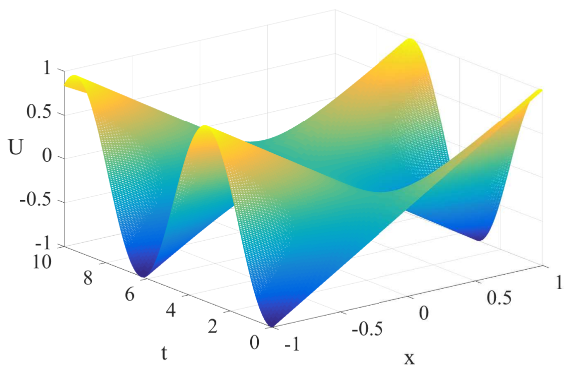

Fig. 3: Solutions up to $t = 10$ with $\Delta t = 0.0001$, $h = 0.01$.

3.3 Third example

The another exact solution of the KGE is of

$$u(x,t) = B\tan(K(x+ct))$$

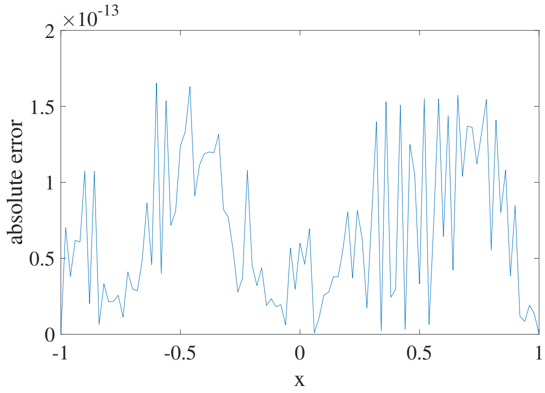

Fig. 4: Absolute error of the method in $t = 10$ with $\Delta t = 0.0001$, $h = 0.01$.

where $B = \sqrt{\frac{\beta}{\gamma}}$ and $K = \sqrt{\frac{-\beta}{2(\alpha+c^2)}}$. The initial conditions and boundary conditions are given as follows

$$u(x,0) = B\tan(Kx), \quad u_t(x,0) = BcK\sec^2(Kx),$$
$$u(0,t) = B\tan(Kct), \quad u(1,t) = B\tan(K+Kct)$$

with $f(x,t) = 0$, $\alpha = -2.5$, $\beta = 1$, $\gamma = 1.5$, $k = 3$ and the space domain as $[0,1]$ for this example. The given problem has been solved by taking $h = 0.01$, 0.02, 0.05 and $\Delta t = 0.001$. L_∞ errors for two different c values such that $c = 0.5$ and $c = 0.05$ are presented in Tables 3 and 4, respectively. The results obtained by present method are compared with the results obtained by [7,8,12] in Tables 3 and 4. It is obvious that for values of $c = 0.5$ and $c = 0.05$, the proposed method is more accurate than the other methods [7,8,12]. The propagations of the numerical solutions at various times for values of $c = 0.5$ and $c = 0.05$ are shown in Figs. 5 and 6. The simulations of the absolute errors for $c = 0.5$ and $c = 0.05$ are seen in Figs. 7 and 8.

Table 3: The error norms for third example with $\Delta t = 0.001$, $c = 0.5$.

Method	h	$t = 1$	$t = 2$
present	0.01	1.693×10^{-11}	8.182×10^{-11}
present	0.02	2.585×10^{-10}	1.258×10^{-9}
[7]	0.01	5.996×10^{-6}	2.197×10^{-5}
[8]	0.01	2.694×10^{-8}	8.746×10^{-8}
[12]	0.02	5.213×10^{-6}	2.180×10^{-5}
		$t = 3$	$t = 4$
present	0.01	8.421×10^{-10}	7.625×10^{-8}
present	0.02	1.266×10^{-8}	1.047×10^{-6}
[7]	0.01	9.089×10^{-5}	8.294×10^{-4}
[8]	0.01	3.090×10^{-7}	1.939×10^{-6}
[12]	0.02	9.011×10^{-5}	8.237×10^{-4}

Table 4: The error norms for third example with $\Delta t = 0.001$, $c = 0.05$.

Method	h	$t = 1$	$t = 2$
present	0.01	4.050×10^{-12}	4.319×10^{-12}
present	0.05	2.323×10^{-9}	2.434×10^{-9}
[7]	0.01	3.649×10^{-7}	3.895×10^{-7}
[8]	0.01	1.198×10^{-8}	2.473×10^{-8}
[12]	0.05	2.178×10^{-7}	3.064×10^{-7}
		$t = 3$	$t = 4$
present	0.01	5.066×10^{-12}	5.850×10^{-12}
present	0.05	2.849×10^{-9}	3.259×10^{-9}
[7]	0.01	4.213×10^{-7}	4.592×10^{-7}
[8]	0.01	2.895×10^{-8}	1.991×10^{-8}
[12]	0.05	3.700×10^{-7}	3.423×10^{-7}

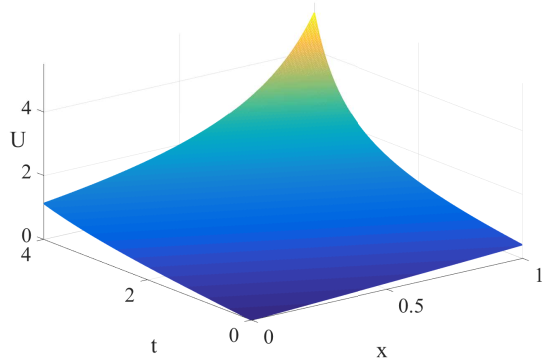

Fig. 5: Solutions up to $t = 4$ with $\Delta t = 0.001$, $h = 0.01$, $c = 0.5$.

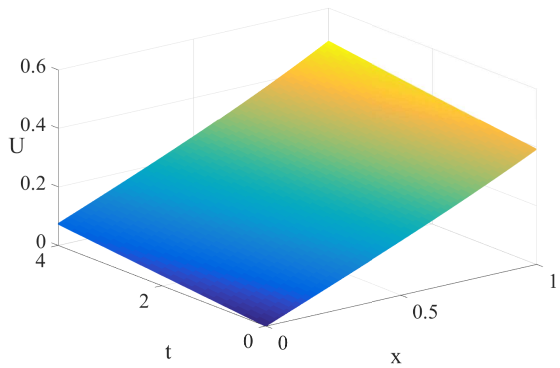

Fig. 6: Solutions up to $t = 4$ with $\Delta t = 0.001$, $h = 0.01$, $c = 0.05$.

4 Conclusion

In this paper, we have proposed a high order accurate algorithm for the numerical solution of the KGE. This algorithm is obtained by using cubic B-spline functions with the well known finite element method as Galerkin method for the space discretization of the KGE and the

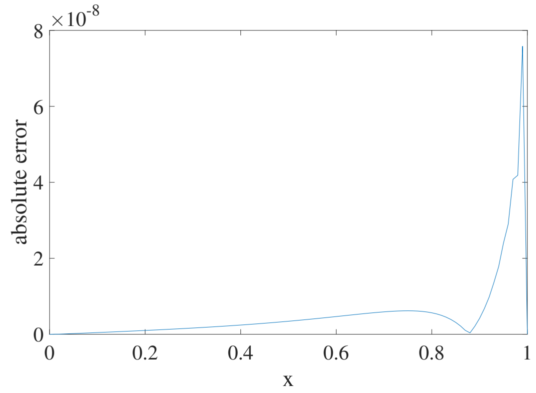

Fig. 7: Absolute error of the method in $t = 4$ with $\Delta t = 0.001$, $h = 0.01$, $c = 0.5$.

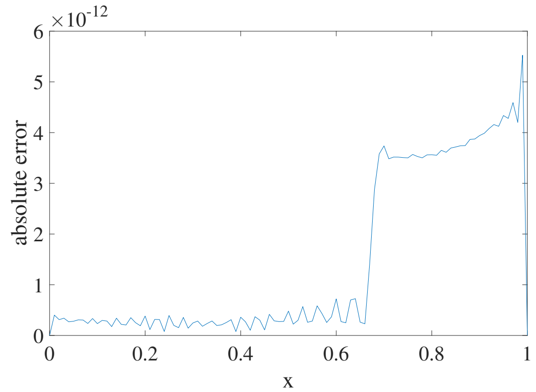

Fig. 8: Absolute error of the method in $t = 4$ with $\Delta t = 0.001$, $h = 0.01$, $c = 0.05$.

fourth order finite difference method in time for time discretization of the KGE. To test the achievement of the proposed method, three examples are examined. By comparing the maximum errors L_∞ obtained by the proposed method with those of the others (radial basis function approximation method, cubic B-spline collocation method and multiquadric quasi-interpolation method), it has been seen that the accuracy of the proposed method is high. Consequently, the finite element method which is of order four used for time discretization increases the accuracy of the proposed algorithm.

Acknowledgement

This work was supported by the Scientific Research Council of Eskisehir Osmangazi University under Project No. 2018-2090.

Conflict of Interest The authors declare that they have no conflict of interest

References

[1] J. J. Sakurai, *Advanced quantum mechanics*, Pearson Education, India, 1967.
[2] X. Li and B. Y. Guo, A Legendre pseudospectral method for solving nonlinear Klein-Gordon equation, *Journal of Computational Mathematics*, **15**, 105-126, 1997.
[3] D. B. Duncan, Symplectic finite difference approximations of the nonlinear Klein-Gordon equation, *SIAM Journal on Numerical Analysis*, **34**, 1742-1760, 1997.
[4] D. Kaya and S. M. El-Sayed, A numerical solution of the Klein-Gordon equation and convergence of the decomposition method, *Applied Mathematics and Computation*, **156**, 341-353, 2004.
[5] M. E. Khalifa and M. Elgamal, A numerical solution to Klein-Gordon equation with Dirichlet boundary condition, *Applied Mathematics and Computation*, **160**, 451-475, 2005.
[6] F. Shakeri and M. Dehghan, Numerical solution of the Klein–Gordon equation via He's variational iteration method, *Nonlinear Dynamics*, **51**, 89-97, 2008.
[7] M. Dehghan and A. Shokri, Numerical solution of the nonlinear Klein Gordon equation using radial basis functions, *Journal of Computational and Applied Mathematics*, **230**, 400-410, 2009.
[8] J. Rashidinia, M. Ghasemi and R. Jalilian, Numerical solution of the nonlinear Klein-Gordon equation, *Journal of Computational and Applied Mathematics*, **233**, 1866-1878, 2010.
[9] W. Bao and X. Dong, Analysis and comparison of numerical methods for the Klein–Gordon equation in the nonrelativistic limit regime, *Numerische Mathematik*, **120**, 189-229, 2012.
[10] A. Verma, R. Jiwari and S. Kumar, A numerical scheme based on differential quadrature method for numerical simulation of nonlinear Klein-Gordon equation, *International Journal of Numerical Methods for Heat & Fluid Flow*, **24**, 1390-1404, 2014.
[11] A. Hussain, S. Haq and M. Uddin, Numerical solution of Klein-Gordon and sine-Gordon equations by meshless method of lines, *Engineering Analysis with Boundary Elements*, **37**, 1351-1366, 2013.
[12] M. Sarboland, A. Aminataei, Numerical solution of the nonlinear Klein-Gordon equation using multiquadric quasi-interpolation scheme, *Universal Journal of Applied Mathematics*, **3**, 40-49, 2015.
[13] S. C. Shiralashetti, L. M. Angadi, A. B. Deshi and M. H. Kantli, Haar wavelet method for the numerical solution of Klein–Gordan equations, *Asian-European Journal of Mathematics*, **9**, 2016.
[14] O. Ersoy Hepson, A. Korkmaz and I. Dag, On the numerical solution of the Klein-Gordon equation by exponential cubic B-spline collocation method, *Communications Faculty of Sciences University of Ankara Series A1 Mathematics and Statistics*, **68**, 412-421, 2018.
[15] H. Yang, High-Order Energy and Linear Momentum Conserving Methods for the Klein-Gordon Equation, *Mathematics*, **6**, 2018.
[16] H. Selvitopi, M. Yazıcı, Numerical results for the Klein-Gordon equation in de Sitter spacetime, *Mathematical Methods in the Applied Sciences*, **42**, 5446-5454, 2018.

Dursun Irk received the M.S. degree from the Department of Mathematics at Eskisehir Osmangazi University in 2002 and the Ph.D. from the same department at Eskisehir Osmangazi University in 2007. He is a Professor of the numerical analysis, working as both research and teaching staff at the Department of Matematics and Computer Science, Eskisehir Osmangazi University since 2000. His research interest includes the numerical solutions of the partial differential equations using B-spline finite element and difference methods.

Emre Kirli received Bsc degree from the Department of Mathematics at Anadolu University in 2013. He is a PhD student from the Department of Matematics and Computer Science at Eskisehir Osmangazi University supervised under Prof. Dursun Irk. He has been working in the Department of Matematics at Istanbul Bilgi University as a Research Assistant since 2015. He is interested in finite element methods, numerical analysis and partial differentail equations.

Melis Zorsahin Gorgulu received the M.S. degree from the Department of Mathematics at Eskisehir Osmangazi University in 2009 and the Ph.D. degree from the Department of Matematics and Computer Science at the same University in 2016. She has been working in the Department of Matematics and Computer Science at Eskisehir Osmangazi University as a Lecturer since 2011. She is interested in numerical solutions of partial differential equations, finite difference and element methods and B-spline functions.