Galerkin and Collocation Methods for the Solution of Klein-Gordon Equation Using Interpolating Scaling Functions

Saba Izadkhah\textsuperscript{1}, Mohammad Shahriari\textsuperscript{2}, Behzad Nemati Saray\textsuperscript{3,*}

\textsuperscript{1} Department of Computer, Faculty of Computer Science University of Tabriz, Tabriz, Iran
\textsuperscript{2} Department of Applied Mathematics, Faculty of Sciences, University of Maragheh, Maragheh, Iran
\textsuperscript{3} Department of Applied Mathematics, Faculty of Mathematical Science University of Tabriz, Tabriz, Iran

(Received 6 June 2012, accepted 12 August 2013)

Abstract: A numerical technique is presented for the solution of Klein-Gordon equation. This method uses interpolating scaling functions. The method consists of expanding the required approximate solution as the elements of interpolating scaling functions. Using the operational matrix of derivatives, we reduce the problem to a set of algebraic equations. Some numerical examples are included to demonstrate the validity and applicability of the technique. The method is easy to implement and produces accurate results.

Keywords: Galerkin method; Klein-Gordon equation; interpolating scaling function; operational matrix of derivative.

1 Introduction

In this article, we present numerical scheme for solving initial-value problem of the one-dimensional nonlinear Klein-Gordon equation as

\[ u_{tt} + \alpha u_{xx} + g(u) = f(x, t), \] (1)

with

\[ u(x, 0) = g_1(x), \quad u_t(x, 0) = g_2(x), \] (2)

\[ u(0, t) = h_1(x), \quad u(1, t) = h_2(t), \]

where \( u = u(x, t) \) represents the wave displacement at \((x, t)\) and \( \alpha \) is a known constant and \( g(u) \) is the nonlinear force. In the well-known sine-Gordon equation, the nonlinear force is given by \( g(u) = \sin u \). In the physical applications, the nonlinear force \( g(u) \) has also other forms [10, 12]. The cases \( g(u) = \sin u + \sin 2u \) and \( g(u) = \sinh u + \sinh 2u \) are called the double sine-Gordon equation and the double sinh-Gordon equation, respectively [5, 11]. We note that one key feature is that the Klein-Gordon equation is a Hamiltonian PDE, and for a wide class of functions \( g(u) \), it has conserved Hamiltonian (or energy)

\[ H = \int \left( \frac{1}{2} u_t^2 + \frac{1}{2} u_x^2 + G(u) \right) dx, \]

where \( G(u) = g(u) \).

Nonlinear partial differential equations (NLPDEs) arise in many fields of science, particularly in physics, engineering, chemistry and finance, and are fundamental for the mathematical formulation of continuum models. The nonlinear Klein-Gordon equation appears in many types of nonlinearities. The Klein-Gordon equation plays an important role in mathematical physics [10, 11, 15, 26]. The equation has attracted much attention in studying solitons and condensed matter physics [7], in investigating the interaction of solitons in a collisionless plasma, the recurrence of initial states, and in examining the nonlinear wave equations [13]. Eq. (1) is one of the important mathematical models in quantum mechanics [22] and it also occurs in relativistic physics as a model of dispersive phenomena [8].

Many papers have been published for solving Klein-Gordon equation. For example, Tension spline approach [21] and Collocation and finite difference-collocation methods for solving nonlinear Klein-Gordon equation with quadratic and

\*Corresponding author. E-mail address: b_nemati@tabrizu.ac.ir

Copyright © World Academic Press, World Academic Union

ISSN 1749-3889 (print), 1749-3897 (online)

International Journal of Nonlinear Science

Vol.16 (2013) No.2, pp.113-124
cubic nonlinearity [18]. Several second-order of finite difference schemes is in [20]. We note that alternative approaches using spectral and pseudo-spectral methods have recently been presented in [16, 19]. Dehghan and Ghesmati [9] developed a technique based on the dual reciprocity boundary element method to obtain the approximate solution of the nonlinear Klein-Gordon equation. Note that the solution of the nonlinear Klein-Gordon equation in unbounded domain is also investigated in [27]. The interested reader can see [4, 6, 14, 25] for more works on this equation.

In this paper interpolating scaling functions (ISF) are constructed. These bases are used to construct Alpert multi- wavelets [2, 3]. Operational matrix of derivative is derived in [17, 23]. In the present work we use interpolation property and solve the Klein-Gordon equation. Interpolating scaling functions are used to reduce Klein-Gordon equation to a system of nonlinear equation and Newton method are applied to solve the system at nonlinear equation. Recently Lakestani and Nemati Saray used these bases to solve Telegraph equation[17] and nonlinear generalized Burgers-Huxley equation [12].

2 Interpolating scaling functions

Suppose $P_r$ is the Legendre polynomial of order $r$ and $r$ is any fixed nonnegative integer number. Let $\tau_k$ denotes the roots of $P_r$ for $k = 0, \ldots, r - 1$. The interpolating scaling functions (ISF) are given by [23]

$$\phi^k(t) := \begin{cases} \sqrt{\frac{2}{\omega_k}} L_k(2t - 1), & t \in [0, 1], \\ 0, & \text{otherwise}, \end{cases}$$

where $\omega_k$ is the Gauss-Legendre quadrature weight

$$\omega_k = \frac{2}{r P_r(\tau_k) P_r - 1(\tau_k)},$$

and $L_k(t)$ is the Lagrange interpolating polynomial defined by [2]

$$L_k(t) = \prod_{i=0, i \neq k}^{r-1} \left( \frac{t - \tau_i}{\tau_k - \tau_i} \right).$$

for $k = 0, \ldots, r - 1$. We can expand any polynomial $g$ of degree less than $r$ with the functions with an orthonormal basis on $[0, 1]$ as following,

$$g(t) = \sum_{k=0}^{r-1} d_k \phi^k(t),$$

where the coefficients are given by

$$d_k = \sqrt{\frac{\omega_k}{2}} g(\hat{\tau}_k), \quad k = 0, \ldots, r - 1,$$

and

$$\hat{\tau}_k = \frac{\tau_k + 1}{2}.$$

Let $\phi^k_{nl}(t)$ defined by

$$\phi^k_{nl}(t) = 2^{(n/2)} \phi^k(2^n t - l)$$

where $k = 0, \ldots, r - 1$, $l = 0, \ldots, 2^n - 1$, and $n$ is any fixed nonnegative integer number. Note that we have the following orthonormality relation

$$\int_0^1 \phi^k_{nl}(t) \phi^k_{ni}(t) dt = \delta_{li} \delta_{kk},$$

$k, \hat{k} = 0, \ldots, r - 1$ and $l, \hat{l} = 0, \ldots, 2^n - 1$. 

IJNS email for contribution: editor@nonlinearscience.org.uk
2.1 Function approximation

For any two fixed nonnegative integer numbers \(r\) and \(n\), a function \(f(t)\) defined over \([0, 1]\) may be represented by ISF expansion as,

\[
f(t) = \sum_{k=0}^{r-1} \sum_{l=0}^{2^n-1} s_{nl}^k \phi_{nl}^k(t) = S^T \Phi(t) \tag{4}
\]

where

\[
S = \left[ s_{n0}, \ldots, s_{n0}^{-1}, |s_{n1}^0, \ldots, s_{n1}^{r-1}|, \ldots |s_{n,2^n-1}^0, \ldots, s_{n,2^n-1}^{r-1} \right]^T,
\]

\[
\Phi(t) = \left[ \phi_{n0}^0(t), \ldots, \phi_{n0}^{r-1}(t)|\phi_{n1}^0(t), \ldots, \phi_{n1}^{r-1}(t)| \ldots |\phi_{n,2^n-1}^0(t), \ldots, \phi_{n,2^n-1}^{r-1}(t) \right]^T,
\]

and the coefficients \(e_{nl}^k\) are computed by

\[
s_{nl}^k = \int_0^1 f(t) \phi_{nl}^k(t) dt = \int_{h_l}^{h_{l+1}} f(t) \phi_{nl}^k(t) dt,
\]

where

\[
h_l = \frac{l}{2^n} \quad \text{and} \quad l = 0, \ldots, 2^n - 1.
\]

These coefficients may be computed using Gauss-Legendre quadrature \([12, 17]\).

\[
s_{nl}^k = 2^{-n/2} \sqrt{\frac{\omega_k}{2}} f(2^{-n}(\hat{\tau}_k + l)), \quad k = 0, \ldots, r - 1, \quad l = 0, \ldots, 2^n - 1. \tag{6}
\]

Also a function \(g(x, t)\) of two independent variables for \(0 \leq x \leq 1\), and \(0 \leq t \leq 1\), may be expanded in terms of interpolating scaling functions as

\[
g(x, t) = \sum_{i=1}^N \sum_{j=1}^N g_{ij} \Phi_i(x) \Phi_j(t) = \Phi^T(x) G \Phi(t), \tag{7}
\]

such that \(G\) is an \(N \times N\) matrix as

\[
G = \left[ \begin{array}{ccc}
g_{11} & \cdots & g_{1N} \\
\vdots & \ddots & \vdots \\
g_{N1} & \cdots & g_{NN} \end{array} \right],
\]

\(N = r 2^n\), and

\[
g_{ij} = \int_0^1 \int_0^1 g(x, t) \Phi_i(x) \Phi_j(t) dx dt.
\]

We used two step of (6) to obtain

\[
g_{i,j} = 2^{-n} \sqrt{\frac{\omega_k}{2}} \sqrt{\frac{\omega_{k'}}{2}} g \left( 2^{-n}(\hat{\tau}_k + l), 2^{-n}(\hat{\tau}_{k'} + l') \right), \tag{8}
\]

where \(i = rl + (k + 1), j = rl' + (k' + 1), k = 0, \ldots, r - 1\) and \(l = 0, \ldots, 2^n - 1\).

2.2 The operational matrix of derivative

Let the derivative of \(f(t)\) in Eq. (4) be given by

\[
\frac{d}{dt} f(t) = \sum_{k=0}^{r-1} \sum_{l=0}^{2^n-1} s_{nl}^k \phi_{nl}^k(t) = \tilde{S}^T \Phi(t), \tag{9}
\]

where \(\tilde{S}\) is a vector defined similarly to (5). We express relation between \(S\) and \(\tilde{S}\) by

\[
\tilde{S} = DS, \tag{10}
\]
where $D$ is the operational matrix for derivatives of the scaling function. By using (9) we get [17, 23]

$$\tilde{s}_{nl}^k = \int_{h_l}^{h_{l+1}} \phi_{nl}(t) \left( \frac{d}{dt} \phi(t) \right) dt. \quad k = 0, \ldots, r - 1, \ l = 0, \ldots, 2^n - 1,$$

Using integration by parts, we obtain

$$\tilde{s}_{nl}^k = \left[ f(t) \phi_{nl}(t) \right]_{h_l}^{h_{l+1}} - \int_{h_l}^{h_{l+1}} f(t) \left( \frac{d}{dt} \phi_{nl}(t) \right) dt.$$

From (3) and (4) we get

$$\tilde{s}_{nl}^k = 2^{(n/2)} \left[ f(h_{l+1}) \phi^k(1) - f(h_l) \phi^k(0) \right] - 2^n \sum_{i=0}^{r-1} q_{ki} \tilde{s}_{nl}^i,$$

where

$$q_{ki} = \int_0^1 \phi^i(t) \left( \frac{d}{dt} \phi^k(t) \right) dt.$$

Employing the Gauss-Legendre quadrature formula, we obtain

$$q_{ki} = \sqrt{\frac{\omega_i}{2}} \frac{d}{dt} \phi^k(\tilde{\tau}_i).$$

To evaluate $f(h_l)$ and $f(h_{l+1})$ we use the average of left and right limit on $h_l$ as

$$f(h_l) = \frac{1}{2} \left( \sum_{i=0}^{r-1} s_{n,l-1}^i \phi_{n,l-1}(h_l) + \sum_{i=0}^{r-1} s_{nl}^i \phi_{nl}(h_l) \right), \ l = 1, \ldots, 2^n - 1.$$

Using (3), we can express (12) as

$$f(h_l) = 2^{n/2} \frac{1}{2} \left( \sum_{i=0}^{r-1} s_{n,l-1}^i \phi^i(1) + \sum_{i=0}^{r-1} s_{nl}^i \phi^i(0) \right).$$

Also, to evaluate the values of functions $f$ at the point $h_0$ and $h_{2^n}$ we have

$$f(h_0) = \sum_{i=0}^{r-1} s_{i0}^i \phi_{i0}(h_0) = 2^{n/2} \sum_{i=0}^{r-1} s_{i0}^i \phi^i(0),$$

$$f(h_{2^n}) = \sum_{i=0}^{r-1} s_{i2^n-1}^i \phi_{i2^n-1}(h_{2^n}) = 2^{n/2} \sum_{i=0}^{r-1} s_{i2^n-1}^i \phi^i(1).$$

Substituting (13)-(15) in (11), we obtain

$$\tilde{s}_{n0}^k = 2^n \left[ \sum_{i=0}^{r-1} \left( \frac{1}{2} \phi^i(1) \phi^k(1) - \phi^i(0) \phi^k(0) - q_{ki} \right) s_{i0}^i + \sum_{i=0}^{r-1} \frac{1}{2} \phi^i(0) \phi^k(1) s_{n1}^i \right],$$

$$\tilde{s}_{nl}^k = 2^n \left[ \sum_{i=0}^{r-1} \left( - \frac{1}{2} \phi^i(1) \phi^k(0) \right) s_{n,l-1}^i + \sum_{i=0}^{r-1} \left( \frac{1}{2} \phi^i(1) \phi^k(1) \right) \right.$$

$$\left. - \frac{1}{2} \phi^i(0) \phi^k(0) - q_{ki} \right) s_{nl}^i + \sum_{i=0}^{r-1} \left( \frac{1}{2} \phi^i(0) \phi^k(1) s_{n,l+1}^i \right],$$

$$l = 0, \ldots, 2^n - 2,$$

$$\tilde{s}_{n,2^n-1}^k = 2^n \left[ \sum_{i=0}^{r-1} \left( - \frac{1}{2} \phi^i(1) \phi^k(0) s_{n,2^n-2}^i + \sum_{i=0}^{r-1} \left( \phi^i(1) \phi^k(1) \right) \right. \right.$$

$$\left. - \frac{1}{2} \phi^i(0) \phi^k(0) - q_{ki} \right) s_{n,2^n-1}^i + \sum_{i=0}^{r-1} \left( \frac{1}{2} \phi^i(0) \phi^k(1) s_{n,2^n}^i \right],$$

$$n = 0, \ldots, 2^n - 1,$$
\[-\frac{1}{2} \phi'(0)\phi^k(0) - q_{ki} \}\right].

From the above equations the matrix $D$ can be expressed as a block tridiagonal matrix which is obtained from

$$D = 2^n \begin{bmatrix}
R & H & & \\
-H^T & R & H & \\
& \ddots & \ddots & \ddots \\
& & -H^T & R & H \\
& & & -H^T & R
\end{bmatrix},$$

here, each block is an $r \times r$ matrix and for $k, i = 1, \ldots, r$, we have

$$[R]_{ki} = \frac{1}{2} \phi^i(1)\phi^k(1) - \phi^i(0)\phi^k(0) - q_{ki},$$

$$[R]_{ki} = \frac{1}{2} \phi^i(1)\phi^k(1) - \frac{1}{2} \phi^i(0)\phi^k(0) - q_{ki},$$

$$[R]_{ki} = \phi^i(1)\phi^k(1) - \frac{1}{2} \phi^i(0)\phi^k(0) - q_{ki},$$

$$[H]_{ki} = \frac{1}{2} \phi^i(0)\phi^k(1).$$

Since (11)-(15) are exact for polynomials up to degree $r - 1$, the operational matrix of derivative is exact for polynomials up to degree $r - 1$.

3 Description of numerical method

We consider Klein-Gordon equation which has the form

$$u_{tt} + \alpha u_{xx} + g(u) = f(x, t), \quad x \in \Omega = [0, 1] \subset R, 0 < t \leq 1,$$

with the initial and boundary conditions

$$u(x, 0) = g_1(x), \quad u_t(x, 0) = g_2(x), \quad x \in \Omega,$$

$$u(0, t) = h_1(x), \quad u(1, t) = h_2(t), \quad x \in \delta \Omega, 0 < t \leq 1.$$

3.1 Galerkin method based on interpolating scaling function (GCM)

The solution $u(x, t)$ of (16) can be approximated as

$$u(x, t) \simeq \Phi^T(t)U\Phi(x).$$

By using (9), we obtain

$$u_t(x, t) \simeq \Phi^T(t)DU\Phi(x), \quad u_{tt}(x, t) \simeq \Phi^T(t)D^2U\Phi(x),$$

Also

$$u_{xx}(x, t) = \Phi^T(t)UD^2\Phi(x).$$

We suppose that

$$\hat{g}(x, t) = g(u(x, t)),$$

Using (9), we can approximate $\hat{g}(x, t)$ as the following form

$$\hat{g}(x, t) \simeq \Phi^T(t)G\Phi(x),$$

IJNS homepage: http://www.nonlinearscience.org.uk/
where \( G \) is a \((N \times N)\) matrix. The entries of this matrix are obtained by

\[
G_{i,j} = 2^{-n} \sqrt{\frac{\omega_k}{2}} \sqrt{\frac{\omega_{k'}}{2}} \phi \(2^{-n}(\hat{r}_k + l), 2^{-n}(\hat{r}_{k'} + l')\), \quad i = rl + (k + 1), j = rl' + (k' + 1).
\]

also the function \( f(x, t) \) is

\[
f(x, t) = \Phi(T(t))F\Phi(x).
\]

using the (20), (21) and (23), we obtain

\[
\Phi(T(t))D^2U\Phi(x) + \alpha\Phi(T(t))UD^2\Phi(x) + \Phi(T(t))G\Phi(x) = \Phi(T(t))F\Phi(x).
\]

The entries of vectors \( \Phi(t) \) and \( \Phi(x) \) are independent, so from (24) we get

\[
\Upsilon = D^2U + \alpha UD^2 + G - F.
\]

One can obtain the equation (25) has \((N - 1) \times (N - 2)\) independent equations, because the rank of \( D^2 \) is \( N - 2 \). Here we choose

\[
\Upsilon_{i,j} = 0 \quad \text{for} \quad i = 2, \ldots, N - 1, \quad j = 2, \ldots, N - 1.
\]

Using (4), we can approximate the functions \( g_1(x), g_2(x), h_1(t) \) and \( h_2(t) \) as:

\[
\begin{align*}
g_1(x) &= X_1^T\Phi(x), \\
g_2(x) &= X_2^T\Phi(x), \\
h_1(t) &= \Phi(T(t))X_3, \\
h_2(t) &= \Phi(T(t))X_4,
\end{align*}
\]

where \( X_1, X_2, X_3 \) and \( X_4 \) are vectors of dimension \( N \), and can be found as

\[
\begin{align*}
X_{1i} &= 2^{-\frac{n}{2}} \sqrt{\frac{\omega_k}{2}} g_1(2^{-n}(\hat{r}_k + l)), \\
X_{2i} &= 2^{-\frac{n}{2}} \sqrt{\frac{\omega_k}{2}} g_2(2^{-n}(\hat{r}_k + l)), \\
X_{3i} &= 2^{-\frac{n}{2}} \sqrt{\frac{\omega_k}{2}} h_1(2^{-n}(\hat{r}_k + l)), \\
X_{4i} &= 2^{-\frac{n}{2}} \sqrt{\frac{\omega_k}{2}} h_2(2^{-n}(\hat{r}_k + l)),
\end{align*}
\]

for

\[
l = \frac{i - k - 1}{r}, \quad k = 0, \ldots, r - 1, \quad i = 1, \ldots, N.
\]

Applying (19) in the initial and boundary conditions (17) and (18), we get

\[
\Phi(T(0))U\Phi(x) = X_1^T\Phi(x),
\]

\[
\Phi(T(0))DU\Phi(x) = X_2^T\Phi(x),
\]

\[
\Phi(T(t))U\Phi(0) = \Phi(T(t))X_3,
\]

\[
\Phi(T(t))U\Phi(1) = \Phi(T(t))X_4.
\]

The entries of vector \( \Phi(x) \) and \( \Phi(t) \) are independent, so (28) gives

\[
\begin{align*}
\Phi(T(0))U &= X_1^T, \\
D\Phi(T(0))U &= X_2^T, \\
U\Phi(0) &= X_3.
\end{align*}
\]
\[ U \Phi(1) = X_4. \]

Assume that
\[
\begin{align*}
\Omega_1 &= \Phi^T(0)U - X_1^T, \\
\Omega_2 &= D\Phi^T(0)U - X_2^T, \\
\Omega_3 &= U\Phi(0) - X_3, \\
\Omega_4 &= U\Phi(1) - X_4.
\end{align*}
\]

By choosing
\[
\begin{align*}
\Omega_1 &= 0, & i &= 2, \ldots, N - 1, \\
\Omega_2 &= 0, & i &= 2, \ldots, N - 1, \\
\Omega_3 &= 0, & i &= 1, \ldots, N, \\
\Omega_4 &= 0, & i &= 1, \ldots, N.
\end{align*}
\]

From (26) and (31) we have \( N^2 \) nonlinear equations, which can be solved for \( U_{ij}, i, j = 1, \ldots, N \). This nonlinear system of equations solve by Newton method and we will get the approximated solution of the Klein-Gordon equation.

### 3.2 Collocation method interpolating scaling functions (CCM)

In this method, \( \Upsilon \) in (25) is given by another manner. For applying collocation method, we need to have some collocation points. These points are given by \( x_i = \frac{i}{N-1} \) for \( i = 0, \ldots, N - 1 \). By putting (20)-(22) in equation (16), we have
\[
\Phi^T(t)D^2U\Phi(x) + \alpha\Phi^T(t)UD^2\Phi(x) + g(\Phi^T(t)U\Phi(x)) - \Phi^T(t)F\Phi(x) = 0,
\]
(32)

Using collocation points to obtain \( N^2 \) equations but we have \( (N - 2) \times (N - 2) \) independent equations because the rank of \( D^2 \) is \( N - 2 \). By interplaiting (27) in these collocation points, we give \( N^2 \) equations. These system of equations are nonlinear and Newton method is used to solve them.

### 4 Test problems

In this section we give some computational results of numerical experiments with methods based on preceding section, to support our theoretical discussion. To show the efficiency of the present method for our problems in comparison with the exact solution, we report norm infinity and the \( L_2 \) errors of the solution which are defined by
\[
L_\infty = \max_{0 \leq i \leq 10} |u_i - \tilde{u}_i|
\]
and
\[
L_2 = \left( \int_0^1 |u_i - \tilde{u}_i|^2 dx \right)^{\frac{1}{2}}
\]
where \( t_i = \frac{i}{10}, i = 0, \ldots, 10 \). Also \( u_i \) and \( \tilde{u}_i \) are the exact and computed values of the solution \( u \) at point \( t_i \).

**Example 1.** Consider the following Klein-Gordon equation
\[
u_{tt} - u_{xx} + u^2 = 6xt(x^2 - t^2) + x^6t^6.
\]
(33)
The analytical solution is given in [18] as
\[
u(x, t) = x^2t^3,
\]
also the initial and boundary conditions are given by
\[
\begin{align*}
&u(x, 0) = 0, & &u_t(x, 0) = 0, & &0 \leq x \leq 1, \\
&u(0, t) = 0, & &u(1, t) = t^3, & &0 \leq t \leq 1.
\end{align*}
\]

Table 1 consist of norm infinity and \( L_2 \) norm of example 1 for \( n = 1, 2 \). Also we show that the methods represented in this paper is the better than the collocation method used in [18].

Example 2. Consider the following Klein-Gordon equation \((1)\)

\[ u_{tt} - \frac{5}{2}u_{xx} + u + \frac{3}{2}u^3 = 0, \quad x \in [0, 1], \, t > 0, \]  

with initial conditions

\[ u(x, 0) = \sqrt{\frac{2}{3}} \tan \left( \sqrt{\frac{-1}{2c^2 - 5}} x \right), \]

\[ u_t(x, 0) = \sqrt{\frac{-2}{3(2c^2 - 5)}} \sec^2 \left( \sqrt{\frac{-1}{2c^2 - 5}} x \right), \quad x \in [0, 1], \]

and Dirichlet boundary conditions

\[ u(0, t) = \sqrt{\frac{2}{3}} \tan \left( \sqrt{\frac{-1}{2c^2 - 5}} ct \right), \]

\[ u(1, t) = \sqrt{\frac{2}{3}} \tan \left( \sqrt{\frac{-1}{2c^2 - 5}} (1 + ct) \right), \quad t > 0. \]

The exact solution for this problem is

\[ \sqrt{\frac{2}{3}} \tan \left( \sqrt{\frac{-1}{2c^2 - 5}} (x + ct) \right). \]

We applied our Methods to solve this problem. The computed solution and exact solution at grid points are compared, the observed errors, \(L_\infty, L_2\) are tabulated in Table 2 for different times and absolute errors are showed in figure 1.
from the exact solution. The exact solution of this equation is

\[ u(x,t) = B \tan(Kx), \quad 0 \leq x \leq 1, \]

and the exact solution by [11, 18] is

\[ u(x,t) = BrK \sec^2(kx) \quad 0 \leq x \leq 1, \]

where \( B = \sqrt{\frac{\beta}{\gamma}} \) and \( K = \sqrt{\frac{-\beta}{2(\alpha+c^2)}} \). In this example \( f = 0 \). We extract the boundary conditions from the exact solution.

Table 3 shows \( L_2 \) and \( L_\infty \) errors for presented methods, with \( r = 5, n = 1 \). Fig. 2 show the plot of error using presented methods.

**Example 4.** In this example Sinc-Gordon equation is solved by presented method. Sinc-Gordon equation is a equation with \( g(u) = \sin(u) \) nonlinearity and \( \alpha = -1, c = 0.2 \) and \( f(x,t) = 0 \). We extract the boundary and initial conditions from the exact solution. The exact solution of this equation is

\[ u(x,t) = 4 \arctan \left( \frac{e \sinh \left( \frac{\pi}{\sqrt{1-c^2}} \right)}{\cosh \left( \frac{ct}{\sqrt{1-c^2}} \right)} \right). \]

Fig.3 shows the plots of absolute error.
Figure 2: plot of absolute errors for example 3 with $r = 5$, $n = 1$, left(GCM) and right(CCM)

Figure 3: plot of absolute errors for example 4 with $r = 5$, $n = 1$, left(GCM) and right(CCM)
5 Conclusion

In this paper we presented the numerical schemes for solving the Klein-Gordon equation. This technique is based on the interpolating scaling functions and Galerkin method. The numerical results given in the previous section demonstrate the accuracy of these schemes. The obtained results showed that this technique can solve the problem effectively.

References


