

# Numerical Solution of Nonlinear Klein-Gordon Equation Using Polynomial Wavelets

Jalil Rashidinia and Mahmood Jokar

**Abstract** The main aim of this paper is to apply the polynomial wavelets for the numerical solution of nonlinear Klein-Gordon equation. Polynomial scaling and wavelet functions are rarely used in the contexts of numerical computation. A numerical technique for the solution of nonlinear Klein-Gordon equation is presented. Our approach consists of finite difference formula combined with the collocation method, which uses the polynomial wavelets. Using the operational matrix of derivative, we reduce the problem to a set of algebraic equations by expanding the approximate solution in terms of polynomial wavelets with unknown coefficients. An estimation of error bound for this method is investigated. Some illustrative examples are included to demonstrate the validity and applicability of the approach.

## 1 Introduction

In this work, we are dealing with the numerical solutions of the following nonlinear partial differential equation, namely a Nonlinear Klein-Gordon equation:

$$u_{tt} + \alpha u_{xx} + \beta u + \gamma u^k = f(x, t), \quad x \in \Omega = [-1, 1], \quad t \in (0, T], \quad (1)$$

subject to the initial conditions

$$\begin{cases} u(x, 0) = g_1(x), & x \in \Omega, \\ u_t(x, 0) = g_2(x), & x \in \Omega, \end{cases} \quad (2)$$

---

J. Rashidinia · M. Jokar (✉)  
School of Mathematics, Iran University of Science and Technology, Tehran, Iran  
e-mail: jokar.mahmod@gmail.com

J. Rashidinia  
e-mail: rashidinia@iust.ac.ir

and Dirichlet boundary condition

$$u(x, t) = h(x, t), \quad x \in \partial\Omega, \quad t \in (0, T], \quad (3)$$

where  $\alpha$ ,  $\beta$  and  $\gamma$  are nonzero real constants and  $f$  is a known analytic function. Nonlinear phenomena occur in a wide variety of scientific applications such as plasma physics, solid state physics, fluid dynamics and chemical kinetics [1]. The nonlinear Klein-Gordon equation is one of the important models in quantum mechanics and mathematical physics. The equation has received considerable attention in studying solitons and condensed matter physics, in investigating the interaction of solitons in collisionless plasma, the recurrence of initial states, and in examining the nonlinear wave equations [2].

There are a lot of studies on the numerical solution of initial and initial-boundary problems of the linear or nonlinear Klein-Gordon equation. For instance, Chowdhury and Hashim [3] employed the homotopy-perturbation method to obtain approximate solutions of the Klein-Gordon and sine-Gordon equations. Four finite difference schemes for approximating the solution of nonlinear Klein-Gordon equation were discussed in [4]. Deeba and Khuri [5] presented a decomposition scheme for obtaining numerical solutions of the Eq. (1). In [6], a spline collocation approach for the numerical solution of a generalized nonlinear Klein-Gordon equation was investigated. Dehghan and Shokri [7] proposed a numerical scheme to solve the one-dimensional nonlinear Klein-Gordon equation with quadratic and cubic nonlinearity using the collocation points and approximating the solution by Thin Plate Splines radial basis functions. Authors in [8] considered a numerical method based on the cubic B-splines collocation technique on the uniform mesh points. Lakestani and Dehghan [9] presented two numerical techniques. The first one is Mixed Finite Difference in time and Collocation Methods using cubic B-spline functions in space (MFDCM) and the second one is fully Collocation Method (CM) which approximates the solution in both space and time variables using cubic B-spline functions. A fully implicit and discrete energy conserving finite difference scheme for the solution of an initial-boundary value problem of the nonlinear Klein-Gordon equation derived by Wong et al. [10]. A three-level spline-difference scheme to solve the one dimensional Klein-Gordon equation which is based on using the finite difference approximation for the time derivative and the spline approximation for the second-order spatial derivative was derived by authors in [11]. Most recently, authors in [12] proposed a spectral method using Legendre wavelets.

In this article we study the application of polynomial scaling functions and wavelets for computation of numerical solution of nonlinear Klein-Gordon equation. A numerical technique based on the finite difference and Collocation methods is presented. At the first stage, our method is based on the discretization of the time variable by means of the Crank-Nicolson method and freezing the coefficients of the resulting ordinary differential equation at each time step. At the second stage, we use the Wavelet-Collocation method on the yield linear ordinary differential equations at each time step resulting from the time semidiscretization. Considering this basis being wavelet functions, our method is essentially a spectral method.

Polynomial scaling and wavelet functions are rarely used in the contexts of numerical computation [13]. One of the advantages of using polynomial scaling function as expansion functions is the good representation of smooth functions by finite Chebyshev expansion. The Crank-Nicolson method is an unconditionally stable, implicit numerical scheme with second-order accuracy in both time and space. Our approach consists of reducing the Klein-Gordon equation to a set of algebraic equations by expanding the approximate solution in terms of wavelet functions with unknown coefficients. The operational matrix of derivative is presented. This matrix together with the Collocation method are then utilized to evaluate the unknown coefficients of the solution at each time step. Finally, the convergence analysis of the proposed method for the Eq. (1) is developed.

The organization of this paper is as follows. In Sect. 2, we describe the polynomial scaling and wavelet functions on  $[-1, 1]$  and some basic properties. In Sect. 3, the proposed method is used to approximate the solution of the problem. As a result, a set of algebraic equations is formed and a solution of the considered problem is introduced at each time step. In Sect. 4, the error bounds of the method based on the Crank-Nicolson and polynomial wavelets are presented. In Sect. 5, we discuss the accuracy and efficiency of the employed method by applying to several test problems. A brief conclusion is given at the end of the paper in Sect. 6.

## 2 Preliminary

In this section, we shall give a brief introduction of the polynomial wavelets on  $[-1, 1]$  and their basic properties [14]. Also the construction of the operational matrix of the derivative and some approximation results [15] are presented.

### 2.1 Polynomial Wavelets

Suppose that  $T_n$  and  $U_n$  be the following Chebyshev polynomials of the first and second kind respectively,

$$T_n(x) = \cos(n \arccos(x)) \quad \text{and} \quad U_{n-1}(x) = \frac{\sin(n \arccos(x))}{\sin(\arccos(x))},$$

here to introduce polynomial scaling function we need  $\omega_j$  which is defined as:

$$\omega_j(x) = (1 - x^2)U_{2^j-1}(x) = \frac{(1 - x^2)}{2^j}T'_{2^j}(x), \quad j = 0, 1, 2, \dots$$

The zeros of  $\omega_j(x)$  are  $x_k = \cos(\frac{k\pi}{2^j})$  for  $k = 0, 1, \dots, 2^j$  and it should be pointed out that the zeros of  $\omega_j$  are also zeros of  $\omega_{j+1}$ .

Let

$$\varepsilon_{j,l} = \begin{cases} \frac{1}{2} & \text{for } l = 0 \text{ or } l = 2^j \\ 1 & \text{for } l = 1, 2, \dots, 2^j - 1 \end{cases}$$

for any  $j \in \mathbb{N}_0 = \mathbb{N} \cup \{0\}$ , the polynomial scaling functions are defined as:

$$\phi_{j,l}(x) = \frac{\omega_j(x)}{2^j(-1)^{l+1}(x - x_l)} \varepsilon_{j,l}, \quad l = 0, 1, \dots, 2^j. \tag{4}$$

Given  $j \in \mathbb{N}_0$ , the space of polynomial scaling functions on  $[-1, 1]$  is defined by  $V_j = \text{span}\{\phi_{j,l} : l = 0, 1, \dots, 2^j\}$ .

It is easy to see that the spaces  $V_j = \Pi_{2^j}$  where  $\Pi_n$  denotes the set of all polynomials of degree at most  $n$ . The interpolatory property of this functions which helps to accelerate the computations is:

$$\phi_{j,l}(\cos(\frac{k\pi}{2^j})) = \delta_{k,l}, \quad k, l = 0, 1, \dots, 2^j. \tag{5}$$

The wavelet spaces are defined by  $W_j = \text{span}\{\psi_{j,l} : l = 0, 1, \dots, 2^j - 1\}$ , where

$$\psi_{j,l}(x) = \frac{T_{2^j}(x)}{2^j(x - \cos((2l + 1)\pi/2^{j+1}))} \left( 2\omega_j(x) - \omega_j\left(\cos\frac{(2l + 1)\pi}{2^{j+1}}\right) \right).$$

The same interpolating property holds with the zeros of  $\omega_{j+1}$  as:

$$\psi_{j,l}(\cos(\frac{(2k + 1)\pi}{2^{j+1}})) = \delta_{k,l}, \quad k, l = 0, 1, \dots, 2^j - 1.$$

We note that  $\dim W_j = 2^j$  and  $\dim V_j = 2^{j+1}$ , also for all  $j \in \mathbb{N}_0$  we have  $V_{j+1} = V_j \oplus W_j$  and by denoting  $W_{-1}$  as  $V_0$  we obtain

$$\Pi_{2^{j+1}} = V_{j+1} = \bigoplus_{k=-1}^j W_k. \tag{6}$$

### 2.2 Function Approximation

For any  $j \in \mathbb{N}_0$ , the operator  $L_j$  mapping any real-valued function  $f(x)$  on  $[-1, 1]$  into the space  $V_j$  by the Lagrange formula

$$L_j f(x) = \sum_{l=0}^{2^j} f(x_l)\phi_{j,l}(x) = U^T \Phi_j(x), \tag{7}$$

where  $U$  and  $\Phi_j$  are vectors with  $2^j + 1$  components as:

$$U = [f(x_0), f(x_1), \dots, f(x_{2^j})]^T, \tag{8}$$

$$\Phi_j(x) = [\phi_{j,0}(x), \phi_{j,1}(x), \dots, \phi_{j,2^j}(x)]^T. \tag{9}$$

Considering (6) it follows that

$$L_j f(x) = \sum_{k=0}^1 a_k \phi_{0,k}(x) + \sum_{l=0}^{j-1} \sum_{i=0}^{2^l-1} b_{l,i} \psi_{l,i}(x) = C^T \Psi_{j-1}(x), \tag{10}$$

where  $\phi_{0,k}(x)$  and  $\psi_{l,i}(x)$  are scaling and wavelet functions, respectively, and  $C$  and  $\Psi_{j-1}$  are vectors with  $2^j + 1$  components as:

$$C = [a_0, a_1, b_{0,0}, b_{1,0}, b_{1,1}, \dots, b_{j-1,2^{j-1}-1}]^T, \tag{11}$$

$$\Psi_{j-1}(x) = [\phi_{0,0}, \phi_{0,1}, \psi_{0,0}, \psi_{1,0}, \psi_{1,1}, \dots, \psi_{j-1,2^{j-1}-1}]^T. \tag{12}$$

The vector  $C$  can be obtained by considering,

$$\Psi_{j-1} = \mathbf{G} \Phi_j, \tag{13}$$

where  $\mathbf{G}$  is a  $(2^j + 1) \times (2^j + 1)$  matrix, which can be determined as follows. Using the two scale relations and decomposition between polynomial scaling and wavelet functions represented in [14, pp. 100], we have

$$\Phi_{j-1} = \lambda_{j-1} \Phi_j, \quad \Psi_{j-1} = \mu_{j-1} \Phi_j, \tag{14}$$

where  $\lambda_{j-1}$  is a  $(2^{j-1} + 1) \times (2^j + 1)$  matrix and  $\mu_{j-1}$  is a  $(2^{j-1}) \times (2^j + 1)$  matrix. Following [16] and by using Eqs. (13) and (14), we obtain

$$\mathbf{G} = \begin{bmatrix} \lambda_0 \times \lambda_1 \times \dots \times \lambda_{j-1} \\ \mu_0 \times \lambda_1 \times \dots \times \lambda_{j-1} \\ \vdots \\ \mu_{j-3} \times \mu_{j-2} \times \lambda_{j-1} \\ \mu_{j-2} \times \mu_{j-1} \\ \mu_{j-1} \end{bmatrix}.$$

By using Eqs. (13) and (7), we get

$$L_j f(x) = U^T \Phi_j = U^T \mathbf{G}^{-1} \Psi_{j-1},$$

so that we have  $C^T = U^T \mathbf{G}^{-1}$ .

Interpolation properties of polynomial scaling functions could help to obtaining the coefficients very fast, because it just needs to replacing the variable of function by the zeros of  $\omega_j(x)$  and no need of integration. In the rest of the paper for simplicity and abbreviation we denote  $\Phi_j(x)$  and  $\Psi_{j-1}(x)$  by  $\Phi(x)$  and  $\Psi(x)$ , respectively.

### 2.3 Operational Matrix of Derivative

Polynomial scaling functions operational matrix of derivative was derived in [15]. Here, we just list the theorem and a corollary as follows.

**Theorem 1** *The differentiation of vector  $\Phi(x)$  in (9) can be expressed as:*

$$\Phi'(x) = D_\phi \Phi(x), \tag{15}$$

where  $D_\phi$  is  $(2^j + 1) \times (2^j + 1)$  and the entries of operational matrix of derivative for polynomial scaling functions  $D_\phi$  are:

$$d_{k,l} = \begin{cases} \sum_{i=0, i \neq k}^{2^j} \frac{1}{x_l - x_i}, & \text{if } l = k, \\ 2^{2^j - j - 1} (-1)^k \varepsilon_{j,k} \prod_{r=0, r \neq l, k}^{2^j} (x_l - x_r), & \text{if } l \neq k. \end{cases}$$

*Proof* See [15].

**Corollary 2** *Using matrix  $\mathbf{G}$  the operational matrix of derivative for polynomial wavelets can be represented as:*

$$D_\psi = \mathbf{G}^{-1} D_\phi \mathbf{G}. \tag{16}$$

## 3 The Polynomial Wavelet Method (PWM)

In this section, we solve nonlinear partial differential equation (1) on a bounded domain. For this end, we use finite difference method for one variable to reduce these equations to a system of ordinary differential equations, then we solve this system and find the solution of the given Klein-Gordon equation at the points  $t_n = n\delta t$  for  $\delta t = \frac{T-0}{N}$ ,  $n = 0, 1, \dots, N$ .

In order to perform temporal discretization, we discretize (1) according to the following  $\theta$ -weighted type scheme

$$\frac{u^{n+1} - 2u^n + u^{n-1}}{(\delta t)^2} + \theta (\alpha u_{xx}^{n+1} + \beta u^{n+1}) + (1 - \theta) (\alpha u_{xx}^n + \beta u^n) + \gamma (u^n)^k = f(x, t_n), \tag{17}$$

where  $\delta t$  is the time step size and  $u^{n+1}$  is used to show  $u(x, t + \delta t)$ . By choosing  $\theta = \frac{1}{2}$  (Crank-Nicolson scheme) and rearranging Eq. (17) we obtain

$$u^{n+1} + \frac{\beta(\delta t)^2}{2}u^{n+1} + \frac{\alpha(\delta t)^2}{2}u_{xx}^{n+1} = \left(2 - \frac{\beta(\delta t)^2}{2}\right)u^n - \frac{\alpha(\delta t)^2}{2}u_{xx}^n - \gamma(\delta t)^2(u^n)^k - u^{n-1} + (\delta t)^2 f(x, t_n). \quad (18)$$

Using Eq. (7), the approximate solution for  $u^n(x)$  via scaling functions is represented by formula

$$L_j u^n(x) = \mathbf{U}_n^T \Phi(x), \quad (19)$$

where vectors  $\mathbf{U}_n$  and  $\Phi(x)$  are defined as (8) and (9) respectively.

For the derivatives of  $u^n(x)$  by using (15) we can write the following relations,

$$L_j u_x^n(x) = \mathbf{U}_n^T \Phi'(x) = \mathbf{U}_n^T D_\phi \Phi(x), \quad (20)$$

$$L_j u_{xx}^n(x) = \mathbf{U}_n^T \Phi''(x) = \mathbf{U}_n^T D_\phi^2 \Phi(x). \quad (21)$$

Replacing Eqs. (19) and (21) in Eq. (18) we obtain

$$\begin{aligned} & \mathbf{U}_{n+1}^T \Phi(x) + \frac{\beta(\delta t)^2}{2} \mathbf{U}_{n+1}^T \Phi(x) + \frac{\alpha(\delta t)^2}{2} \mathbf{U}_{n+1}^T D_\phi^2 \Phi(x) \\ &= \left(2 - \frac{\beta(\delta t)^2}{2}\right) \mathbf{U}_n^T \Phi(x) - \frac{\alpha(\delta t)^2}{2} \mathbf{U}_n^T D_\phi^2 \Phi(x) - \gamma(\delta t)^2 (\mathbf{U}_n^T \Phi(x))^k \\ & \quad - \mathbf{U}_{n-1}^T \Phi(x) + (\delta t)^2 f(x, t_n). \end{aligned} \quad (22)$$

Substituting Eq. (13) into Eq. (22), we change current base to the polynomial wavelets bases

$$\begin{aligned} & \mathbf{U}_{n+1} \left[ I + \frac{\beta(\delta t)^2}{2} I + \frac{\alpha(\delta t)^2}{2} D_\phi^2 \right] \mathbf{G}^{-1} \Psi(x) \\ &= \left(2 - \frac{\beta(\delta t)^2}{2}\right) \mathbf{U}_n^T \mathbf{G}^{-1} \Psi(x) - \frac{\alpha(\delta t)^2}{2} \mathbf{U}_n^T D_\phi^2 \mathbf{G}^{-1} \Psi(x) \\ & \quad - \gamma(\delta t)^2 (\mathbf{U}_n^T \mathbf{G}^{-1} \Psi(x))^k - \mathbf{U}_{n-1}^T \mathbf{G}^{-1} \Psi(x) + (\delta t)^2 f(x, t_n). \end{aligned} \quad (23)$$

By collocating Eq. (23) in the points  $x_k = \cos(\frac{k\pi}{2^j})$ ,  $k = 0, 1, \dots, 2^j$ , we get,

$$\begin{aligned} & \mathbf{U}_{n+1} \left[ I + \frac{\beta(\delta t)^2}{2} I + \frac{\alpha(\delta t)^2}{2} D_\phi^2 \right] \mathbf{G}^{-1} \Psi(x_k) \\ &= \left(2 - \frac{\beta(\delta t)^2}{2}\right) \mathbf{U}_n^T \mathbf{G}^{-1} \Psi(x_k) - \frac{\alpha(\delta t)^2}{2} \mathbf{U}_n^T D_\phi^2 \mathbf{G}^{-1} \Psi(x_k) \\ & \quad - \gamma(\delta t)^2 (\mathbf{U}_n^T \mathbf{G}^{-1} \Psi(x_k))^k - \mathbf{U}_{n-1}^T \mathbf{G}^{-1} \Psi(x_k) + (\delta t)^2 f(x_k, t_n), \end{aligned} \quad (24)$$

which represents a system of  $(2^j + 1) \times (2^j + 1)$  equations.

Using Eq. (19) in (3) we have

$$\mathbf{U}_{n+1}^T \mathbf{G}^{-1} \Psi(-1) = h(-1, t_{n+1}), \tag{25}$$

$$\mathbf{U}_{n+1}^T \mathbf{G}^{-1} \Psi(1) = h(1, t_{n+1}). \tag{26}$$

Because the rank of matrix  $D_\phi$  is  $2^j$  and the rank of  $D_\phi^2$  is  $2^j - 1$  we replace Eqs. (25)–(26) instead of first and last equations of the system (24), so we finally obtain a following matrix form of equations,

$$\mathcal{A}_n \mathbf{U}_{n+1} = \mathcal{B}_n, \quad n = 1, 2, \dots \tag{27}$$

where  $\mathcal{A}_n$  is a matrix with dimension  $(2^j + 1) \times (2^j + 1)$  as,

$$\mathcal{A}_n = \begin{bmatrix} \Psi^T(-1)(\mathbf{G}^{-1})^T \\ \Psi^T(x_1)(\mathbf{G}^{-1})^T \left[ I + \frac{\beta(\delta t)^2}{2} I + \frac{\alpha(\delta t)^2}{2} D_\phi^2 \right]^T \\ \vdots \\ \Psi^T(x_{2^{j-1}})(\mathbf{G}^{-1})^T \left[ I + \frac{\beta(\delta t)^2}{2} I + \frac{\alpha(\delta t)^2}{2} D_\phi^2 \right]^T \\ \Psi^T(1)(\mathbf{G}^{-1})^T \end{bmatrix}$$

and

$$\mathcal{B}_n = \begin{bmatrix} h(-1, t_{n+1}) \\ \Gamma_1 \\ \vdots \\ \Gamma_{2^j-1} \\ h(1, t_{n+1}) \end{bmatrix}$$

with

$$\Gamma_i = \left( 2 - \frac{\beta(\delta t)^2}{2} \right) \mathbf{U}_n^T \mathbf{G}^{-1} \Psi(x_i) - \frac{\alpha(\delta t)^2}{2} \mathbf{U}_n^T D_\phi^2 \mathbf{G}^{-1} \Psi(x_i) - \gamma(\delta t)^2 (\mathbf{U}_n^T \mathbf{G}^{-1} \Psi(x_i))^k - \mathbf{U}_{n-1}^T \mathbf{G}^{-1} \Psi(x_i) + (\delta t)^2 f(x_i, t_n), \quad i = 1, \dots, 2^j - 1.$$

Using the first initial condition of Eq. (2), we have

$$\mathbf{U}_0^T \mathbf{G}^{-1} \Psi(x) = g_1(x), \tag{28}$$



By using the second initial condition of Eq. (2), one can get

$$\frac{u^1(x) - u^{-1}(x)}{2(\delta t)^2} = g_2(x), \quad x \in \Omega. \tag{29}$$

Equation (29) can be rewritten as

$$\mathbf{U}_{-1}^T \mathbf{G}^{-1} \Psi(x) = \mathbf{U}_1^T \mathbf{G}^{-1} \Psi(x) - 2(\delta t)^2 g_2(x). \tag{30}$$

Equation (27) using Eqs. (28) and (30) as the starting points, gives the system of equations with  $2^j + 1$  unknowns and equations, which can be solved to find  $\mathbf{U}_{n+1}$  in any step  $n = 1, 2, \dots$ . So the unknown functions  $u(x, t_n)$  in any time  $t = t_n, n = 0, 1, 2, \dots$  can be found.

### 4 Error Bounds

Here we give the error analysis of the method presented in the previous section for the Nonlinear Klein-Gordon equation. Suppose that  $BV$  be the set of real valued functions  $\mathcal{P} : \mathbb{R} \rightarrow \mathbb{R}$  with bounded variation on  $[-1, 1]$ . The value  $V(\mathcal{P}(x))$  is defined as total variation of  $\mathcal{P}(x)$  on  $[-1, 1]$ . Let for the given weight function  $w(x) = \frac{1}{\sqrt{1-x^2}}$ , and for  $2 \leq p < \infty$ , we define

$$\|\mathcal{P}\|_p := \left( \int_{-1}^1 |\mathcal{P}(x)|^p w(x) dx \right)^{\frac{1}{p}}.$$

Here we need to recall two corollaries from [17].

**Corollary 3** Let  $p \geq 2, \mathcal{P}^{(s)} \in BV$  and  $0 \leq s \leq 2^j$  then,

$$\|\mathcal{P} - L_j \mathcal{P}\|_p \leq \xi 2^{-j(s+1/p)} V(\mathcal{P}^{(s)}). \tag{31}$$

**Corollary 4** Let  $p \geq 2, 0 \leq l \leq s$  and  $\mathcal{P}^{(s)} \in BV$ , then for the interpolatory polynomial based on the zeros of the Jacobi polynomial we have,

$$\|(\mathcal{P} - L_j \mathcal{P})^{(l)}\|_p \leq \xi 2^{-j(s+1/p - \max\{l, 2l-1/p\})} V(\mathcal{P}^{(s)}). \tag{32}$$

In the above corollaries  $\xi$  is constant depends on  $s$ .

We consider Eq. (18) as an operator equation in the form

$$\mathcal{H}u^{n+1} = \left( \left(1 + \frac{\beta(\delta t)^2}{2}\right) \mathcal{I} + \frac{\alpha(\delta t)^2}{2} \mathcal{D} \right) u^{n+1} = F, \tag{33}$$

where  $\mathcal{I}$  is an identity operator and

$$\mathcal{D} = \frac{d^2}{dx^2},$$

$$F = \left(2 - \frac{\beta(\delta t)^2}{2}\right) u^n - \frac{\alpha(\delta t)^2}{2} u_{xx}^n - \gamma(\delta t)^2 (u^n)^k - u^{n-1} + (\delta t)^2 f(x, t_n).$$

For the operator equation (33) the approximate equation is

$$L_j(\mathcal{H}) u_j^{n+1} = L_j \left( \left(1 + \frac{\beta(\delta t)^2}{2}\right) \mathcal{I} + \frac{\alpha(\delta t)^2}{2} \mathcal{D} \right) u_j^{n+1} = F_j. \tag{34}$$

System (34) may be solved numerically to yield an approximate solution equation (1) at each level of time given by the expression  $u_j^{n+1} = \mathbf{U}_{n+1}^T \Phi(x)$ . Next Lemma will give the approximation results of the differential operator, and then the total error bound for  $\|E_j^{n+1}\|_p = \|u^{n+1} - u_j^{n+1}\|_p$  will be presented.

**Lemma 5** *If  $(u^{n+1})^{(s)} \in BV, s \geq 0$  then for the operator  $\mathcal{D}$  we have*

$$\|\mathcal{D}u^{n+1} - L_j \mathcal{D}u^{n+1}\|_p \leq C_1 2^{-j(s+2/p-4)} \|V((u^{n+1})^{(s)})\|_p. \tag{35}$$

*Proof* Using (32) by considering  $l = 2$  we have

$$\begin{aligned} \|\mathcal{D}u^{n+1} - L_j \mathcal{D}u^{n+1}\|_p &\leq \|u_{xx}^{n+1} - L_j u_{xx}^{n+1}\|_p \\ &\leq \|(u^{n+1} - L_j u^{n+1})_{xx}\|_p \\ &\leq C_1 2^{-j(s+2/p-4)} V((u^{n+1})^{(s)}). \end{aligned}$$

**Theorem 6** *If  $u^{n+1}$  and  $u_j^{n+1}$  be the exact and approximate solution of (1) at each level of time  $n + 1$  respectively, also assume that the operator  $\mathcal{H} = \left(1 + \frac{\beta(\delta t)^2}{2}\right) \mathcal{I} + \frac{\alpha(\delta t)^2}{2} \mathcal{D}$  has bounded inverse and  $(u^{n+1})^{(s)}, F^{(s)} \in BV, s \geq 0$ , then*

$$\|E_j^{n+1}\|_p \leq C \zeta \|(L_j \mathcal{H})^{-1}\|_p 2^{-j(s+2/p-4)},$$

where

$$\zeta = \max \{V((u^{n+1})^{(s)}), V(F^{(s)})\},$$

so for  $s \geq 4$  we ensure the convergence when  $j$  goes to the infinity.

*Proof* Subtracting Eq. (34) from (1) yields

$$-L_j \mathcal{H}(u^{n+1} - u_j^{n+1}) = (\mathcal{H} - L_j \mathcal{H})u^{n+1} - (F - F_j),$$

provided that  $\mathcal{H}^{-1}$  exists and bounded, we obtain the error bound

$$\|E_j^{n+1}\|_p = \|(L_j \mathcal{H})^{-1}\|_p \|(\mathcal{H} - L_j \mathcal{H})u^{n+1} - (F - F_j)\|_p, \quad (36)$$

where  $\|(L_j \mathcal{H})^{-1}\|_p = \left\| \left( (1 + \frac{\beta(\delta t)^2}{2})I + \frac{\alpha(\delta t)^2}{2} D_\phi^2 \right)^{-1} \right\|_p$ .

Furthermore, by using Lemma 5 and relation (31), we have

$$\|(\mathcal{H} - L_j \mathcal{H})u^{n+1}\|_p \leq |1 + \frac{\beta(\delta t)^2}{2}| \|(\mathcal{S} - L_j \mathcal{S})u^{n+1}\|_p \quad (37)$$

$$+ | \frac{\alpha(\delta t)^2}{2} | \|(\mathcal{D} - L_j \mathcal{D})u^{n+1}\|_p \quad (38)$$

$$\leq C_1 | \frac{\alpha(\delta t)^2}{2} | 2^{-j(s+2/p-4)} V((u^{n+1})^{(s)}) \quad (39)$$

$$+ C_2 |1 + \frac{\beta(\delta t)^2}{2}| 2^{-j(s+1/p)} V((u^{n+1})^{(s)}), \quad (40)$$

and

$$\|F - F_j\|_p \leq C_3 2^{-j(s+1/p)} V(F^{(s)}), \quad (41)$$

Substituting Eqs. (40)–(41) in (36), we have

$$\begin{aligned} \|u^{n+1} - u_j^{n+1}\|_p &\leq \|(L_j \mathcal{H})^{-1}\|_p \left[ C_1 | \frac{\alpha(\delta t)^2}{2} | 2^{-j(s+2/p-4)} V((u^{n+1})^{(s)}) \right. \\ &\quad \left. + C_2 |1 + \frac{\beta(\delta t)^2}{2}| 2^{-j(s+1/p)} V((u^{n+1})^{(s)}) + C_3 2^{-j(s+1/p)} V(F^{(s)}) \right]. \end{aligned}$$

By choosing  $C = \max \left\{ C_1 | \frac{\alpha(\delta t)^2}{2} |, C_2 |1 + \frac{\beta(\delta t)^2}{2}|, C_3 \right\}$  finally we can obtain

$$\|u^{n+1} - u_j^{n+1}\|_p \leq C \zeta \|(L_j \mathcal{H})^{-1}\|_p 2^{-j(s+2/p-4)}.$$

*Remark 7* We know that the order of accuracy for the Crank-Nicolson method is  $O(\delta t^2)$ . Therefore total error bound can be represented as

$$\|u - u_j^{n+1}\|_p \leq C \zeta \|(L_j \mathcal{H})^{-1}\|_p 2^{-j(s+2/p-4)} + O(\delta t^2).$$

## 5 Numerical Examples

In this section, we give some computational results of numerical experiments with method based on applying the technique discussed in Sect. 3 to find numerical solution of nonlinear Klein-Gordon equation and compare our results with exact solutions

and those already available in literature [7, 12]. In order to test the accuracy of the presented method we use the error norms  $L_2$ ,  $L_\infty$  and Root-Mean-Square (RMS) through the examples. The numerical computations have been done by the software Matlab.

*Example 8* As the first test problem, we consider the nonlinear Klein-Gordon equation (1) with quadratic nonlinearity as

$$\frac{\partial^2 u}{\partial t^2}(x, t) + \alpha \frac{\partial^2 u}{\partial x^2}(x, t) + \beta u(x, t) + \gamma u^2(x, t) = -x \cos(t) + x^2 \cos^2(t).$$

The provided parameters are  $\alpha = -1$ ,  $\beta = 0$  and  $\gamma = 1$  in the interval  $[-1, 1]$  and the initial conditions are given by

$$\begin{cases} u(x, 0) = x, & x \in [-1, 1], \\ u_t(x, 0) = 0, & x \in [-1, 1], \end{cases}$$

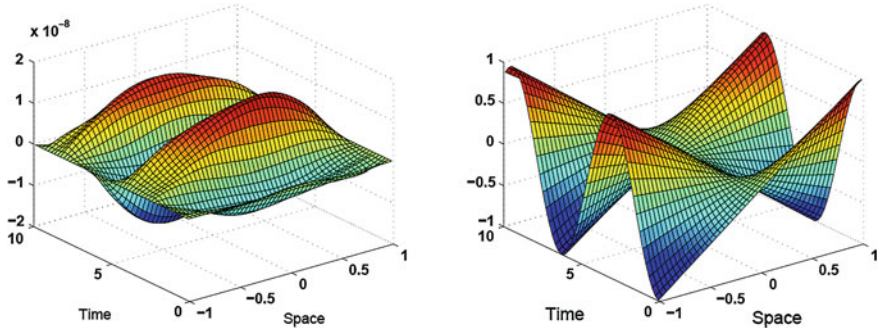
with the Dirichlet boundary condition

$$u(x, t) = h(x, t).$$

The analytical solution is given in [7] as  $u(x, t) = x \cos(t)$ . The prescribed Dirichlet boundary function  $h(x, t)$  can be extracted from the exact solution. The  $L_2$ ,  $L_\infty$  and RMS errors by applying method discussed in Sect. 3 for  $j = 3$ , in different times and  $\delta t = 0.0001$  are presented in Table 1 and compared with the RBFs method proposed in [7]. As it can be shown from Table 1, our method (PWM) is more accurate than RBFs method while PWM uses much less number of grid points (9 grid points) in comparison with RBFs which uses 100 grid points. Figure 1, shows the graph of errors in the computed solution and approximate solution for  $\delta t = 0.0001$  and  $j = 3$ . The graph of errors in the computed solution for different values of time and  $\delta t = 0.0001$ ,  $j = 3$  are plotted in Fig. 2.

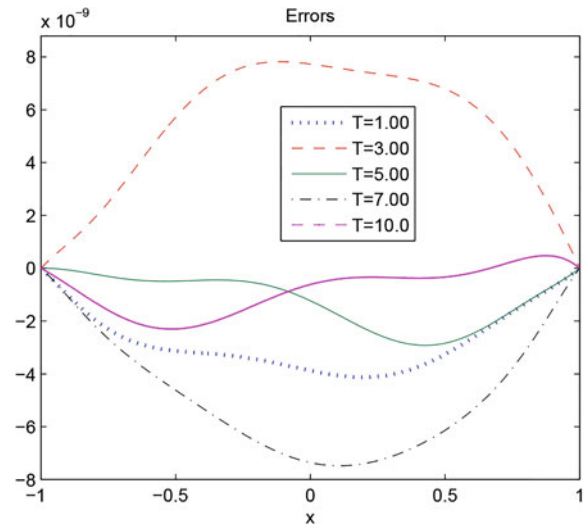
**Table 1**  $L_2$ ,  $L_\infty$  and RMS errors for  $j = 3$  and  $\delta t = 0.0001$  compared with [7], Example 1

$t$	$L_\infty$ -error		$L_2$ -error		RMS-error	
	PWM (9)	RBFs (100)	PWM (9)	RBFs (100)	PWM (9)	RBFs (100)
1.0	$4.12 \times 10^{-9}$	$1.25 \times 10^{-5}$	$1.27 \times 10^{-8}$	$6.54 \times 10^{-5}$	$4.23 \times 10^{-9}$	$6.50 \times 10^{-6}$
3.0	$7.81 \times 10^{-9}$	$1.55 \times 10^{-5}$	$2.50 \times 10^{-8}$	$1.17 \times 10^{-4}$	$8.34 \times 10^{-9}$	$1.16 \times 10^{-5}$
5.0	$2.91 \times 10^{-9}$	$3.37 \times 10^{-5}$	$6.55 \times 10^{-9}$	$2.20 \times 10^{-4}$	$2.18 \times 10^{-9}$	$2.19 \times 10^{-5}$
7.0	$7.47 \times 10^{-9}$	$3.77 \times 10^{-5}$	$2.25 \times 10^{-8}$	$2.58 \times 10^{-4}$	$7.53 \times 10^{-9}$	$2.57 \times 10^{-5}$
10	$2.28 \times 10^{-9}$	$1.30 \times 10^{-5}$	$4.99 \times 10^{-9}$	$7.98 \times 10^{-5}$	$1.66 \times 10^{-9}$	$7.94 \times 10^{-6}$



**Fig. 1** Plot of errors (left) and approximate solution (right) with  $\delta t = 0.0001$ ,  $j = 3$ , example 1

**Fig. 2** Errors graph for Example 1, with  $j = 3$ ,  $\delta t = 0.0001$  and different times



*Example 9* This illustrated example presents the nonlinear Klein-Gordon equation (1) and cubic nonlinearity as

$$\begin{aligned} & \frac{\partial^2 u}{\partial t^2}(x, t) + \alpha \frac{\partial^2 u}{\partial x^2}(x, t) + \beta u(x, t) + \gamma u^3(x, t) \\ &= (x^2 - 2) \cosh(x + t) - 4x \sinh(x + t) + x^6 \cosh^3(x + t). \end{aligned}$$

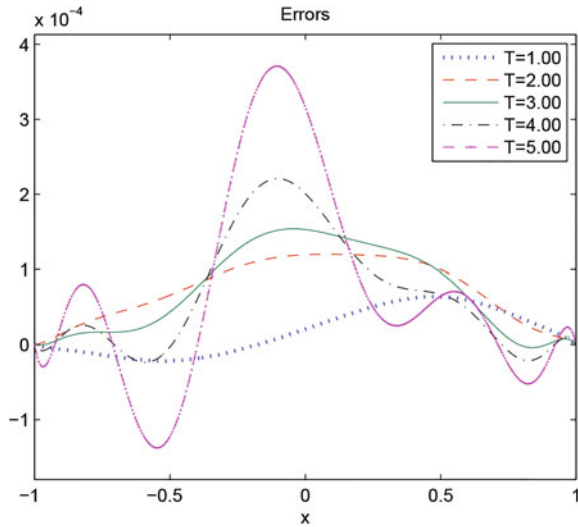
The provided parameters are  $\alpha = -1$ ,  $\beta = 1$  and  $\gamma = 1$  in the interval  $[-1, 1]$  with the initial conditions are given by

$$\begin{cases} u(x, 0) = x^2 \cosh(x + t), & x \in [-1, 1], \\ u_t(x, 0) = x^2 \sinh(x + t), & x \in [-1, 1], \end{cases}$$

**Table 2**  $L_2, L_\infty$  and RMS errors for  $j = 3$  and  $\delta t = 0.0001$  compared with [12], Example 2

$t$	$L_\infty$ -error		$L_2$ -error		RMS-error	
	PWM (9)	LWSCM (24)	PWM (9)	LWSCM (24)	PWM (9)	LWSCM (24)
1.0	$6.38 \times 10^{-5}$	$9.45 \times 10^{-5}$	$1.47 \times 10^{-4}$	$1.79 \times 10^{-4}$	$4.90 \times 10^{-5}$	$3.66 \times 10^{-5}$
2.0	$1.19 \times 10^{-4}$	$9.79 \times 10^{-4}$	$3.55 \times 10^{-4}$	$2.06 \times 10^{-3}$	$1.18 \times 10^{-4}$	$4.22 \times 10^{-4}$
3.0	$1.52 \times 10^{-4}$	$3.97 \times 10^{-3}$	$3.91 \times 10^{-4}$	$7.91 \times 10^{-3}$	$1.30 \times 10^{-4}$	$1.61 \times 10^{-3}$
4.0	$2.20 \times 10^{-4}$	$1.29 \times 10^{-2}$	$4.34 \times 10^{-4}$	$2.44 \times 10^{-2}$	$1.44 \times 10^{-4}$	$4.98 \times 10^{-3}$
5.0	$3.40 \times 10^{-4}$	$3.72 \times 10^{-2}$	$4.49 \times 10^{-4}$	$6.99 \times 10^{-2}$	$2.16 \times 10^{-4}$	$1.42 \times 10^{-2}$

**Fig. 3** Errors graph for Example 2, with  $j = 3$ ,  $\delta t = 0.0001$  and different times



with the Dirichlet boundary condition

$$u(x, t) = h(x, t).$$

The analytical solution is given in [7, 12] as  $u(x, t) = x^2 \cosh(x + t)$ . We extract the boundary function  $h(x, t)$  from the exact solution. The  $L_2, L_\infty$  and RMS errors by applying method discussed in Sect. 3 for  $j = 3$ , in different times and  $\delta t = 0.0001$  are presented in Table 2 and compared with the Legendre wavelets spectral collocation method (LWSCM) [12]. As it can be seen from Table 2, our method is accurate than LWSCM while PWM uses 9 number of Polynomial wavelet basis functions in comparison with LWSCM which uses 24 number of Legendre wavelet basis. On the other hand, the accuracy of our results remains consistent when the time increases, which is the advantage of using PWM, but in the case of LWSCM the accuracy decreases fastly. Figure 3, shows the graph of errors in the computed solution for

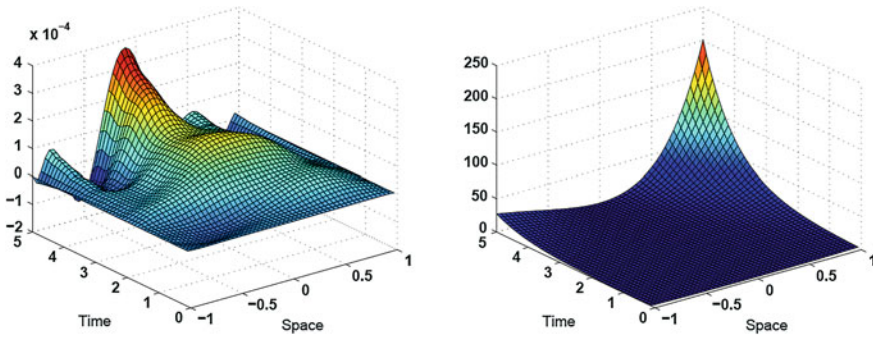


Fig. 4 Plot of errors (left) and approximate solution (right) with  $\delta t = 0.0001$ ,  $j = 3$ , example 2

different values of time and  $\delta t = 0.0001$ ,  $j = 3$ . The graph of errors in the computed solution and approximate solution for  $\delta t = 0.0001$  and  $j = 3$  are plotted in Fig. 4.

## 6 Conclusion

A numerical method was employed successfully for the Nonlinear Klein-Gordon equation. This approach is based on the Crank-Nicolson method for temporal discretization and the Wavelet-Collocation method in the spatial direction. After temporal discretization, the operational matrix of derivative along with a collocation method, is used to reduce the considered problem to the corresponding systems of algebraic equations at each time steps. One of the advantages of using polynomial wavelets is that the effort required to implement the method is very low, while the accuracy is high. The convergence analysis is developed. The method is computationally attractive and applications are demonstrated through illustrative examples.

## References

1. Ablowitz, M.J., Clarkson, P.A.: Solitons, Nonlinear Evolution Equations and Inverse Scattering. Cambridge University Press, Cambridge (1990)
2. Dodd, R.K., Eilbeck, J.C., Gibbon, J.D., Morris, H.C.: Solitons and Nonlinear Wave Equations. Academic, London (1982)
3. Chowdhury, M.S.H., Hashim, I.: Application of homotopy-perturbation method to Klein-Gordon and sine-Gordon equations. Chaos, Solitons Fractals **39**(4), 1928–1935 (2009)
4. Jimenez, S., Vazquez, L.: Analysis of four numerical schemes for a nonlinear Klein-Gordon equation. Appl. Math. Comput. **35**, 61–94 (1990)
5. Deeba, E., Khuri, S.A.: A decomposition method for solving the nonlinear Klein-Gordon equation. J. Comput. Phys. **124**, 442–448 (1996)
6. Khuri, S.A., Sayfy, A.: A spline collocation approach for the numerical solution of a generalized nonlinear Klein-Gordon equation. Appl. Math. Comput. **216**, 1047–1056 (2010)

7. Dehghan, M., Shokri, A.: Numerical solution of the nonlinear Klein-Gordon equation using radial basis functions. *J. Comput. Appl. Math.* **230**, 400–410 (2009)
8. Rashidinia, J., Ghasemi, M., Jalilian, R.: Numerical solution of the nonlinear Klein-Gordon equation. *J. Comput. Appl. Math.* **233**, 1866–1878 (2010)
9. Lakestani, M., Dehghan, M.: Collocation and finite difference-collocation methods for the solution of nonlinear Klein-Gordon equation. *Comput. Phys. Commun.* **181**, 1392–1401 (2010)
10. Wong, Y.S., Chang, Q., Gong, L.: An initial-boundary value problem of a nonlinear Klein-Gordon equation. *Appl. Math. Comput.* **84**, 77–93 (1997)
11. Rashidinia, J., Mohammadi, R.: Tension spline approach for the numerical solution of nonlinear Klein-Gordon equation. *Comput. Phys. Commun.* **181**, 78–91 (2010)
12. Yin, F., Tian, T., Song, J., Zhu, M.: Spectral methods using Legendre wavelets for nonlinear Klein/Sine-Gordon equations. *J. Comput. Appl. Math.* **275**, 321–334 (2015)
13. Maleknejad, K., Khademi, A.: Filter matrix based on interpolation wavelets for solving Fredholm integral equations. *Commun. Nonlinear. Sci. Numer. Simulat.* **16**, 4197–4207 (2011)
14. Kilgore, T., Prestin, J.: Polynomial wavelets on the interval. *Constr. Approx.* **12**, 95–110 (1996)
15. Rashidinia, J., Jokar, M.: Application of polynomial scaling functions for numerical solution of telegraph equation. *Appl. Anal.* (2015). doi:[10.1080/00036811.2014.998654](https://doi.org/10.1080/00036811.2014.998654)
16. Lakestani, M., Razzaghi, M., Dehghan, M.: Semiorthogonal spline wavelets approximation for Fredholm integro-differential equations. *Math. Probl. Eng.* 1–12 (2006)
17. Prestin, J.: Mean convergence of Lagrange interpolation. *Seminar Analysis, Operator Equation and Numerical Analysis 1988/89*, pp. 75–86. Karl-WeierstraB-Institut für Mathematik, Berlin (1989)