

## Efficient high-order finite difference methods for nonlinear Klein–Gordon equations. I: Variants of the phi-four model and the form-I of the nonlinear Klein–Gordon equation\*

Ascensión Hernández Encinas<sup>a</sup>, Jesus Martín-Vaquero<sup>a</sup>, Araceli Queiruga-Dios<sup>a</sup>, Víctor Gayoso-Martínez<sup>b</sup>

<sup>a</sup>Higher Technical School of Industrial Engineering (ETSII)  
University of Salamanca  
Salamanca, Spain  
ascen@usal.es; jesmarva@usal.es; queirugadios@usal.es

<sup>b</sup>Information Security Institute (ISI)  
Spanish National Research Council (CSIC)  
Madrid, Spain  
victor.gayoso@iec.csic.es

**Received:** January 15, 2014 / **Revised:** September 15, 2014 / **Published online:** January 19, 2015

**Abstract.** In this paper, the problem of solving some nonlinear Klein–Gordon equations (KGEs) is considered. Here, we derive different fourth- and sixth-order explicit and implicit algorithms to solve the phi-four equation and the form-I of the nonlinear Klein–Gordon equation. Stability and consistency of the proposed schemes are studied under certain conditions. Numerical results are presented and then compared with others obtained from some methods already existing in the scientific literature to explain the efficiency of the new algorithms. It is also shown that similar schemes can be proposed to solve many classes of nonlinear KGEs.

**Keywords:** Klein–Gordon equations, finite difference methods, numerical methods, phi-four model.

### 1 Introduction

The nonlinear Klein–Gordon equation (KGE)

$$u_{tt} - u_{xx} + h_1(u) = h_2(x, t); \quad 0 \leq x \leq l, \quad 0 \leq t \leq t_e,$$

where  $u = u(x, t)$  represents the wave displacement at position  $x$  and time  $t$ , and  $h_1(u)$  is the nonlinear force, arises in the study of theoretical physics (see [18, 26] and references

---

\*This work was supported by Fundación Memoria D. Samuel Solórzano Barruso under the projects FS/23-2013 and FS-28/2014.

therein). This equation is also known as the relativistic version of the Schrödinger's equation [5]. It represents the equation of motion of a quantum scalar or a pseudo-scalar field (see [20] and references therein), which is a field whose quanta are spinless particles. The KGE is second order in time and therefore, it cannot be directly interpreted as a Schrödinger equation for a quantum state. Moreover, it does not admit a positive definite conserved probability density. Nevertheless, with a proper interpretation, KGE describes the quantum amplitude for finding a point particle in various places and it also describes the relativistic wave function. Moreover, complex-valued KGE is a special case of Higgs equation that is studied during interaction of scalar nucleons and mesons in particle physics [4].

Many nonlinear Klein–Gordon equations are Hamiltonian partial differential equations (PDEs), and for a wide class of force  $h_1(u)$ , they have the conserved Hamiltonian quantity (or energy)

$$H = \int \left( \frac{u_t^2}{2} + u_x^2 + H_1(u) \right),$$

where  $H_1'(u) = h_1(u)$ .

There has been enormous theoretical studies of the KGE with various potentials. There are numerous papers dealing with the existence and uniqueness of the smooth and weak solutions of these equations or studying their conservation laws (see [3, 10, 19, 21, 26, 27] and references therein). In this paper, we will not study them analytically, but rather we will provide numerical procedures to obtain the solutions of the nonlinear KGE.

The solutions, thus obtained, are of interest in nuclear physics, condensed matter physics, high-energy physics, quantum field theory and many more (see [8, 12, 17, 23] and references therein). The KGE also appears in nonlinear optics, where it plays an important role for Bose–Einstein condensates trapped in strong optical lattices formed by the interference patterns of laser beams. The KGE with vector and scalar potentials can only be solved exactly for certain potentials such as the Coloumb, Morse, or the harmonic oscillator cases, for example. In this context, it provides a good description of the scalar particles. In particular, for the case of pionic bound states in the Coloumb field of the nuclei, the KGE gives a good first order approximation of the problem [2].

In the present work, we deal with the numerical approximation of two of the best known nonlinear Klein–Gordon equations, the phi-four and its variant, the form-I of these nonlinear equations:

$$\begin{aligned} u_{tt} - u_{xx} - u + u^3 &= f(x, t); & 0 \leq x \leq l, 0 \leq t \leq t_e, \\ u_{tt} - u_{xx} - u + u^m &= f(x, t); & 0 \leq x \leq l, 0 \leq t \leq t_e. \end{aligned} \quad (1)$$

The numerical solutions to the nonlinear Klein–Gordon equation have received considerable attention in the literature. Different low-order finite difference schemes for approximating the nonlinear KGE were discussed in [13] and [24], while the stability of the methods derived in the former paper was later analyzed in [15]. Many methods have proposed spline collocation methods (see [14, 17] and references therein). Radial basis functions have also been proposed for similar problems in [9]. Using pseudo-spectral methods (of spectral accuracy in space and second-order accuracy in time) the solution

of Klein–Gordon–Schrödinger equations were approximated in [1]. The numerical solution of the damped nonlinear Klein–Gordon equations with Dirichlet boundary condition using variational method and finite element approximation was studied in [25]. In [6], a numerical method based on collocation nodes is developed to solve some nonlinear Klein–Gordon equations when  $h_1(u)$  is a polynomial. However, most of the previous methods are low-order in time or space and some of them present some limitations when they are extended to solve a large class of nonlinear Klein–Gordon equations. In this article, we will derive fourth- and sixth-order methods for one-dimensional nonlinear KGEs. The numerical methods are stable, either explicit or implicit, and easy to program.

The organization of the paper is as follows. In Section 2, we will derive higher-order (fourth- and sixth- order) explicit and implicit finite difference algorithms for the phi-four equation without forces  $f(x, t)$ . We will demonstrate numerical stability when the solution of the problem is smooth enough. In Section 3, we will show that these procedures can “easily” be generalized for the form-I of the nonlinear Klein–Gordon equation. Later, in Section 4, the reader will be able to check the efficiency of the algorithms proposed in this paper, compared with others that have appeared recently in the scientific literature. In addition, we will show that these schemes can be used with small modifications to solve the PDEs previously considered with forces in space and time.

## 2 The phi-four equation

Consider the following one-dimensional nonlinear Klein–Gordon equation:

$$u_{tt} - u_{xx} - u + u^3 = 0; \quad 0 \leq x \leq l, \quad 0 \leq t \leq t_e, \quad (2)$$

with initial conditions

$$u(x, 0) = f(x), \quad 0 \leq x \leq l, \quad (3)$$

$$u_t(x, 0) = \bar{f}(x), \quad 0 \leq x \leq l, \quad (4)$$

and Dirichlet boundary conditions

$$u(0, t) = g_1(t), \quad 0 \leq t \leq t_e, \quad (5)$$

$$u(l, t) = g_2(t), \quad 0 \leq t \leq t_e. \quad (6)$$

Let us consider a uniform mesh of the region  $\Omega = [0, l] \times [0, t_e]$ . We can take  $t_e = 1$  and  $l = 1$  without loss of generality. The vertices of the mesh will be  $(x_i, t_n)$ , where  $x_i = ih, i = 0, 1, \dots, M, t_n = nk, n = 0, 1, \dots, N, k = \Delta t$ , and  $h = \Delta x$ , and we will use the notation  $u_i^n = u(x_i, t_n)$ .

It is well known that the scheme  $u_i^{n+1} = u_{i-1}^n + u_{i+1}^n - u_i^{n-1}$  works exactly for the equation  $u_{tt} - u_{xx} = 0$  when  $h = k$ . Using this fact and cancelling the first terms of the local truncation error, a family of high order methods was derived in [16] for the non-homogeneous hyperbolic equation  $u_{tt} - u_{xx} = q(x, t)$ . We are using a similar idea to construct high-order finite difference methods for the phi-four equation first and later for the form-I of the nonlinear Klein–Gordon equation.

## 2.1 First iterations

The numerical schemes derived in this paper have three or more levels, hence it is necessary to use another procedure to obtain an approximation for the first step. Since we will derive sixth-order methods later in this section, here we will also build a sixth-order approximation for the first iteration to solve the phi-four equation (2) with initial conditions (3) and (4):

$$\begin{aligned}
 u_i^1 &= u_i^0 + h(u_t)_i^0 + \frac{h^2(u_{tt})_i^0}{2} + \frac{h^3(u_{ttt})_i^0}{6} + \frac{h^4(u_{tttt})_i^0}{24} + \frac{h^5(u_{ttttt})_i^0}{120} + O(h^6) \\
 &= f(ih) + h\bar{f}(ih) + \frac{h^2(f_{xx}(ih) + f(ih) - f(ih)^3)}{2} \\
 &\quad + \frac{h^3(\bar{f}_{xx}(ih) + \bar{f}(ih) - 3\bar{f}(ih)f(ih)^2)}{6} \\
 &\quad + \frac{h^4(-4f(ih)^3 + 3f(ih)^5 + 2f_{xx}(ih)(1 - 3f(ih)^2))}{24} \\
 &\quad + \frac{h^4(f_{xxx}(ih) + f(ih)(1 - 6f_x(ih)^2 - 6\bar{f}(ih)^2))}{24} \\
 &\quad + \frac{h^5((1 - 24f(ih)^2 + 27f(ih)^4 - 6f_x(ih)^2 - 24f(ih)f_{xx}(ih))\bar{f}(ih) - 6\bar{f}(ih)^3)}{120} \\
 &\quad + \frac{h^5(-12f(ih)f_x(ih)\bar{f}_x(ih) + 2\bar{f}_{xx}(ih)(1 - 3f(ih)^2) + \bar{f}_{xxx}(ih))}{120} \\
 &\quad + O(h^6). \tag{7}
 \end{aligned}$$

Whenever we use a method with four or more levels, we will use the method proposed above in the first iteration and later the fourth-order three-level scheme (9) (derived below) until the other algorithm can be used (for example, if we want to use the algorithm given by equation (10), we would need to use (7) and later (9) in the second iteration).

In these cases, the order of accuracy of the algorithm is given by the following well-known theorem.

**Theorem 1.** (See [22].) *If the initialization of a multistep scheme uses schemes of order of accuracy  $r'$  to compute the initial solution values  $v^j$  for  $j$  from 1 to  $J$  such that  $kh^{r'}$  is  $O(h^r)$ , and the initial data is in  $H^\rho$ , where  $[r, \rho]$  is the order of accuracy of the multistep scheme, then the order of accuracy of the solution is  $r$ .*

## 2.2 Fourth-order methods

The first method analyzed in [16] was the fourth-order scheme (for the nonhomogeneous hyperbolic equation  $u_{tt} - u_{xx} = q(x, t)$ )

$$u_i^{n+1} = u_{i-1}^n + u_{i+1}^n - u_i^{n-1} + k^2 \left( q_i^n + \frac{k^2((q_{xx})_i^n + (q_{tt})_i^n)}{12} \right). \tag{8}$$

If we approximate second derivatives of  $q$  (in what follows in this paper,  $q = q(x, t, u) = u - u^3$ ) in both space and time by three-term central methods, we obtain the formula

$$u_i^{n+1} = u_{i-1}^n + u_{i+1}^n - u_i^{n-1} + k^2 \left( \frac{2(u_i^n - (u_i^n)^3)}{3} + \frac{u_i^{n+1} - (u_i^{n+1})^3 + u_{i-1}^n - (u_{i-1}^n)^3 + u_{i+1}^n - (u_{i+1}^n)^3 + u_i^{n-1} - (u_i^{n-1})^3}{12} \right), \quad (9)$$

which is a fourth-order stable algorithm for the phi-four equation as it is demonstrated in the following theorem.

**Theorem 2.** *Formula (9) (when  $k = h$ ) applied to equation (2) is fourth-order and unconditionally von Neumann stable when  $u$ , the solution of (2), is smooth enough.*

*Proof.* Let us call  $u_{tt} = G = G(u, u_{xx}) = u_{xx} + u - u^3$ . Using Taylor series with (9), and after some manipulation of the formula, we obtain

$$\begin{aligned} (u_{tt} - G) &+ \frac{h^2(u_{tttt} - G_{tt} - G_{xx} + u_{ttxx})}{12} \\ &+ \frac{h^4(\phi_1(u, u_x, u_{xx}, u_{xxx}, u_{xxxx}, u_{xxxxx}) + \phi_2(u, u_t, u_{tt}, u_{ttt}, u_{tttt}, u_{ttttt}))}{720} \\ &+ O(h^6), \end{aligned}$$

where

$$\begin{aligned} \phi_1(u, u_x, u_{xx}, u_{xxx}, u_{xxxx}, u_{xxxxx}) \\ = 180u_x^2u_{xx} + 90uu_x^2 + 120uu_xu_{xxx} - 5u_{xxxx} + 15u^2u_{xxxx} - 2u_{xxxxxx} \end{aligned}$$

and

$$\begin{aligned} \phi_2(u, u_t, u_{tt}, u_{ttt}, u_{tttt}, u_{ttttt}) \\ = 180(u_t)^2u_{tt} + 90uu_{tt}^2 + 120uu_tu_{ttt} - 5u_{tttt} + 15u^2u_{tttt} + 2u_{ttttt}. \end{aligned}$$

Since the two first terms are zero for equation (2), the numerical scheme is fourth-order and the local truncation error of the method is

$$\frac{h^4(\phi_1(u, u_x, u_{xx}, u_{xxx}, u_{xxxx}, u_{xxxxx}) + \phi_2(u, u_t, u_{tt}, u_{ttt}, u_{tttt}, u_{ttttt}))}{720}.$$

The von Neumann stability analysis is not rigorously justified for nonlinear equations, but it is often justified approximately, assuming that the solution  $u(x, t)$  (and its numerical counterpart) does not vary too rapidly, which happens with most of these nonlinear Klein–Gordon equations. When analyzing similar nonlinear equations (see, for example, the analysis in [11] for Burger's equation), the equation and its numerical method are linearized and small terms are dropped, as a small deviation between two solutions of a nonlinear partial differential equation satisfies a linearization of that PDE on the background of an exact solution.

Hence, the amplification polynomial of (9) will be considered as follows:

$$g^2 - 2 \cos(\theta)g + 1 - h^2 \frac{g^2 + 2 \cos(\theta)g + 1 + 8g}{12} (1 - u_0^2).$$

The terms in  $g^2$  and independent on  $g$  are both  $1 - (h^2/12)(1 - u_0^2)$ . Then, by Cardano formulae, the product of both roots is 1. Therefore, taking  $\alpha = 1 - u_0^2$ , whenever the radical  $\Delta = -144 + 144 \cos(\theta)^2 + \alpha(24h^2 + 96h^2 \cos(\theta) + 24h^2 \cos(\theta)^2) + \alpha^2(15h^4 + 8h^4 \cos(\theta) + h^4 \cos(\theta)^2)$  is negative, both roots are complex conjugate, the roots satisfy  $|g_{1,2}| = 1$  (because  $|g_1||g_2| = 1$ ), and the numerical method is von Neumann stable.

Note that  $-144 + 144 \cos(\theta)^2 \leq 0$ , but still, for some values of  $\alpha$  and  $\theta$ , the radical can be positive. However, the radical can be considered as  $\Delta = -144 + 144 \cos(\theta)^2 + C_1(\theta, u_0)h^2 + C_2(\theta, u_0)h^4$ , hence whenever the radical is positive, then  $|g_1 - g_2| = 2\sqrt{\Delta} \leq 2C(u_0)h$  (we consider  $h \leq 1$  since we are proving stability) and  $|g_1||g_2| = 1$ . Thus, in any case, both roots satisfy  $|g_i| \leq 1 + C(u_0)h$  and the numerical method is von Neumann stable.

In [7], it is established that this condition is not sufficient for stability. In addition, it is required that the powers of the matrix

$$E_k(\theta, h) = \begin{pmatrix} \beta & -1 \\ 1 & 0 \end{pmatrix},$$

where  $\beta = -2(4\alpha h^2 + (12 + \alpha h^2) \cos(\theta))/(-12 + \alpha h^2)$ , are determined by its eigenvalues. A simple calculus give us

$$E_k(\theta, h)^s = \begin{pmatrix} \beta^s & (-1)^s \\ 1 & 0 \end{pmatrix}$$

with eigenvalues  $(\beta^s \pm \sqrt{(-1)^n 4 + \beta^{2s}})/2$ . □

**Remark 1.** It is not surprising that all the roots of the amplification polynomial of method (9) satisfy the condition  $|g_i| \leq 1 + C(u_0)k$ . In fact, as we stated before, all the numerical schemes in this paper come from modifying (8), which is well-known to be stable (with both roots of the amplification polynomial  $|g_i| = 1$ ), adding to the scheme terms that are  $O(k^2)$  uniformly in  $\theta$ . That is, since  $|g_i| \leq 1$  when  $k = 0$ , considering in addition that the solution of the PDE is smooth, if we use Taylor series, we can conclude that for small values of  $k$ , the roots of the amplification polynomials of the schemes in this paper satisfy  $|g_i| \leq 1 + C(u_0)k$  (see Corollary 2.2.2 and Theorem 2.2.3 in [22]), and only two eigenvalues are  $\approx 1$ .

This suggests that when the solution  $u$  of (2) is smooth enough, it is possible to use any of the following options to demonstrate stability: (i) a similar procedure as in the previous Theorem, or (ii) a similar procedure as in Theorems 2.2.1 and 2.2.3, and Corollary 2.2.2 in [22]. Hence, in what follows whenever we analyze a method, we will only study the local truncation error.

The first numerical method, presented in (9), was obtained approximating the second derivatives of  $q$  in both space and time by three-term central methods. Thus, (9) is a three-level algorithm, where it is necessary to solve a third-order equation at each point for the phi-four model. Although this is relatively easy to calculate, for other nonlinear Klein–Gordon equations, this could be a little more difficult, even though a simple Newton–Raphson scalar method can be used with good results.

In case, we want to use a fully explicit scheme, we would only need to approximate the term  $(q_{tt})_i^n$  in (8) by the three-term backward scheme  $(q_i^n - 2q_i^{n-1} + q_i^{n-2})/h^2$  (and  $(q_{xx})_i^n$  by the central scheme in space as for (9)) to obtain a third-order four-level method. If we want to obtain a fourth-order explicit algorithm, then it is necessary to use five levels, which we can achieve by replacing  $(q_{tt})_i^n$  in (8) by

$$\frac{2q_i^n - 5q_i^{n-1} + 4q_i^{n-2} - q_i^{n-3}}{h^2}$$

(and again replacing  $(q_{xx})_i^n$  by the central scheme in space).

**Theorem 3.** *The four-level explicit formula*

$$u_i^{n+1} = u_{i-1}^n + u_{i+1}^n - u_i^{n-1} + k^2 \left( \frac{11(u_i^n - (u_i^n)^3) + u_{i-1}^n - (u_{i-1}^n)^3}{12} + \frac{(u_{i+1}^n - (u_{i+1}^n)^3) - 2(u_i^{n-1} - (u_i^{n-1})^3) + u_i^{n-2} - (u_i^{n-2})^3}{12} \right) \quad (10)$$

(when  $k = h$ ), applied to equation (2), is only third order and its local truncation error is the following:

$$\frac{h^3(u_{ttt}(1 - 3u^2) + 6u_t(3u(-u + u^3 - u_{xx}) - u_t^2))}{12}.$$

**Theorem 4.** *The explicit formula*

$$u_i^{n+1} = u_{i-1}^n + u_{i+1}^n - u_i^{n-1} + k^2 \left( \frac{12(u_i^n - (u_i^n)^3) + u_{i-1}^n - (u_{i-1}^n)^3 + u_{i+1}^n - (u_{i+1}^n)^3}{12} + \frac{-5(u_i^{n-1} - (u_i^{n-1})^3) + 4(u_i^{n-2} - (u_i^{n-2})^3) - u_i^{n-3} + (u_i^{n-3})^3}{12} \right) \quad (11)$$

(when  $k = h$ ), applied to equation (2), is fourth order and its local truncation error is the following:

$$\begin{aligned} & \frac{h^4(2805u^5 - 1485u^7 + 2970uu_{xx} - 30u^2(88u_{xx} + 5u_{xxxx}))}{720} \\ & + \frac{h^4(55u^3(-25 + 18u_x^2 + 54u_t^2))}{720} \\ & - \frac{h^4(5u(-11 + 66u_x^2 + 180u_{xx}^2 - 24u_xu_{xxx} + 462u_t^2 + 264u_tu_{ttt}))}{720} \\ & + \frac{h^4(2(25u_{xxxx} - u_{xxxxx} + u_{xx}(55 + 90u_x^2 - 990u_t^2) + u_{ttttt}))}{720}. \end{aligned}$$

Another possibility is considering  $q(x, t, u) = u - u^3$ . In that case, a simple calculus provide us the formula

$$q_{tt} + q_{xx} = (2u_{xx} + u - u^3)(1 - 3u^2) - 6u(u_t^2 + u_x^2).$$

Here, we can consider central schemes in space (for  $u_x$  and  $u_{xx}$ ) and both central and backward schemes in time (for  $u_t$ ). In this way, we obtain three different algorithms similar to those derived above: one explicit with order three, a fourth-order three-level implicit scheme, and a fourth-order five-level explicit method.

### 2.3 Sixth-order methods

In a similar way as in the previous subsection, we can derive sixth-order methods from the sixth-order algorithm derived in [16] for the wave equation. However, in this case, it is necessary to approximate the term  $(q_{xx})_i^n + (q_{tt})_i^n$  with a fourth-order accurate formula, and the term  $(q_{xxxx})_i^n + (q_{ttxx})_i^n + (q_{tttt})_i^n$  with a second-order method.

Thus, we can obtain, for example, the following two sixth-order algorithms.

**Theorem 5.** *The local truncation error of the sixth-order six-level implicit method*

$$\begin{aligned}
 u_i^{n+1} = & u_{i-1}^n + u_{i+1}^n - u_i^{n-1} + k^2 \left( \frac{54q_i^{n+1} - 3q_{i+2}^n + 76q_{i+1}^n}{720} \right. \\
 & + \frac{481q_i^n + 76q_{i-1}^n - 3q_{i-2}^n - 10q_{i+1}^{n-1} + 32q_i^{n-1} - 10q_{i-1}^{n-1}}{720} \\
 & \left. + \frac{8q_{i+1}^{n-2} + 26q_i^{n-2} + 8q_{i-1}^{n-2} - 2q_{i+1}^{n-3} - 14q_i^{n-3} - 2q_{i-1}^{n-3} + 3q_i^{n-4}}{720} \right) \quad (12)
 \end{aligned}$$

(when  $k = h$ ), applied to equation (2), can be written briefly as

$$\frac{h^6(28q_{xxxx} - 14q_{ttxxx} + 154q_{ttttxx} - 224q_{ttttt} + 3u_{tttttt} - 3u_{xxxxxxx})}{60480}.$$

**Theorem 6.** *The local truncation error of the sixth-order explicit and seven-level scheme*

$$\begin{aligned}
 u_i^{n+1} = & u_{i-1}^n + u_{i+1}^n - u_i^{n-1} + k^2 \left( \frac{-3q_{i+2}^n + 76q_{i+1}^n + 805q_i^n + 76q_{i-1}^n}{720} \right. \\
 & + \frac{-3q_{i-2}^n - 10q_{i+1}^{n-1} - 778q_i^{n-1} - 10q_{i-1}^{n-1} + 8q_{i+1}^{n-2} + 1106q_i^{n-2}}{720} \\
 & \left. + \frac{8q_{i-1}^{n-2} - 2q_{i+1}^{n-3} - 824q_i^{n-3} - 2q_{i-1}^{n-3} + 327q_i^{n-4} - 54q_i^{n-5}}{720} \right) \quad (13)
 \end{aligned}$$

(when  $k = h$ ), applied to equation (2), can be written briefly as

$$\frac{h^6(28q_{xxxx} - 14q_{ttxxx} + 154q_{ttttxx} + 4312q_{ttttt} + 3u_{tttttt} - 3u_{xxxxxxx})}{60480}.$$

Obviously, methods given by equations (12) and (13) can be used only when  $i = 2, \dots, M - 2$ , so it is necessary to calculate other schemes for the cases  $i = 1$  and  $i = M - 1$ . For  $i = 1$ , it is necessary to approximate  $(q_{xx})_i^n$  and  $(q_{xxxx})_i^n$  with analogous formulae but evaluated in  $q_0^n, q_1^n, q_2^n, q_3^n, q_4^n$ , and  $q_5^n$ . We need to proceed similarly when  $i = M - 1$ , approximating  $(q_{xx})_i^n$  and  $(q_{xxxx})_i^n$  with the formulae evaluated in  $q_M^n, q_{M-1}^n, q_{M-2}^n, q_{M-3}^n, q_{M-4}^n$ , and  $q_{M-5}^n$ .

It is also possible to construct similar sixth-order methods using the fact that  $q(x, t, u) = u - u^3$ , calculating

$$q_i^n + \frac{k^2((q_{xx})_i^n + (q_{tt})_i^n)}{12} + \frac{k^4((q_{xxxx})_i^n + (q_{ttxx})_i^n + (q_{tttt})_i^n)}{360}$$

in terms of  $u$  and its derivatives in space and time, and finally replacing these derivatives with fourth-order numerical derivative formulae.

With both procedures it is also possible to obtain higher-order methods for the phi-four equation.

### 3 Nonlinear Klein–Gordon equation (form-I)

As we commented above, the phi-four equation is only a particular case of the form-I of the nonlinear Klein–Gordon equation

$$u_{tt} - u_{xx} - u + u^m = 0; \quad 0 \leq x \leq l, \quad 0 \leq t \leq t_e, \quad (14)$$

again with initial conditions (3) and (4), and boundary conditions (5) and (6) as above.

Hence, procedures as those employed in the previous Section can be used again, and actually the numerical schemes are very similar to those obtained in the previous theorems.

Thus, for the first iteration, now the approximation is

$$\begin{aligned} u_i^1 &= u_i^0 + h(u_t)_i^0 + \frac{h^2(u_{tt})_i^0}{2} + \frac{h^3(u_{ttt})_i^0}{6} + \frac{h^4(u_{tttt})_i^0}{24} + \frac{h^5(u_{ttttt})_i^0}{120} + O(h^6) \\ &= f(ih) + h\bar{f}(ih) + \frac{h^2(f_{xx}(ih) + f(ih) - f(ih)^m)}{2} \\ &\quad + \frac{h^3(\bar{f}_{xx}(ih) + \bar{f}(ih) - m\bar{f}(ih)f(ih)^{m-1})}{6} \\ &\quad + \frac{h^4(f(ih) - (1+m)f(ih)^m + mf(ih)^{2m-1} - 2mf(ih)^{m-1}f_{xx}(ih))}{24} \\ &\quad + \frac{h^4(2f_{xx}(ih) + f_{xxxx}(ih) - (m-1)mf(ih)^{m-2}(f_x(ih)^2 + \bar{f}(ih)^2))}{24} \\ &\quad + \frac{h^5(4m-3)mf(ih)^{2m-2}\bar{f}(ih)}{120} \\ &\quad - \frac{h^5m(2-3m+m^2)f(ih)^{m-3}\bar{f}(ih)(f_x(ih)^2 + \bar{f}(ih)^2)}{120} \end{aligned}$$

$$\begin{aligned}
 & + \frac{h^5(-2(m-1)mf(ih)^{m-2}(2f_{xx}(ih)\bar{f}(ih) + f_x(ih)\bar{f}_x(ih)))}{120} \\
 & - \frac{h^5(mf(ih)^{m-1}((3m-1)\bar{f}_x(ih) + 2\bar{f}_{xx}(ih)) + \bar{f}(ih))}{120} \\
 & + \frac{h^5(2\bar{f}_{xx}(ih) + \bar{f}_{xxxx}(ih))}{120} + O(h^6). \tag{15}
 \end{aligned}$$

Fourth-order numerical schemes, equivalent to those given by equations (9) and (11), and whose properties were studied in Theorems 2 and 4, are given with the following theorem.

**Theorem 7.** *The implicit three-level formula*

$$\begin{aligned}
 u_i^{n+1} = & u_{i-1}^n + u_{i+1}^n - u_i^{n-1} + k^2 \left( \frac{2(u_i^n - (u_i^n)^m)}{3} + \frac{u_i^{n+1} - (u_i^{n+1})^m}{12} \right. \\
 & \left. + \frac{u_{i-1}^n - (u_{i-1}^n)^m + u_{i+1}^n - (u_{i+1}^n)^m + u_i^{n-1} - (u_i^{n-1})^m}{12} \right) \tag{16}
 \end{aligned}$$

and the five-level explicit method

$$\begin{aligned}
 u_i^{n+1} = & u_{i-1}^n + u_{i+1}^n - u_i^{n-1} \\
 & + k^2 \left( \frac{12(u_i^n - (u_i^n)^m) + u_{i-1}^n - (u_{i-1}^n)^m + u_{i+1}^n - (u_{i+1}^n)^m}{12} \right. \\
 & \left. + \frac{-5(u_i^{n-1} - (u_i^{n-1})^m) + 4(u_i^{n-2} - (u_i^{n-2})^m) - u_i^{n-3} + (u_i^{n-3})^m}{12} \right) \tag{17}
 \end{aligned}$$

are both fourth order, and their local truncation error are respectively

$$\begin{aligned}
 & \frac{h^4(5m(m^3 - 6m^2 + 11m - 6)u^{m-4}(u_x^4 + u_t^4))}{720} \\
 & + \frac{h^4(30m(m^2 - 3m + 2)u^{m-3}(u_x^2u_{xx} + u_t^2u_{tt}))}{720} \\
 & + \frac{h^4(5(m-1)mu^{m-2}(3u_{xx}^2 + 4u_xu_{xxx} + 3u_{tt}^2 + 4u_tu_{ttt}))}{720} \\
 & + \frac{h^4(5mu^{m-1}(u_{xxxx} + u_{tttt}) - 5u_{xxxx} - 2u_{xxxxx} - 5u_{tttt} + 2u_{ttttt})}{720}
 \end{aligned}$$

and

$$\begin{aligned}
 & \frac{h^4(5m(m^3 - 6m^2 + 11m - 6)u^{m-4}(u_x^4 - 11u_t^4))}{720} \\
 & + \frac{h^4(30m(m^2 - 3m + 2)u^{m-3}(u_x^2u_{xx} - 11u_t^2u_{tt}))}{720} \\
 & + \frac{h^4(5(m-1)mu^{m-2}(3u_{xx}^2 + 4u_xu_{xxx} - 33u_{tt}^2 - 44u_tu_{ttt}))}{720} \\
 & + \frac{h^4(5mu^{m-1}(u_{xxxx} - 11u_{tttt}) - 5u_{xxxx} - 2u_{xxxxx} + 55u_{tttt} + 2u_{ttttt})}{720}.
 \end{aligned}$$

Numerical stability of the algorithm given by (16) can be proven as in (2), just changing  $u - u^3$  by  $u - u^m$  and writing  $\alpha = 1 - u_0^{m-1}$  instead of  $\alpha = 1 - u_0^2$ . Again, it is necessary that the solution of problem (14) is smooth enough. For the stability of the numerical scheme (17) it is better to proceed as it was suggested in Remark 1.

Analogously, it is also relatively easy to construct sixth-order and eight-order methods, and other higher-order schemes with a similar procedure as above. Actually, Theorems 5 and 6 can be used for the problem given by formula (14), it is only necessary to write  $q = u - u^m$  instead of  $q = u - u^3$ .

## 4 Numerical examples

In this section, we offer some numerical examples developed with the methods shown in the preceding sections.

### 4.1 First numerical example

For the first numerical example, we shall consider problem (1) with the traditional initial and boundary conditions (3)–(6) such that the solution is

$$u(x, t) = \frac{\sqrt{2}}{\cosh((x - 2t)/\sqrt{3})}.$$

As we commented above for our fourth- and sixth-order methods, we need to approximate the solution at the first step, and for that we use equation (7) to calculate it. In this numerical experiment, we will employ the implicit fourth-order method (9) and the explicit sixth-order method (13). For the former one, we require additional approximations of the solution at the first levels. Hence, there we can employ (9) at the first steps.

The numerical errors when the method (9) is used (for different values of  $h = 1/8, 1/16, 1/32, 1/64$  and  $1/128$ ) are given at  $t = 1, x = 0.25, 0.50$  and  $0.75$  in Table 1.

As we can check in Table 2, this methods converge at the expected rate. There, we show the values  $r_i = \log_{c_i} err_i / err_1$ , where  $err_i$  is the error obtained with  $h_i$  and  $c_i = h_i/h_1$ .

As we also would like to study the numerical convergence of the sixth-order method, and for this one, we would need to employ several times another algorithm at the first steps, we will proceed in two different ways to better study the convergence rate of the

Table 1. Errors for the fourth-order method (9), 1st example.

$h$	$x = 0.25$	$x = 0.5$	$x = 0.75$
1/8	$8.489 \cdot 10^{-7}$	$2.481 \cdot 10^{-6}$	$2.018 \cdot 10^{-6}$
1/16	$3.383 \cdot 10^{-8}$	$1.097 \cdot 10^{-7}$	$7.968 \cdot 10^{-8}$
1/32	$1.711 \cdot 10^{-9}$	$5.703 \cdot 10^{-9}$	$3.661 \cdot 10^{-9}$
1/64	$9.730 \cdot 10^{-11}$	$3.244 \cdot 10^{-10}$	$1.897 \cdot 10^{-10}$
1/128	$5.657 \cdot 10^{-12}$	$1.913 \cdot 10^{-12}$	$1.052 \cdot 10^{-11}$

Table 2. Convergence, fourth-order method (9), 1st example.

$h$	$x = 0.25$	$x = 0.5$	$x = 0.75$
1/16	4.649	4.499	4.663
1/32	4.477	4.383	4.553
1/64	4.364	4.300	4.459
1/128	4.299	4.246	4.387

Table 3. Errors for the sixth-order method (13), 1st example, real solution.

$h$	$x = 0.25$	$x = 0.5$	$x = 0.75$
1/8	$1.300 \cdot 10^{-5}$	$1.945 \cdot 10^{-5}$	$4.633 \cdot 10^{-5}$
1/16	$1.240 \cdot 10^{-7}$	$5.658 \cdot 10^{-7}$	$6.609 \cdot 10^{-7}$
1/32	$2.970 \cdot 10^{-10}$	$4.255 \cdot 10^{-9}$	$4.002 \cdot 10^{-9}$
1/64	$1.527 \cdot 10^{-11}$	$1.022 \cdot 10^{-11}$	$2.136 \cdot 10^{-11}$
1/128	$2.898 \cdot 10^{-13}$	$3.524 \cdot 10^{-13}$	$1.121 \cdot 10^{-12}$

Table 4. Convergence, sixth-order method (13), 1st example, real solution.

$h$	$x = 0.25$	$x = 0.5$	$x = 0.75$
1/16	6.711	5.103	6.131
1/32	7.708	6.079	6.749
1/64	6.566	6.953	7.016
1/128	6.355	6.430	6.325

Table 5. Errors for the sixth-order method (13) combined with the fourth-order method (9), 1st example.

$h$	$x = 0.25$	$x = 0.5$	$x = 0.75$
1/8	$7.745 \cdot 10^{-6}$	$2.600 \cdot 10^{-5}$	$5.040 \cdot 10^{-5}$
1/16	$1.287 \cdot 10^{-7}$	$5.904 \cdot 10^{-7}$	$7.366 \cdot 10^{-7}$
1/32	$3.199 \cdot 10^{-9}$	$9.605 \cdot 10^{-10}$	$2.705 \cdot 10^{-9}$
1/64	$8.960 \cdot 10^{-11}$	$1.110 \cdot 10^{-10}$	$1.188 \cdot 10^{-10}$
1/128	$1.861 \cdot 10^{-12}$	$3.878 \cdot 10^{-12}$	$4.751 \cdot 10^{-12}$

Table 6. Convergence, sixth-order method (13) combined with the fourth-order method (9), 1st example.

$h$	$x = 0.25$	$x = 0.5$	$x = 0.75$
1/16	5.911	5.461	6.097
1/32	5.621	7.362	7.903
1/64	5.467	5.946	6.231
1/128	5.497	5.669	5.835

scheme: (i) we will use the real solution for the first steps and (ii) we will employ the implicit fourth-order method (9) in these first steps.

First, in Tables 3 and 4 the numerical errors and the study for the convergence rate are shown with the real solution in the first steps. Later, in Tables 5 and 6, we proceed in a similar way with procedure (ii). The convergence rate is as expected (see Theorem 1).

As we can check in Tables 4 and 6 this method converges at the expected rate. There, we showed the values  $r_i = \log_{c_i} \text{err}_i / \text{err}_1$ , where  $\text{err}_i$  is the error obtained with  $h_i$  and  $c_i = h_i/h_1$ .

## 4.2 Second numerical example

For the second numerical example we shall consider a problem with unknown solution, where the equation and initial conditions are as follows:

$$u_{tt} - u_{xx} - u + u^2 = f(x, t); \quad 0 \leq x \leq l, \quad 0 \leq t \leq t_e;$$

$$u(x, 0) = f(x) = -1 + x + \cos(\pi x),$$

$$u_t(x, 0) = \bar{f}(x) = 1 - x - \cos(\pi x),$$

and the following boundary conditions:

$$u(0, t) = g_1(t) = 0,$$

$$u(l, t) = g_2(t) = -e^{-t}.$$

As in the previous example for our fourth- and sixth-order methods, we need to approximate the solution at the first step, and for that we use equation (15) to calculate it. In this numerical experiment we will employ the implicit fourth-order method (16) for iterations 2 and 3, and the explicit fourth-order scheme (17) in the followings steps.

We shall also consider a sixth-order method, in this case, the one given by formula (13). As iterations 2 to 5 are unknown, we have used (16) before we can employ the sixth-order explicit formula (13).

The numerical errors when the method (17) is employed for different values of  $h = 1/8, 1/16, 1/32, 1/64, \text{ and } 1/128$  are given at  $t = 1, x = 0.25, 0.50, \text{ and } 0.75$  in Table 7.

As in the previous problem, we can check in Table 8 that this method converges at the expected rate. Again, we showed the values  $r_i = \log_{c_i} \text{err}_i / \text{err}_1$ , where  $\text{err}_i$  is the error obtained with  $h_i$  and  $c_i = h_i/h_1$ .

We also would like to study the numerical convergence of the sixth-order method and for this one, we need to employ several times a fourth-order method at the first steps as we comment before. In Tables 9 and 10, the numerical errors and the study for the convergence rate are shown.

Table 7. Errors for the fourth-order method (17), 2nd example.

$h$	$x = 0.25$	$x = 0.5$	$x = 0.75$
1/8	$3.363 \cdot 10^{-4}$	$2.509 \cdot 10^{-4}$	$2.470 \cdot 10^{-4}$
1/16	$2.964 \cdot 10^{-5}$	$1.381 \cdot 10^{-5}$	$1.510 \cdot 10^{-5}$
1/32	$1.609 \cdot 10^{-6}$	$7.390 \cdot 10^{-7}$	$7.205 \cdot 10^{-7}$
1/64	$9.226 \cdot 10^{-8}$	$4.147 \cdot 10^{-8}$	$3.826 \cdot 10^{-8}$
1/128	$5.564 \cdot 10^{-9}$	$2.420 \cdot 10^{-9}$	$2.103 \cdot 10^{-9}$

Table 8. Convergence, fourth-order method (17), 2nd example.

$h$	$x = 0.25$	$x = 0.5$	$x = 0.75$
1/16	3.504	4.184	4.032
1/32	3.854	4.204	4.211
1/64	3.944	4.188	4.219
1/128	3.971	4.165	4.210

Table 9. Errors for the sixth-order method (13) combined with the fourth-order method (16), 2nd example.

$h$	$x = 0.25$	$x = 0.5$	$x = 0.75$
1/8	$1.434 \cdot 10^{-3}$	$5.438 \cdot 10^{-4}$	$9.161 \cdot 10^{-4}$
1/16	$8.779 \cdot 10^{-6}$	$2.232 \cdot 10^{-5}$	$3.955 \cdot 10^{-5}$
1/32	$2.307 \cdot 10^{-7}$	$8.038 \cdot 10^{-7}$	$2.404 \cdot 10^{-6}$
1/64	$3.565 \cdot 10^{-8}$	$3.453 \cdot 10^{-8}$	$1.526 \cdot 10^{-7}$
1/128	$3.617 \cdot 10^{-9}$	$1.791 \cdot 10^{-9}$	$9.456 \cdot 10^{-9}$

Table 10. Convergence, sixth-order method (13) combined with the fourth-order method (16), 2nd example.

$h$	$x = 0.25$	$x = 0.5$	$x = 0.75$
1/16	7.351	4.607	4.534
1/32	6.301	4.701	4.287
1/64	5.099	4.648	4.184
1/128	4.650	4.553	4.141

### 4.3 Third numerical example

In this test example, we compare our schemes with the fourth example used by Rashidinia et al. in [17]. That paper is devoted to the numerical solution of a one-dimensional nonlinear Klein–Gordon equation, which is given in the following form:

$$u_{tt} - u_{xx} + u + u^3 = s(t, x), \quad -1 \leq x \leq 1, \quad 0 \leq t,$$

where

$$s(t, x) = (-2 + x^2) \cosh(t + x) + x^6 \cosh^3(t + x) - 4x \sinh(t + x),$$

and subjected to the initial conditions

$$\begin{aligned} u(x, 0) &= f(x) = x^2 \cosh(x), & -1 \leq x \leq 1, \\ u_t(x, 0) &= \bar{f}(x) = x^2 \sinh(x), & -1 \leq x \leq 1, \end{aligned}$$

with the exact solution  $u(x, t) = x^2 \cosh(x + t)$ . They obtained the boundary conditions from the exact solution, and they solved this problem with the algorithm derived in [17] with  $h = 0.01$  and  $k = 0.0001$ .

As the reader can check, this equation has not been exactly considered previously in the paper (since the phi-four equation is  $u_{tt} - u_{xx} - u + u^3 = 0$ , a sign is different and  $s(t, x) \neq 0$  in this case). Also the interval, where the equations are considered, is different, ranging from  $-1$  to  $1$  instead of  $0$  to  $1$ . However, similar algorithms can be derived and again stability for (smooth enough) functions can be proved as in previous methods from Remark 1 and Duhamel’s principle (a scheme is stable for the equation  $P_{k,h}v = f$  if it is stable for the equation  $P_{k,h}v = 0$  as it is shown in [22, Chap. 9]).

In this case, we have used a scheme similar to the one given by equation (7) for the first iteration, and later, for the following 4 iterations, we have used an implicit method similar to the one derived in equation (8). Finally, we have applied a six-order explicit algorithm similar to the one given by equation (13).

Table 11. Comparison between our method and others given in [17], 3rd example.

$t = 1$	Dehghan et al.	Rashidinia et al.	Our method
$l_{\text{inf}}$	$5.0705 \cdot 10^{-5}$	$3.5666 \cdot 10^{-6}$	$1.02839 \cdot 10^{-9}$
$l_2$	$2.9474 \cdot 10^{-4}$	$2.5993 \cdot 10^{-5}$	$3.7079 \cdot 10^{-9}$
RMS	$2.0789 \cdot 10^{-5}$	$2.5930 \cdot 10^{-6}$	$2.61535 \cdot 10^{-10}$

We have used the same problem than they have, but in this our with  $h = k = 0.01$ . Hence, our  $k$  is much bigger than theirs, which means a large reduction in computing time. However, the methods studied here are higher-order. We did not want to compare our schemes with theirs in other numerical problems since it looks “unfair”, but we considered that we had to do it in one instance to make clearer the differences to the reader. Table 11 offers the comparison results. We calculated the errors at  $t_e = 1$  with the infinity and  $L_2$  norm and also the root mean square (RMS) which can be calculated in this case as the error with the  $L_2$  norm divided by the root square of the number of nodes (201). We took the values with the other methods from Table 4 in [17].

## 5 Conclusions

In this paper, the problem of solving some nonlinear Klein–Gordon equations (KGE) has been considered. More exactly, we have derived different fourth- and sixth-order explicit and implicit algorithms to solve the phi-four equation and the form-I of the nonlinear Klein–Gordon equation. We have also demonstrated that similar schemes can be obtained for similar equations. The stability and consistency of the proposed schemes were studied under certain conditions of smoothness of the solution of these equations.

We have provided numerical results to explain the efficiency of the new algorithms, their stability, and the convergence rate. We have compared the results obtained with the new methods against others obtained with some schemes already existing in the scientific literature. In comparison, they show an improvement in both, the computational effort and the error rate, as they are smaller than in the other schemes.

Regarding the future work, we think it would be interesting to derive and study the numerical implementation of numerical schemes for other nonlinear Klein–Gordon when  $G(u)$  in  $u_{tt} - u_{xx} - G(u, u_x, u_t, \dots) = 0$  is not a polynomial or suffers perturbations.

**Acknowledgment.** The authors would like to thank J.C. Strikwerda for useful discussions and comments.

## References

1. W. Bao, L. Yang, Efficient and accurate numerical methods for the Klein-Gordon-Schrödinger equations, *J. Comput. Phys.*, **225**(2):1863–1893, 2007.
2. M. Bawin, M. Jaminon, Pion form factor and the Klein–Gordon equation, *Phys. Rev. C*, **30**:331–334, 1984.

3. A. Biswas, A.H. Kara, A.H. Bokhari, F.D. Zaman, Solitons and conservation laws of Klein–Gordon equation with power law and log law nonlinearities, *Nonlinear Dyn.*, **73**:2191–2196, 2013.
4. A. Biswas, A.H. Kara, A.H. Bokhari, F.D. Zaman, Solitary waves and conservation laws of complex-valued Klein–Gordon equation in  $\phi$ -4 field theory, *Indian J. Phys.*, **88**:311–315, 2014.
5. A. Biswas, C. Zony, E. Zerrad, Soliton perturbation theory for the quadratic nonlinear Klein–Gordon equation, *Appl. Math. Comput.*, **203**:153–156, 2008.
6. B. Bülbül, M. Sezer, A new approach to numerical solution of nonlinear Klein–Gordon equation, *Math. Probl. Eng.*, **2013**, Article ID 869749, 7 pp., 2013. .
7. P.G. Ciarlet, J.L. Lions (Eds.), *Handbook of Numerical Analysis. Vol. I: Finite Difference Methods (Part 1), Solution of Equations in  $R^n$  (Part 1)*, Elsevier, Amsterdam, 1990.
8. Z. Dai, D. Xian, Homoclinic breather-wave solutions for sine-Gordon equation, *Commun. Nonlinear Sci. Numer. Simul.*, **14**(8):3292–3295, 2009.
9. M. Dehghan, A. Shokri, Numerical solution of the nonlinear Klein–Gordon equation using radial basis functions, *J. Comput. Appl. Math.*, **230**(2):400–410, 2009.
10. X. Deng, M. Zhao, X. Li, Travelling wave solutions for a nonlinear variant of the phi-four equation, *Math. Comput. Modelling* **49**(3–4):617–622, 2009.
11. R.H. Enns, G.C. McGuire, *Nonlinear Physics with Mathematica for Scientists and Engineers*, Birkhäuser, Boston, 2001.
12. N. Ercolani, M.G. Forest, D.W. McLaughlin, Geometry of the modulational instability. III: Homoclinic orbits for the periodic sine-Gordon equation, *Physica D*, **43**(2–3):349–384, 1990.
13. S. Jiménez, L. Vázquez, Analysis of four numerical schemes for a nonlinear Klein–Gordon equation, *Appl. Math. Comput.*, **35**(1):61–94, 1990.
14. S.A. Khuri, A. Sayfy, A spline collocation approach for the numerical solution of a generalized nonlinear Klein–Gordon equation, *Appl. Math. Comput.*, **216**(4):1047–1056, 2010.
15. M.A.M. Lynch, Large amplitude instability in finite difference approximations to the Klein–Gordon equation, *Appl. Numer. Math.*, **31**(2):173–182, 1999.
16. J. Martín-Vaquero, B.A. Wade, On efficient numerical methods for an initial-boundary value problem with nonlocal boundary conditions, *Appl. Math. Modelling*, **36**(8):3411–3418, 2012.
17. J. Rashidinia, M. Ghasemi, R. Jalilian, Numerical solution of the nonlinear Klein–Gordon equation, *J. Comput. Appl. Math.*, **233**(8):1866–1878, 2010.
18. R. Sassaman, A. Biswas, Soliton perturbation theory for phi-four model and nonlinear Klein–Gordon equations, *Commun. Nonlinear Sci. Numer. Simul.*, **14**(8):3239–3249, 2009.
19. R. Sassaman, A. Biswas, Topological and non-topological solitons of the generalized Klein–Gordon equations, *Appl. Math. Comput.*, **215**(1):212–220, 2009.
20. R. Sassaman, A. Heidari, A. Biswas, Topological and non-topological solitons of nonlinear Klein–Gordon equations by He’s semi-inverse variational principle, *J. Franklin Inst.*, **347**(7):1148–1157, 2010.

21. A.A. Soliman, Exact traveling wave solution of nonlinear variants of the RLW and the phi-four equations, *Physics Lett. A*, **368**(5):383–390, 2007.
22. J.C. Strikwerda, *Finite Difference Schemes and Partial Differential Equations*, 2nd ed., SIAM, Philadelphia, 2004.
23. N.K. Vitanov, Modified method of simplest equation: Powerful tool for obtaining exact and approximate traveling-wave solutions of nonlinear PDEs, *Commun. Nonlinear Sci. Numer. Simul.*, **16**(3):1176–1185, 2011.
24. L. Vu-Quoc, L. Shaofan, Invariant-conserving finite difference algorithms for the nonlinear Klein–Gordon equation, *Comput. Methods Appl. Mech. Eng.*, **107**(3):341–391, 1993.
25. Q. Wang, D. Cheng, Numerical solution of damped nonlinear Klein–Gordon equations using variational method and finite element approach, *Appl. Math. Comput.*, **162**(1):381–401, 2005.
26. A. Wazwaz, Generalized forms of the phi-four equation with compactons, solitons and periodic solutions, *Math. Comput. Simul.*, **69**(5–6):580–588, 2005.
27. H. Zhou, J. Shen, Bifurcations of travelling wave solutions for modified nonlinear dispersive phi-four equation, *Appl. Math. Comput.*, **217**(4):1584–1597, 2010.