Computer Methods in Applied Mechanics and Engineering 107 (1993) 341–391
North-Holland

CMA 403

Invariant-conserving finite difference algorithms for the nonlinear Klein–Gordon equation

Loc Vu-Quoc and Shaofan Li

Aerospace Engineering, Mechanics and Engineering Science, University of Florida, Gainesville, FL 32611, USA

Received 13 May 1992

A formalism for systematically deriving second order accurate finite difference algorithms which conserve certain invariant quantities in the original nonlinear PDEs is presented. Three algorithms are derived for the nonlinear Klein–Gordon equation (NLKGE) based on the proposed formalism. The local conservation laws of the NLKGE form the basic starting point in our derivation, which hinges essentially on the commutativity of certain finite difference operators. Such commutativity in the discrete approximations allows a preservation of the derivation properties of the continuous counterparts at the PDE level. With appropriate boundary conditions, the proposed algorithms preserve in the discrete sense either the total system energy or the system's linear momentum. Several variants of the present algorithms and their relation to previously proposed algorithms are discussed. An analysis of the accuracy and stability is conducted to compare the different variants of the proposed algorithms. The preservation of energy of the present algorithms for the NLKGE can also be viewed as providing a method of stabilization for conditionally stable algorithms for the linear wave equation. The computer implementation of the proposed algorithms, with the treatment of the boundary conditions, is presented in detail. Numerical examples are given concerning soliton collisions in the sine-Gordon equation, the double sine-Gordon equation, and the $\phi^4_\pm$ ('phi-four') equation. The numerical results demonstrate that the present algorithms can preserve accurately (up to 10 decimal digits) the total system energy for a very coarse grid. Reliable algorithms for Josephson junction models, which contain dissipation, damping mechanisms and driving bias current, are obtained as direct by-products of the proposed invariant-conserving algorithms for the NLKGE. Even though presented mainly for the 1-D case, the proposed algorithms are generalizable to the 2-D and 3-D cases, and to the case of complex-valued NLKGE.

1. Introduction

For the past two decades, there has been extensive interest in a class of nonlinear evolution equations that admits extremely stable solutions termed solitons. An example of such equations is the sine-Gordon equation

$$U_{tt} - \Delta U + \sin(U) = 0\,, \tag{1.1}$$

where the function U has (x^1, x^2, x^3, t) as arguments, $U_{tt} := \partial^2 U/\partial t^2$ and $\Delta U := \partial^2 U/\partial x^i \partial x^i$. In

Correspondence to: Professor Loc Vu-Quoc, Aerospace Engineering, Mechanics and Engineering Science, University of Florida, Gainesville, FL 32611, USA.

fact, the more general nonlinear Klein–Gordon equation (NLKGE),

$$U_{tt} - \Delta U + G'(U) = 0\,, \tag{1.2}$$

where $G(\cdot)$ is a nonlinear differentiable function and $G'(U) := \mathrm{d}G(U)/\mathrm{d}U$, is the most natural generalization of the linear wave equation [1]. Liouville [2] was the first to study this type of equation in his work on surfaces with constant curvature, in which $G'(U) = \exp(U)$, and provided an exact solution. The NLKGE includes the governing equations of many important physical phenomena. Using the cubic function $G'(U) = \pm(U - U^3)$, Dashen et al. [3] constructed a model field theory; this equation is customarily referred to as the $\phi^4_{\pm}$ ('phi-four') equation (see [4]).[1] As mentioned above, an NLKGE with $G'(U) = \sin(U)$, called the sine-Gordon equation, has received a lot of attention since its exact solution can be obtained.[2] The sine-Gordon equation can be found in the motion of a rigid pendulum attached to an extendible string [5], in rapidly rotating fluids [6], in the physics of Josephson junctions and other applications [7, 8].

There is a very small class of nonlinear partial differential equations (PDE) that can be solved for exact solutions by analytical methods. One such method is the inverse scattering method (ISM),[3] which is often regarded as an important development of analytical methods for nonlinear PDEs in the last 20 years. However, ISM can solve the initial value problems for a very small class of nonlinear PDEs (e.g. [9]).[4] For this reason, it is sometimes said that the set of solvable nonlinear PDEs has a 'measure zero', and that one could conceivably consider linear PDEs and solvable nonlinear PDEs as belonging to a class in which solutions can be added in some function spaces [1]. Apart from the sine-Gordon equation which is solvable by the ISM, most other NLKGEs must be treated by numerical methods.

Conservation laws play an important role in soliton theory (e.g. [8, 10]) as they form the foundation of the structural stability of solitons. After colliding into each other, the two solitons emerge unchanged, i.e., they can pass through one another without changing their initial shape and with only a phase shift.[5] The structural stability of solitons, first observed numerically by Zabusky and Kruskal [11] when they solved for the case of two colliding solitons of the Korteweg–de Vries (KdV) equation, is associated with the fact that there is an infinity of conservation laws in the KdV equation,[6] a classic case in which numerical simulations gave insights and inspired subsequent theoretical development (see also [12]). It is believed, without formal justification, that having an infinite number of conservation laws is

[1] The name 'phi-four' comes from the use of ϕ as variable instead of U, and because of the power 4 of ϕ in the potential function $G(\phi) = \pm\frac{1}{2}(\phi^2 - \frac{1}{2}\phi^4)$.

[2] The linear approximation to the sine-Gordon equation, i.e., $U_{tt} - U_{xx} + U = 0$ is called the (normalized) linear Klein–Gordon equation in quantum theory.

[3] Also called the inverse spectral transform, or simply spectral transform, which is for nonlinear PDEs a direct generalization of the Fourier transform for linear PDEs.

[4] With the requirement that U and various of its derivatives tend to zero as $\|x\| \to \infty$. We assume this type of boundary conditions in the present paper.

[5] The phase shift is the difference in the position of a soliton after it emerges from a collision as compared with the position of a soliton without collision at the same instant of time.

[6] These solitons belong to a shape-preserving class, and are sometimes called 'aristocratic solitons' to distinguish them from 'solitons' which are not shape preserving.

one of the two properties necessary for a nonlinear PDE to be solvable by the ISM [1, p. 304].[7] Indeed, the KdV equation was the first to be solved analytically by the ISM [13]. The sine-Gordon equation also possesses an infinite number of conservation laws [14, 15][8] and can be solved exactly by the ISM [16, 17].

However, not all of the NLKGEs possess an infinite number of conservation laws, and are solvable analytically. An example is the double sine-Gordon equation with $G'(u) = \pm(\sin u + \frac{1}{2}\lambda \sin \frac{1}{2}u)$; this equation possesses only three conservation laws, and is not solvable by any currently known analytical methods for nonlinear PDEs [18]. Collisions of solitary waves in these cases are termed inelastic because of an indelible mark that results from the collision and is carried by the solitary waves after the latter have interacted with each other. The system energy, on the other hand, is always conserved for the general NLKGE. More frequently than not, numerical methods or singular perturbation methods must be employed when dealing with analytically unsolvable PDEs. A survey of existing numerical methods for the solution of the NLKGE is given in [4, 19]. Even though both finite element methods (FEMs) and finite difference methods (FDMs) are surveyed in [19], it seems that FDM is a popular choice to tackle the NLKGE. Concerning the application of FEM to nonlinear wave equations, we mention the work of Argyris and Haase [21], Gardner et al. [22], Carey and Shen [23]. In the present paper, we address an important aspect of finite difference (FD) algorithms for the NLKGE, namely the conservation of the system invariants (e.g., energy) in the numerical solution.

Many numerical algorithms do not preserve the system energy, even though the error decreases as the mesh size tends to zero. For example, Ablowitz et al. [24] devise one of the most useful numerical algorithms for the NLKGE [19, p. 588]; calculations show that the error in energy can fluctuate between 2% to 6% for a mesh size of $h = 0.05$. FD algorithms in general do not necessarily share the same conservation properties as found in the continuous PDE counterpart; this observation also holds true for the system energy.[9] Our goal here is to design algorithms that not only accurately approximate the solution and the system energy, but also duplicate the *energy conservation property* of the NLKGE in the discrete sense, and this with arbitrary mesh size. We will explain the properties of the algorithm by Ablowitz et al. [24] as a variant of an energy-conserving algorithm. An example of the preservation of certain properties of the PDE by FD algorithms at the discrete level can be found in the FD five-point Laplacian, which preserves the maximum principle of elliptic PDEs at the discrete level (e.g., [25]).

Attempts to design energy-conserving algorithms have been made in several areas of mechanics. Some examples are Hughes et al. [26] for nonlinear elastodynamics, Greenspan [27] for general second order ordinary differential equations, Chin and Qin [28] for solving the three-body problem, Simo and Honein [29] for discrete conservation laws in elastoviscoplasticity. Specifically concerning the NLKGE, Strauss and Vazquez [30] propose an algorithm that conserves a discrete energy. We show that the Strauss and Vazquez algorithm can be

[7] The connection between an infinity of conservation laws and the existence of solitons for the KdV equation was pointed out by Lax [20].

[8] These authors work with the sine-Gordon equation under the form $u_{\xi\tau} = \sin u$ obtained from (1.2), known as the Lorentz covariant form, via a transformation to the 'light cone' coordinates $\xi = \frac{1}{2}(x + t)$ and $\tau = \frac{1}{2}(x - t)$.

[9] The conservation of the local linear momentum is in general satisfied in numerical algorithms as it is imposed at all time steps.

thought of as a variant of the Algorithm II in the present paper, which is an explicit, conditionally stable, method for the (un-normalized) linear Klein–Gordon equation (ULKGE).[10] On the other hand, the proposed Algorithm I is implicit, unconditionally stable for the ULKGE, and preserves the system energy for the NLKGE. The connection between the discrete energy and the local conservation law is clearly explained. Both Algorithms I and II are second order accurate. Algorithm III, which is also second order accurate, is designed to conserve the system's linear momentum.

We begin by describing these algorithms for the real-valued NLKGE with one space dimension. Generalization of the proposed algorithms to 2-D and 3-D cases is direct; however, noteworthy is the application of the formalism presented here to derive reliable, robust algorithms for models that do not conserve energy such as that found in Josephson junctions with incorporated effects of dissipation, damping and driving bias current (see also [31]). The proposed algorithms can be generalized to the case of complex-valued NLKGE (see [32]).

2. Invariance and conservation laws for NLKGE

2.1. *Energy conservation law for NLKGE*

The Cauchy problem for the NLKGE (1.2) in the 1-D case, with $U:\mathbb{R}\times\mathbb{R}\to\mathbb{R}$, is given by

$$U_{tt} - U_{xx} + G'(U) = 0\,, \tag{2.1a}$$

$$U(x,0) = f(x) \quad \text{and} \quad U_t(x,0) = g(x)\,, \tag{2.1b}$$

where

$$U_t(x,t) := \frac{\partial U(x,t)}{\partial t}\,, \qquad U_x(x,t) := \frac{\partial U(x,t)}{\partial x}\,,$$

with the requirements

$$|U_t|\,,\ |U_x| \to 0\,, \quad \text{as } |x|\to\infty\,. \tag{2.1c}$$

Since the conservation of invariants is the main focus of the present work, (2.1a) can be recast into the form of an evolution equation (see, e.g., [9])

$$\mathscr{T}_t + \mathscr{X}_x = 0\,, \tag{2.2}$$

in which $\mathscr{T}$ and $\mathscr{X}$ involve only the unknown functions and their x-derivatives, and in which there is only the first derivatives of the unknown functions with respect to time, as follows:

$$\frac{\partial U}{\partial t} = V\,, \qquad \frac{\partial V}{\partial t} = U_{xx} + G'(U)\,, \tag{2.3}$$

[10] See (2.40).

where V is the unknown velocity. The most basic fact about NLKGE is the conservation of energy. By multiplying the NLKGE (2.1a) by U_t, one has a local conservation energy law

$$\frac{\partial}{\partial t}\left[\tfrac{1}{2}(U_t)^2 + \tfrac{1}{2}(U_x)^2 + G(U)\right] - \frac{\partial}{\partial x}(U_x U_t) = 0 \tag{2.4a}$$

or

$$\frac{\partial U}{\partial t} = V\,, \qquad \frac{\partial}{\partial t}\left[\tfrac{1}{2}(V)^2 + \tfrac{1}{2}(U_x)^2 + G(U)\right] = \frac{\partial}{\partial x}(U_x V)\,. \tag{2.4b}$$

The local forms of conservation laws are important, since with appropriate boundary conditions, (2.1c) in the present case, they lead to conserved quantities or constants of motion. Indeed, an integration of (2.4a) with respect to the spatial variable x yields

$$\frac{\partial}{\partial t}\int_{-\infty}^{+\infty}\left[\tfrac{1}{2}(U_t)^2 + \tfrac{1}{2}(U_x)^2 + G(U)\right]\mathrm{d}x - \left[U_x U_t\right]_{-\infty}^{+\infty} = 0\,. \tag{2.5}$$

Let the energy density $\mathscr{E}$ (a conserved density) and the total energy $E(t)$ be defined as

$$\mathscr{E}(x,t) := \left[\tfrac{1}{2}(U_t)^2 + \tfrac{1}{2}(U_x)^2 + G(U)\right], \tag{2.6a}$$

$$E(t) := \int_{-\infty}^{+\infty}\mathscr{E}(x,t)\,\mathrm{d}x\,, \tag{2.6b}$$

respectively. By virtue of the boundary conditions (2.1c), one arrives at the conservation of the total energy from (2.5)

$$\frac{\partial}{\partial t}E(t) = 0 \;\Rightarrow\; E(t) = E(0) \quad \forall t \in \mathbb{R}\,, \tag{2.7a}$$

with $E(0)$ obtained from using the initial values in (2.1b),

$$E(0) := \int_{-\infty}^{+\infty}\left[\tfrac{1}{2}(g(x))^2 + \tfrac{1}{2}(f_x(x))^2 + G(f(x))\right]\mathrm{d}x\,. \tag{2.7b}$$

The requirements then on the functions f and g are such that $E(0)$ be finite.

2.2. *Conservation of linear momentum*

Multiplying NLKGE (2.1a) by U_x, one obtains another conservation law,

$$\frac{\partial}{\partial t}(U_x U_t) - \frac{\partial}{\partial x}\left[\tfrac{1}{2}(U_t)^2 + \tfrac{1}{2}(U_x)^2 - G(U)\right] = 0\,, \tag{2.8a}$$

or equivalently

$$\frac{\partial U}{\partial t} = V\,, \qquad \frac{\partial}{\partial t}(U_x V) = \frac{\partial}{\partial x}\left[\tfrac{1}{2}(V)^2 + \tfrac{1}{2}(U_x)^2 - G(U)\right], \tag{2.8b}$$

which is known as the conservation of linear momentum. Define the linear momentum density

and the total linear momentum as

$$\mathcal{M}(x, t) := U_t U_x\,, \qquad M(t) := \int_{-\infty}^{+\infty} \mathcal{M}(x, t)\,\mathrm{d}x\,, \tag{2.8c,d}$$

respectively, and introduce the quantity

$$\mathcal{P}(x, t) := \tfrac{1}{2}(U_t)^2 + \tfrac{1}{2}(U_x)^2 - G(U)\,. \tag{2.9}$$

Then the boundary conditions (2.1c) imply that

$$\mathcal{P}(x, t)\big|_{x=-\infty}^{x=+\infty} = \left[\tfrac{1}{2}(U_t)^2 + \tfrac{1}{2}(U_x)^2 - G(U)\right]_{x=-\infty}^{x=+\infty} = [G(U)]_{x=-\infty}^{x=+\infty}\,. \tag{2.10a}$$

Thus if the boundary conditions on U at $x = \pm\infty$ together with the function $G(\cdot)$ are such that

$$[G(U)]_{x=-\infty}^{x=+\infty} = 0\,, \tag{2.10b}$$

then the total linear momentum is conserved, i.e.,

$$\frac{\partial M(t)}{\partial t} = 0 \;\Rightarrow\; M(t) = M(0) \quad \forall t \in \mathbb{R}\,, \tag{2.11}$$

$$M(0) := \int_{-\infty}^{+\infty} f_x(x) \cdot g(x)\,\mathrm{d}x\,. \tag{2.12}$$

REMARK 2.1. There are two other local conservation laws which belong to the same category of energy conservation and momentum conservation:

$$\frac{\partial}{\partial t}\left[\tfrac{1}{2}(U_t + U_x)^2 + G(U)\right] - \frac{\partial}{\partial x}\left[\tfrac{1}{2}(U_t + U_x)^2 - G(U)\right] = 0\,, \tag{2.13a}$$

$$\frac{\partial}{\partial t}\left[\tfrac{1}{2}(U_t - U_x)^2 + G(U)\right] + \frac{\partial}{\partial x}\left[\tfrac{1}{2}(U_t - U_x)^2 + G(U)\right] = 0\,. \tag{2.13b}$$

It can be verified easily that among these four local conservation laws, only two are independent, such that each of the remaining two conservation laws is simply a linear combination of the independent ones. That is, considering the two related, independent conservation laws (2.4a) and (2.8a), we have that (2.13a) = (2.4a) − (2.8a) and (2.13b) = (2.4a) + (2.8a).

2.3. *Conservation of angular momentum*

It can be verified that the NLKGE is equivalent to

$$\frac{\partial}{\partial t}\left\{x\mathcal{E}(x, t) + t\mathcal{M}(x, t)\right\} - \frac{\partial}{\partial x}\left\{x\mathcal{M}(x, t) + t\mathcal{P}(x, t)\right\} = 0\,, \tag{2.14}$$

known as the conservation law for angular momentum in the 1-D case. From (2.14), the angular momentum density is defined as

$$\mathscr{A}(x, t) := x\mathscr{E}(x, t) + t\mathscr{M}(x, t)\,, \tag{2.15a}$$

and the total angular momentum as

$$A(t) := \int_{-\infty}^{+\infty} \mathscr{A}(x, t)\,\mathrm{d}x\,. \tag{2.15b}$$

Provided that

$$[x\mathscr{M}(x, t) + t\mathscr{P}(x, t)]_{-\infty}^{+\infty} = 0\,, \tag{2.16}$$

the total angular momentum is preserved at all time, i.e.,

$$\frac{\partial}{\partial t} A(t) = 0 \;\Rightarrow\; A(t) = A(0) \quad \forall t \in \mathbb{R}\,. \tag{2.17}$$

REMARK 2.2. For higher spatial dimensional problems, there are other conservation laws that belong to the above category, such as (see [30])

$$\int_{-\infty}^{+\infty} (x_k U_j - x_j U_k) U_t\,\mathrm{d}x = \text{const.}$$

The above three local conservation laws for the NLKGE, (2.4a), (2.8a), (2.14), are of the Noether type. Other invariant identities of the divergence type also exist (see [33, 34]).

3. Formalism for designing algorithms

3.1. Notations and definitions

Following Richtmeyer and Morton [35], we use different notations to distinguish the exact solution (U) from the finite difference solution (u). Let $h := \Delta x$ and $k := \Delta t$ be the increments in space and time, respectively, in a rectangular mesh. Also let $U_{p,q}$ denote that value of U at the point $(x, t) = (ph, qk)$,

$$U_{p,q} := U(ph, qk)\,. \tag{3.1}$$

The shorthand notation (p, q) designates the point (ph, qk), whereas (i, j) with $i, j \in \mathbb{Z}$ (set of integers) designates a mesh point in a finite difference mesh. In what follows, we let $p = i$ or $i \pm \frac{1}{2}a$, and $q = j$ or $j \pm \frac{1}{2}$. The finite difference counterpart of $U_{p,q}$ is $u_{p,q}$. The following convenient notation for partial differentiation is employed:

$$U_{i,j}^{(m,n)} := \frac{\partial^m}{\partial x^m} \frac{\partial^n}{\partial t^n} U(ih, jk)\,. \tag{3.2}$$

DEFINITION 3.1. At the point (p, q), the *averaging operator* in space $\mathfrak{A}_x$ and in time $\mathfrak{A}_t$ are defined as follows:

$$\mathfrak{A}_x u_{p,q} := \tfrac{1}{2}(u_{p+1/2,q} + u_{p-1/2,q}) = \tfrac{1}{2}[u((p+\tfrac{1}{2})h, qk) + u((p-\tfrac{1}{2})h, qk)]\,, \tag{3.3a}$$

$$\mathfrak{A}_t u_{p,q} := \tfrac{1}{2}(u_{p,q+1/2} + u_{p,q-1/2}) = \tfrac{1}{2}[u(ph,(q+\tfrac{1}{2})k) + u(ph,(q-\tfrac{1}{2})k)]\,. \tag{3.3b}$$

The notation $\mathfrak{A}$ without subscripts is used to designate that either $\mathfrak{A}_x$ or $\mathfrak{A}_t$ is valid. Also the notation u (3.3a,b) indicates the finite difference solution as already noted before.

DEFINITION 3.2. The *central difference operators* $\mathfrak{C}_x$ and $\mathfrak{C}_t$ at point (p, q) are defined to be

$$\mathfrak{C}_x^{(n+1)} u_{p,q} := \frac{1}{h}\,[\mathfrak{C}_x^{(n)} u_{p+1/2,q} - \mathfrak{C}_x^{(n)} u_{p-1/2,q}] \quad \forall n \in \mathbb{N}\,, \tag{3.4a}$$

$$\mathfrak{C}_t^{(n+1)} u_{p,q} := \frac{1}{k}\,[\mathfrak{C}_t^{(n)} u_{p,q+1/2} - \mathfrak{C}_t^{(n)} u_{p,q-1/2}] \quad \forall n \in \mathbb{N}\,, \tag{3.4b}$$

with $\mathfrak{C}_x^{(0)}$ and $\mathfrak{C}_t^{(0)}$ being the identity, i.e.,

$$\mathfrak{C}_x^{(0)} u_{p,q} = \mathfrak{C}_t^{(0)} u_{p,q} = u_{p,q}\,. \tag{3.4c}$$

REMARK 3.1. The customary definition of the central difference operator with the notation δ is without the factor $1/h$, i.e.,

$$\delta_x u_{p,q} := u_{p+1/2,q} - u_{p-1/2,q}\,, \tag{3.5}$$

and similarly for δ_t (e.g., [36, p. 20; 37]), whereas we include h and k in (3.4). Yet, other authors define δ_x similar to (3.4a), but using values at the mesh points $(i+1, j)$ and $(i-1, j)$ as in [25].

Both $\mathfrak{A}$ and $\mathfrak{C}^{(n)}$ are linear algebraic operators. The velocity V and its finite difference counterpart v are denoted by

$$V := \frac{\partial U}{\partial t}\,, \qquad v := \mathfrak{C}_t u\,. \tag{3.6}$$

3.2. *Preliminary assumptions*

The proposed formalism is founded upon approximations of a certain order of accuracy, second order in the present case, which are taken as equalities in the derivation of the algorithms. The validity of these assumptions is justified once the second order accuracy of the derived algorithms had been established. Keeping this remark in mind, we postulate the following rules.

ASSUMPTION 3.1 (Averaging rule). The identity operator **1** is equivalent to the averaging operator $\mathfrak{A}$ for the finite difference solution $u_{p,q}$, i.e.,

$$(\mathbf{1} - \mathfrak{A}_t) u_{p,q} = 0\,, \qquad (\mathbf{1} - \mathfrak{A}_x) u_{p,q} = 0\,,$$

$$p = i \text{ or } i \pm \tfrac{1}{2}\,, \qquad q = j \text{ or } j \pm \tfrac{1}{2}\,, \tag{3.7}$$

at all mesh points (i, j). In other words, the finite difference solution obeys *the averaging rule* for both x and t; we write $(\mathbf{1} - \mathfrak{A})u_{p,q} = 0$.

REMARK 3.3. Note that by taking the Taylor series expansion of the exact solution U, we have the relation

$$(\mathfrak{A}_t - \mathbf{1})U_{i,j+1/2} = \left.\frac{\partial^2 U}{\partial t^2}\right|_{i,j+1/2} \left(\frac{k}{2}\right)^2 + \mathrm{O}(k^4)\,,$$

or

$$(\mathfrak{A}_x - \mathbf{1})U_{i+1/2,j} = \left.\frac{\partial^2 U}{\partial x^2}\right|_{i+1/2,j} \left(\frac{h}{2}\right)^2 + \mathrm{O}(h^4)\,,$$

i.e.,

$$\|(\mathfrak{A} - \mathbf{1})U_{p,q}\| \sim \mathrm{O}(h^2, k^2)\,.$$

On the other hand, we cannot say similar things about the finite difference solution $u_{p,q}$ yet until the second order accuracy of the derived algorithms is established, i.e., $\|u_{p,q} - U_{p,q}\| \sim \mathrm{O}(h^2, k^2)$. This is why at present, we refer to the averaging rule as an assumption, which will be justified after the derivation of the algorithms.

ASSUMPTION 3.2 (Central difference rule).

$$\frac{\partial}{\partial t} U_{p,q} = \mathfrak{C}_t u_{p,q}\,, \qquad \frac{\partial}{\partial x} U_{p,q} = \mathfrak{C}_x u_{p,q}\,. \tag{3.8}$$

REMARK 3.4. The second order accuracy of the central difference rule, i.e.,

$$\left(\mathfrak{C}_t - \frac{\partial}{\partial t}\right)U_{i,j+1/2} = \left.\frac{\partial^2 U}{\partial t^2}\right|_{i+1/2,j} \left(\frac{k}{2}\right)^2 + \mathrm{O}(k^4)\,,$$

or

$$\left(\mathfrak{C}_x - \frac{\partial}{\partial x}\right)U_{i+1/2,j} = \left.\frac{\partial^2 U}{\partial x^2}\right|_{i+1/2,j} \left(\frac{h}{2}\right)^2 + \mathrm{O}(h^4)\,,$$

or

$$\|(\mathfrak{C} - \partial)U_{p,q}\| \sim \mathrm{O}(h^2, k^2)\,,$$

makes it consistent with the averaging rule, which is also second order accurate.

3.3. Basic lemmas

LEMMA 3.1 (*Commutativity of* $\mathfrak{C}_x^{(m)}$ *and* $\mathfrak{C}_t^{(n)}$). *At the point* (p, q), *we have*

$$\mathfrak{C}_t^{(m)}\mathfrak{C}_x^{(n)}u_{p,q} = \mathfrak{C}_x^{(n)}\mathfrak{C}_t^{(m)}u_{p,q}\,, \tag{3.9}$$

for all $m, n \in \mathbb{N}$.

PROOF. The case with $m = n = 0$ is obvious by (3.4c). For $m = n = 1$, we have

$$\mathfrak{C}_t^{(1)}(\mathfrak{C}_x^{(1)}u)_{p,q} = \mathfrak{C}_t^{(1)}\frac{1}{h}[u_{p+1/2,q} - u_{p-1/2,q}]$$

$$= \frac{1}{h}\left[\frac{1}{k}(u_{p+1/2,q+1/2} - u_{p+1/2,q+1/2}) + \frac{1}{k}(u_{p-1/2,q+1/2} - u_{p-1/2,q-1/2})\right]$$

$$= \mathfrak{C}_x^{(1)}\left[\frac{1}{k}\left((u_{p,q+1/2} - u_{p,q-1/2})\right]\right] = \mathfrak{C}_x^{(1)}(\mathfrak{C}_t^{(1)}u)_{p,q}\,. \tag{3.10}$$

By using (3.10) and (3.4), and by induction, one can prove (3.9). □

REMARK 3.3. Since $\mathfrak{C}_x^{(m)}$ and $\mathfrak{C}_x^{(n)}$ also commute, we can write unambiguously that (with (3.9) in mind)

$$\mathfrak{C}^{(m)}\mathfrak{C}^{(n)}u_{p,q} = \mathfrak{C}^{(n)}\mathfrak{C}^{(m)}u_{p,q}\,, \tag{3.11}$$

at any point (p, q), in which any combination of the subscripts x and t for $\mathfrak{C}$ is permissible.

LEMMA 3.2 (Commutativity of $\mathfrak{A}$ and $\mathfrak{C}^{(n)}$). The following relations

$$\mathfrak{A}_x\mathfrak{C}_x^{(n)}u_{p,q} = \mathfrak{C}_x^{(n)}\mathfrak{A}_x u_{p,q}\,, \qquad \mathfrak{A}_t\mathfrak{C}_t^{(n)}u_{p,q} = \mathfrak{C}_t^{(n)}\mathfrak{A}_t u_{p,q}\,, \tag{3.12a}$$

$$\mathfrak{A}_x\mathfrak{C}_t^{(n)}u_{p,q} = \mathfrak{C}_t^{(n)}\mathfrak{A}_x u_{p,q}\,, \qquad \mathfrak{A}_t\mathfrak{C}_x^{(n)}u_{p,q} = \mathfrak{C}_x^{(n)}\mathfrak{A}_t u_{p,q}\,, \tag{3.12b}$$

hold at any point (p, q), or unambiguously

$$\mathfrak{A}\mathfrak{C}^{(n)}u_{p,q} = \mathfrak{C}^{(n)}\mathfrak{A}u_{p,q}\,. \tag{3.12c}$$

PROOF. For $n = 1$, using the definitions (3.3a) and (3.4a) together with the linearity of $\mathfrak{A}$ and $\mathfrak{C}^{(n)}$, we have

$$\mathfrak{A}_x(\mathfrak{C}_x^{(1)}u)_{p,q} = \mathfrak{A}_x\left[\frac{1}{h}(u_{p+1/2,q} - u_{p-1/2,q})\right] = \frac{1}{h}(\mathfrak{A}_x u_{p+1/2,q} - \mathfrak{A}_x u_{p-1/2,q})$$

$$= \frac{1}{2h}[(u_{p+1,q} + u_{p,q}) - (u_{p,q} + u_{p-1,q})]$$

$$= \frac{1}{2h}[(u_{p+1,q} - u_{p,q}) + (u_{p,q} - u_{p-1,q})]$$

$$= \tfrac{1}{2}[\mathfrak{C}_x^{(1)}u_{p+1/2,q} + \mathfrak{C}_x^{(1)}u_{p-1/2,q}] = \tfrac{1}{2}\mathfrak{C}_x^{(1)}[u_{p+1/2,q} + u_{p-1/2,q}]$$

$$= \mathfrak{C}_x^{(1)}(\mathfrak{A}_x u)_{p,q}\,. \tag{3.13a}$$

For $n = 2$, the definition (3.4a), the linearity of $\mathfrak{A}_x$, and the above result (3.13a) lead to a confirmation of (3.12a):

$$\mathfrak{A}_x(\mathfrak{C}_x^{(2)}u)_{p,q} = \mathfrak{A}_x\left[\frac{1}{h}(\mathfrak{C}_x^{(1)}u_{p+1/2,q} - \mathfrak{C}_x^{(1)}u_{p-1/2,q})\right]$$

$$= \frac{1}{h}[\mathfrak{C}_x^{(1)}(\mathfrak{A}_x u)_{p+1/2,q} - \mathfrak{C}_x^{(1)}(\mathfrak{A}_x u)_{p-1/2,q}] = \mathfrak{C}_x^{(2)}(\mathfrak{A}_x u)_{p,q}\,, \tag{3.13b}$$

By induction, assume that $(3.12a)_1$ holds for $n=(k-1)$. Then by the same procedure as in (3.13b), one can show that $(3.12a)_1$ holds for $n=k$. The proof for $(3.12a)_2$ is identical. For $(3.12b)_1$, consider for $n=1$

$$\begin{aligned}\mathfrak{A}_x(\mathfrak{C}_t^{(1)}u)_{p,q} &= \mathfrak{A}_x\,\frac{1}{k}\,[u_{p,q+1/2}-u_{p,q-1/2}] = \frac{1}{k}[\mathfrak{A}_x u_{p,q+1/2}-\mathfrak{A}_x u_{p,q-1/2}]\\ &= \frac{1}{k}\,[\tfrac{1}{2}(u_{p+1/2,q+1/2}+u_{p-1/2,q+1/2})-\tfrac{1}{2}(u_{p+1/2,q-1/2}+u_{p+1/2,q-1/2})]\\ &= \mathfrak{C}_t^{(1)}[\tfrac{1}{2}(u_{p+1/2,q}+u_{p-1/2,q})] = \mathfrak{C}_t^{(1)}(\mathfrak{A}_x u)_{p,q}\,. \end{aligned} \tag{3.14a}$$

Next, for $n=2$, using the result (3.14a), one has

$$\begin{aligned}\mathfrak{A}_x(\mathfrak{C}_t^{(2)}u)_{p,q} &= \mathfrak{A}_x\,\frac{1}{k}\,[\mathfrak{C}_t^{(1)}u_{p,q+1/2}-\mathfrak{C}_t^{(1)}u_{p,q-1/2}]\\ &= \frac{1}{k}\,[\mathfrak{C}_t^{(1)}(\mathfrak{A}_x u)_{p,q+1/2}-\mathfrak{C}_t^{(1)}(\mathfrak{A}_x u)_{p,q-1/2}] = \mathfrak{C}_t^{(2)}(\mathfrak{A}_x u)_{p,q}\,. \end{aligned} \tag{3.14b}$$

By induction, $(3.12b)_1$ can be proved; the proof of $(3.12b)_2$ is similar. Equation (3.12c) follows from the convention on the subscripts of $\mathfrak{A}$ and $\mathfrak{C}$ as stated earlier. □

COROLLARY 3.1. If u satisfies the averaging rule (3.7), then at point (p, q)

$$\mathfrak{A}\mathfrak{C}^{(n)}u_{p,q} = \mathfrak{C}^{(n)}u_{p,q}\,, \tag{3.15}$$

with $p=i$ or $i\pm\frac{1}{2}$, and $q=j$ or $j\pm\frac{1}{2}$, for all mesh points (i,j).

PROOF. By using (3.7) in (3.12c) of Lemma 3.2. Note that the operators $\mathfrak{A}$ and $\mathfrak{C}$ in (3.15) are without subscripts x or t, i.e., any combination of subscripts x and t is permissible. □

COROLLARY 3.2. If u satisfies the averaging rule (3.7), then so do the central difference $\mathfrak{C}^{(n)}u_{p,q}$, i.e.,

$$\mathfrak{C}^{(n)}u_{p,q} = \tfrac{1}{2}\,[\mathfrak{C}^{(n)}u_{p+1/2,q}+\mathfrak{C}^{(n)}u_{p-1/2,q}]\,, \tag{3.16}$$

with $p=i$ or $i\pm\frac{1}{2}$, and $q=j$ or $j\pm\frac{1}{2}$, for all mesh points (i,j).

PROOF. From Corollary 3.1 and (3.12c) of Lemma 3.2, we have that

$$\begin{aligned}\mathfrak{C}^{(n)}u_{p,q} &= \mathfrak{A}\mathfrak{C}^{(n)}u_{p,q} = \mathfrak{C}^{(n)}\mathfrak{A}u_{p,q}\\ &= \mathfrak{C}^{(n)}[\tfrac{1}{2}(u_{p+1/2,q}+u_{p-1/2,q})] = \tfrac{1}{2}[\mathfrak{C}^{(n)}u_{p+1/2,q}+\mathfrak{C}^{(n)}u_{p-1/2,q}]\,. \end{aligned} \tag{3.17}$$

Note that (3.16) is valid for any subscript x or t. □

LEMMA 3.3 (Generalized Leibniz rule). Consider two functions $u, v\colon \mathbb{R}\times\mathbb{R}\to\mathbb{R}$. The central

difference operator when applied to the product function (uv) *at point* (p, q) *yields*

$$\mathfrak{C}^{(1)}(uv)_{p,q} = (\mathfrak{C}^{(1)}u_{p,q})(\mathfrak{A}v_{p,q}) + (\mathfrak{A}u_{p,q})(\mathfrak{C}^{(1)}v_{p,q}), \tag{3.18}$$

with $p = i$ *or* $i \pm \frac{1}{2}$ *and* $q = j$ *or* $j \pm \frac{1}{2}$, *for all mesh points* (i, j).

PROOF.

$$\begin{aligned}
\mathfrak{C}_x^{(1)}(uv)_{p,q} &= \frac{1}{h}[u_{p+1/2,q}v_{p+1/2,q} - u_{p-1/2,q}v_{p-1/2,q}] \\
&= \frac{1}{2h}[(u_{p+1/2,q} - u_{p-1/2,q})(v_{p+1/2,q} + v_{p-1/2,q}) \\
&\qquad + (u_{p+1/2,q} + u_{p-1/2,q})(v_{p+1/2,q} - v_{p-1/2,q})] \\
&= (\mathfrak{C}_x^{(1)}u_{p,q})(\mathfrak{A}_x v_{p,q}) + (\mathfrak{A}_x u_{p,q})(\mathfrak{C}_x^{(1)}v_{p,q}).
\end{aligned} \tag{3.19}$$

The proof for the case with $\mathfrak{C}_t^{(1)}$ is similar. □

COROLLARY 3.3 (Leibniz rule). If u *satisfies the averaging rule (3.7), then*

$$\mathfrak{C}^{(1)}(fg)_{p,q} = (\mathfrak{C}^{(1)}f_{p,q})g_{p,q} + f_{p,q}(\mathfrak{C}^{(1)}g_{p,q}), \tag{3.20}$$

with $p = i$ *or* $i \pm \frac{1}{2}$ *and* $q = j$ *or* $j \pm \frac{1}{2}$, *for all mesh points* (i, j).

REMARK 3.5. Relation (3.18) is basically the discrete counterpart of the derivation property in the continuous case. Together with the averaging rule (3.7), the preservation of certain properties of continuous operators by their discrete counterparts as exemplified by (3.20) will play a crucial role in the development of invariant-conserving algorithms that follows.

LEMMA 3.4 (Chain rule). For a compound function $G(u)$, *where* $u : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, *the following relations hold for the operator* $\mathfrak{C}$:

$$\begin{aligned}
\{\mathfrak{C}_t G(u)\}_{p,q} &= \{\mathfrak{C}_u G(u)|_{p=\text{fixed}}\mathfrak{C}_t u\}_{p,q}, \\
\{\mathfrak{C}_x G(u)\}_{p,q} &= \{\mathfrak{C}_u G(u)|_{q=\text{fixed}}\mathfrak{C}_x u\}_{p,q}.
\end{aligned} \tag{3.21}$$

PROOF. Straightforward. Consider $(3.21)_1$:

$$\begin{aligned}
\mathfrak{C}_t G(u)_{p,q} &= \frac{G(u_{p,q+1/2}) - G(u_{p,q+1/2})}{(q + \frac{1}{2})k - (q - \frac{1}{2})k} \\
&= \frac{G(u_{p,q+1/2}) - G(u_{p,q-1/2})}{u_{p,q+1/2} - u_{p,q-1/2}} \frac{u_{p,q+1/2} - u_{p,q-1/2}}{k} \\
&= \{\mathfrak{C}_u G(u)|_{p=\text{fixed}}\mathfrak{C}_t u\}_{p,q}. \quad \square
\end{aligned}$$

4. Invariant-conserving algorithms for NLKGE

4.1. Energy-conserving algorithm I

4.4.1 Derivation and interpretation

The starting point is the evolution equation (2.4b) expressing local energy conservation. We construct a finite difference discretization of (2.4b) at the point $(i, j+\frac{1}{2})$ using only the averaging operator (3.3) and central difference operator (3.4). An interpretation of the algorithm is given after a derivation of its recurrence formula.

THEOREM 4.1 (Algorithm I). By applying the averaging rule (3.7) to u and v, and the central difference rule (3.8) to approximate $\partial/\partial t$ and $\partial/\partial x$ at the point $(i, j+\frac{1}{2})$, the following algorithm is obtained from the evolution equation (2.4b):

$$\frac{1}{k}(u_{i,j+1}-u_{i,j}) = \frac{1}{2}(v_{i,j+1}+v_{i,j}),$$

$$\frac{1}{k}(v_{i,j+1}-v_{i,j}) + \frac{G(u_{i,j+1})-G(u_{i,j})}{u_{i,j+1}-u_{i,j}} = \frac{1}{2h^2}[(u_{i+1,j+1}-2u_{i,j+1}+u_{i-1,j+1})+(u_{i+1,j}-2u_{i,j}+u_{i-1,j})]. \tag{4.1}$$

which involves only the values of the unknowns u and v at the mesh points (i, j).

PROOF. Applying (3.7) and (3.8) to $(2.4b)_1$ at point $(i, j+\frac{1}{2})$, one obtains

$$\mathfrak{C}_t^{(1)} u_{i,j+1/2} = v_{i,j+1/2} = \mathfrak{A}_t v_{i,j+1/2}, \tag{4.2a}$$

which leads to $(4.1)_1$, and is generally referred to as the trapezoidal rule applying to $(2.4b)_1$. The central difference operators $\mathfrak{C}_t^{(1)}$ and $\mathfrak{C}_t^{(1)}$ when applied to $(2.4b)_2$ at point $(i, j+\frac{1}{2})$ yield

$$\mathfrak{C}_t^{(1)}[\tfrac{1}{2}(v)^2 + \tfrac{1}{2}(\mathfrak{C}_x^{(1)}u)^2 + G(u)]_{i,j+1/2} = \mathfrak{C}_x^{(1)}[(\mathfrak{C}_x^{(1)}u)v]_{i,j+1/2}. \tag{4.2b}$$

We now examine each term of (4.2b). Using the generalized Leibniz rule (3.18) and the trapezoidal rule (4.2a), the first term in (4.2b) becomes

$$\mathfrak{C}_t^{(1)}[\tfrac{1}{2}(v)^2]_{i,j+1/2} = (\mathfrak{C}_t^{(1)}v_{i,j+1/2})(\mathfrak{A}_t v_{i,j+1/2}) = (\mathfrak{C}_t^{(1)}v_{i,j+1/2})(\mathfrak{C}_t^{(1)}u_{i,j+1/2}). \tag{4.2c}$$

Next, for the second term in (4.2b), we apply the generalized Leibniz rule (3.18), the commutativity (3.9), the commutativity (3.12c), the trapezoidal rule (4.2a), and the averaging rule (3.7) to obtain

$$\begin{aligned}\mathfrak{C}_t^{(1)}[\tfrac{1}{2}(\mathfrak{C}_x^{(1)}u)^2]_{i,j+1/2} &= [\mathfrak{C}_t^{(1)}(\mathfrak{C}_x^{(1)}u)_{i,j+1/2}][\mathfrak{A}_t(\mathfrak{C}_x^{(1)}u)_{i,j+1/2}] \\ &= [\mathfrak{C}_x^{(1)}(\mathfrak{C}_t^{(1)}u)_{i,j+1/2}][\mathfrak{C}_x^{(1)}(\mathfrak{A}_t u)_{i,j+1/2}] \\ &= [\mathfrak{C}_x^{(1)}(\mathfrak{A}_t v)_{i,j+1/2}](\mathfrak{C}_x^{(1)}u_{i,j+1/2}) = (\mathfrak{C}_x^{(1)}v_{i,j+1/2})(\mathfrak{C}_x^{(1)}u_{i,j+1/2}).\end{aligned} \tag{4.2d}$$

A direct application of the chain rule $(3.21)_1$ leads to

$$\mathfrak{C}_t^{(1)} G(u)_{i,j+1/2} = \{\mathfrak{C}_u^{(1)} G(u)|_{i=\text{fixed}} \mathfrak{C}_t^{(1)} u\}_{i,j+1/2} . \tag{4.2e}$$

For the fourth term on the right-hand side of (4.2b), using the generalized Leibniz rule (3.18), the commutativity (3.12c), the averaging rule (3.7), and the trapezoidal rule (4.2a), we arrive at

$$\begin{aligned}
\mathfrak{C}_x^{(1)}[(\mathfrak{C}_x^{(1)}u)v]_{i,j+1/2} &= [\mathfrak{C}_x^{(1)}(\mathfrak{C}_x^{(1)}u)_{i,j+1/2}](\mathfrak{A}_x v_{i,j+1/2}) + [\mathfrak{A}_x(\mathfrak{C}_x^{(1)}u)_{i,j+1/2}](\mathfrak{C}_x^{(1)} v_{i,j+1/2}) \\
&= [\mathfrak{C}_x^{(2)} u_{i,j+1/2}](v_{i,j+1/2}) + [\mathfrak{C}_x^{(1)}(\mathfrak{A}_x u)_{i,j+1/2}](\mathfrak{C}_x^{(1)} v_{i,j+1/2}) \\
&= [\mathfrak{C}_x^{(2)} u_{i,j+1/2}](\mathfrak{A}_t v_{i,j+1/2}) + [\mathfrak{C}_x^{(1)} u_{i,j+1/2}](\mathfrak{C}_x^{(1)} v_{i,j+1/2}) \\
&= (\mathfrak{C}_x^{(2)} u_{i,j+1/2})(\mathfrak{C}_t^{(1)} u_{i,j+1/2}) + (\mathfrak{C}_x^{(1)} v_{i,j+1/2})(\mathfrak{C}_x^{(1)} u_{i,j+1/2}) .
\end{aligned} \tag{4.2f}$$

Thus, (4.2c–f) together transform (4.2b) into

$$\begin{aligned}
&(\mathfrak{C}_t^{(1)} v_{i,j+1/2})(\mathfrak{C}_t^{(1)} u_{i,j+1/2}) + (\mathfrak{C}_u^{(1)} G(u_{i,j+1/2})|_{i=\text{fixed}} \mathfrak{C}_t^{(1)} u_{i,j+1/2}) \\
&\quad = (\mathfrak{C}_x^{(2)} u_{i,j+1/2})(\mathfrak{C}_t^{(1)} u_{i,j+1/2}) ,
\end{aligned} \tag{4.2g}$$

which, by assuming that $u_{i,j+1} - u_{i,j} = 0$, can be rewritten as

$$\mathfrak{C}_t^{(1)} v_{i,j+1/2} + \frac{G(u_{i,j+1}) - G(u_{i,j})}{u_{i,j+1/2} - u_{i,j}} = \mathfrak{C}_x^{(2)} u_{i,j+1/2} . \tag{4.2h}$$

Next, by the averaging rule (3.7) together with (3.4a), we have that

$$\begin{aligned}
\mathfrak{C}_x^{(2)} u_{i,j+1/2} &= \mathfrak{C}_x^{(2)} \mathfrak{A}_t u_{i,j+1/2} = \mathfrak{C}_x^{(2)} \tfrac{1}{2}[u_{i,j+1} + u_{i,j}] \\
&= \frac{1}{2h^2}[(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}) + (u_{i+1,j} - 2u_{i,j} + u_{i-1,j})] .
\end{aligned} \tag{4.2i}$$

Thus, substitution of (4.2i) into (4.2h) completes the proof. □

COROLLARY 4.1 (Recurrence formula). The algorithm (4.1) can be rewritten in the following form that involves only u:

$$\frac{1}{k^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) = \tfrac{1}{2}(f_{i,j} + f_{i,j-1}) , \tag{4.3a}$$

where

$$\begin{aligned}
f_{i,j} := &\frac{1}{2h^2}[(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}) + (u_{i+1,j} - 2u_{i,j} + u_{i-1,j})] \\
&- \frac{G(u_{i,j+1}) - G(u_{i,j})}{u_{i,j+1} - u_{i,j}} .
\end{aligned} \tag{4.3b}$$

PROOF. From (4.1) and using the definition (4.3b), it follows that

$$v_{i,j+1} = \frac{1}{k}(u_{i,j+1} - u_{i,j}) + \frac{k}{2} f_{i,j}\,, \tag{4.4a}$$

which implies that at the previous time step j, or more precisely at point (i, j), we have

$$v_{i,j} = \frac{1}{k}(u_{i,j} - u_{i,j-1}) + \frac{k}{2} f_{i,j-1}\,. \tag{4.4b}$$

Then substitution of (4.4a,b) in $(4.1)_1$ leads to (4.3a). □

REMARK 4.2. Geometry of Algorithm I. Recall that the second order finite difference operator in time at the mesh point (i, j) is

$$\mathfrak{F}_t^{(2)} u_{i,j} = \frac{1}{k^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1})\,, \tag{4.5a}$$

which is the left-hand-side of (4.3a). Similarly, we can write

$$f_{i,j} = \tfrac{1}{2}[\mathfrak{F}_x^{(2)} u_{i,j+1} + \mathfrak{F}_x^{(2)} u_{i,j}] - \frac{G(u_{i,j+1}) - G(u_{i,j})}{u_{i,j+1} - u_{i,j}}\,, \tag{4.5b}$$

which can be interpreted as the average of two second order central differences in space at mesh points $(i, j+1)$ and (i, j), minus a forward difference quotient in time that approximates $G'(u)$ at mesh point (i, j), i.e.,

$$G'(u_{i,j}) = \frac{\mathrm{d}G(u_{i,j})}{\mathrm{d}u} \approx \frac{G(u_{i,j+1}) - G(u_{i,j})}{u_{i,j+1} - u_{i,j}}\,. \tag{4.5c}$$

The interpretation of $f_{i,j-1}$ is similar: The average of two second order central differences in space at mesh points (i, j) and $(i, j-1)$, minus a backward difference quotient in time that approximates $G'(u)$ at mesh point (i, j). The right-hand side of (4.3a) is the average of $f_{i,j}$ and $f_{i,j-1}$. Thus using (4.5a,b), we can rewrite (4.3a) into the form

$$\begin{aligned}&\mathfrak{F}_t^{(2)} u_{i,j} - \frac{1}{4}[\mathfrak{F}_x^{(2)} u_{i,j+1} + 2\mathfrak{F}_x^{(2)} u_{i,j} + \mathfrak{F}_x^{(2)} u_{i,j-1}]\\ &\quad + \frac{1}{2}\left[\frac{G(u_{i,j+1}) - G(u_{i,j})}{u_{i,j+1} - u_{i,j}} + \frac{G(u_{i,j}) - G(u_{i,j-1})}{u_{i,j} - u_{i,j-1}}\right] = 0\,.\end{aligned} \tag{4.5d}$$

A comparison of the three terms in (4.5d) to the three terms in the NLKGE (2.1a) reveals an obvious interpretation for the finite difference approximation. Figure 1 summarizes the geometric interpretation of Algorithm I. A final note before we close the remark: The form (4.5d) of Algorithm I can also be derived by applying directly the averaging operators and central difference operators to the second order form of the local conservation law (2.4a) instead of the first order form $(2.4b)_2$.

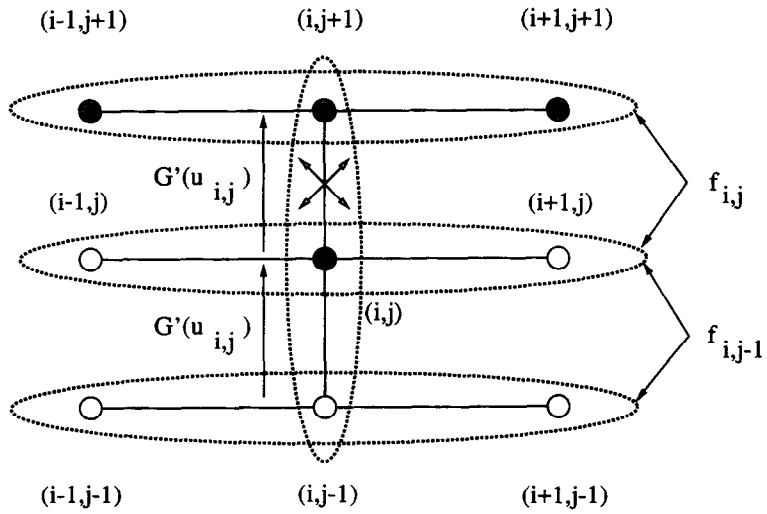

Fig. 1. *Geometry of Algorithm I.* The algorithm is constructed from point $(i, j+1/2)$ indicated by the arrowed cross. Taylor series are expanded about the black mesh point (i, j). The 3 unknowns are the 3 gray mesh points at time $t=(j+1)k$. The 4 second order central differences linking 3 mesh points each are indicated by the 4 dotted-line ellipses: the vertical one in time, and the 3 horizontal ones in space. The forward and the backward difference quotients approximating $G'(u_{i,j})$ are represented by vertical arrows.

4.1.2. Accuracy

We prove that Algorithm I is second order accurate in both space and time.

THEOREM 4.2 (Order of accuracy). The algorithm (4.1) is consistent with the NLKGE (2.1a), and is accurate of order $(2,2)$, *i.e., the truncation error is* $O(h^2)+O(k^2)$.

PROOF. Recall the convenient notation for partial differentiation $U_{i,j}^{(m,n)}$ introduced in (3.2). The Taylor series expansion of $U_{i,j\pm1}$ about the mesh point (i, j)

$$U_{i,j\pm1} = U_{i,j}^{(0,0)} \pm kU_{i,j}^{(0,1)} + \frac{k^2}{2} U_{i,j}^{(0,2)} \pm \frac{k^3}{3!} U_{i,j}^{(0,3)} + O(k^4) \tag{4.6a}$$

leads to the following expression for (4.5a):

$$\mathfrak{S}_t^{(2)} U_{i,j} = U_{i,j}^{(0,2)} + O(k^2)\,. \tag{4.6b}$$

Similarly, we have for the first three terms on the right-hand side of (4.5d)

$$\mathfrak{S}_x^{(2)} U_{i,q} = U_{i,q}^{(2,0)} + O(h^2)\,, \quad \text{for } q = j+1,\ j,\ j-1\,. \tag{4.6c}$$

Writing $G(U(ih, jk)) = (G \circ U)_{i,j}$, we have that

$$(G\circ U)_{i,j\pm1} = (G\circ U)_{i,j} \pm k(G\circ U)_{i,j}^{(0,1)} + \frac{k^2}{2}(G\circ u)_{i,j}^{(0,2)} + O(k^3)\,,$$

$$= (G \circ U)_{i,j} \pm kG'(U_{i,j})U_{i,j}^{(0,1)} + \frac{k^2}{2}\left[G'(U_{i,j})U_{i,j}^{(0,2)} + G''(U_{i,j})(U_{i,j}^{(0,1)})^2\right] + O(k^3)\,. \tag{4.6d}$$

Then using (4.6a), we obtain the truncation error of the last two terms in (4.5d),

$$\frac{(G \circ U)_{i,j+1} - (G \circ U)_{i,j}}{U_{i,j+1} - U_{i,j}} = G'(U_{i,j}) + \frac{k}{2} G''(U_{i,j})U_{i,j}^{(0,1)} + O(k^2)\,, \tag{4.6e}$$

$$\frac{(G \circ U)_{i,j} - (G \circ U)_{i,j-1}}{U_{i,j} - U_{i,j-1}} = G'(U_{i,j}) - \frac{k}{2} G''(U_{i,j})U_{i,j}^{(0,1)} + O(k^2)\,. \tag{4.6f}$$

Denote[11]

$$L_{i,j}^{(I)}(U) := \mathfrak{C}_t^{(2)} u_{i,j} - \frac{1}{4}[\mathfrak{C}_x^{(2)} u_{i,j+1} + 2\mathfrak{C}_x^{(2)} u_{i,j} + \mathfrak{C}_x^{(2)} u_{i,j-1}] + \frac{1}{2}\left[\frac{G(u_{i,j+1}) - G(u_{i,j})}{u_{i,j+1} - u_{i,j}} + \frac{G(u_{i,j}) - G(u_{i,j-1})}{u_{i,j} - u_{i,j-1}}\right]. \tag{4.6g}$$

Substituting (4.6b,c) and (4.6e,f) in (4.6g) and considering the fact that

$$U_{i,j}^{(0,2)} - U_{i,j}^{(2,0)} + G'(U_{i,j}) = 0\,, \tag{4.6h}$$

we obtain

$$\|L_{i,j}^{(I)}(U)\| \sim O(h^2, k^2)\,. \quad \square \tag{4.6i}$$

REMARK 4.3. It makes sense now to consider the meaning of the averaging rule (3.7), i.e., how does (3.7) fit with the second order accuracy of the algorithm since we have assumed that the variables u and v satisfy (3.7). It is easy to see that both the averaging operator $\mathfrak{A}$ and the central difference operator $\mathfrak{C}$ are second order accurate with respect to h and k; hence any composition of these two linear operators is also of second order accuracy. Also, as already noted about (4.2a), the central difference operator and the averaging operator when applied to a first order differential equation such as $(2.4b)_1$ give rise to the trapezoidal rule.

4.1.3. Energy conservation, boundary conditions, stability

We show that the conservation of the total system energy is, with appropriate boundary conditions, a by-product of the derivation of the algorithm starting from the local conservation law. A discrete energy can be defined as follows:

$$E_{N,j}^{(I)} := h \sum_{i=-N}^{N} \mathscr{E}_{i,j}\,, \tag{4.7a}$$

[11] The superscript (I) refers to Algorithm I.

for some large integer N, where $\mathscr{E}_{i,j} = \mathscr{E}(ih, jk)$ with $\mathscr{E}(x, t)$ defined in (2.6a). It can be seen that $E^{(\mathrm{I})}_{N,j}$ is a truncated series approximating the exact energy $E(jk)$ in (2.6b) using the trapezoidal rule. Using (3.7) and (3.12a), we obtain

$$\begin{aligned}\mathscr{E}_{i,j} &= \frac{1}{2}(\mathfrak{C}_t^{(1)}\mathfrak{A}_t u_{i,j})^2 + \frac{1}{2}(\mathfrak{C}_x^{(1)}\mathfrak{A}_x u_{i,j})^2 + G(u_{i,j}) \\ &= \frac{1}{4k^2}(u_{i,j+1} - u_{i,j-1})^2 + \frac{1}{4h^2}(u_{i+1,j} - u_{i-1,j})^2 + G(u_{i,j})\,. \end{aligned} \tag{4.7b}$$

With the same formalism employed to derive Algorithm I, the following finite difference form of the local conservation law (2.4a)

$$\mathfrak{C}_t^{(1)}\mathscr{E}_{i,j+1/2} - \mathfrak{C}_x^{(1)}(\mathfrak{C}_x^{(1)}u\mathfrak{C}_t^{(1)}u)_{i,j+1/2} = 0 \tag{4.8a}$$

leads to

$$\frac{h}{k}[\mathscr{E}_{i,j+1} - \mathscr{E}_{i,j}] - [(\mathfrak{C}_x^{(1)}u\mathfrak{C}_t^{(1)}u)_{i+1/2,j+1/2} - (\mathfrak{C}_x^{(1)}u\mathfrak{C}_t^{(1)}u)_{i-1/2,j+1/2}] = 0\,. \tag{4.8b}$$

In fact, (4.8b) can be considered as a discrete local conservation law, a counterpart of (2.4a). Consider an FD mesh with spatial mesh points ranging from $i = -N - 1$ to $i = N + 1$, i.e., $2(N + 1)$ segments in the space dimension. Summing (4.8b) from $i = -N$ to $i = N$, and in view of the definition of the discrete energy in (4.7), we have

$$\frac{1}{k}[E^{(\mathrm{I})}_{N,j+1} - E^{(\mathrm{I})}_{N,j}] - [(\mathfrak{C}_x^{(1)}u\mathfrak{C}_t^{(1)}u)_{N+1/2,j+1/2} - (\mathfrak{C}_x^{(1)}u\mathfrak{C}_t^{(1)}u)_{-N-1/2,j+1/2}] = 0\,. \tag{4.9}$$

A choice of symmetric boundary conditions such that the last bracketed term in (4.9) vanishes, yields the conservation of the discrete energy $E^{(\mathrm{I})}_{N,\cdot}$,[12] i.e., Theorem 4.3.

THEOREM 4.3 (Energy conservation). Algorithm I described in (4.1) or (4.5d) conserves the discrete energy $E^{(\mathrm{I})}_{N,\cdot}$ defined in (4.7) in the sense that

$$E^{(\mathrm{I})}_{N,j+1} = E^{(\mathrm{I})}_{N,j} \quad \forall j\,, \tag{4.10a}$$

provided that the boundary conditions are chosen to be symmetric such that

$$(\mathfrak{C}_x^{(1)}u\mathfrak{C}_t^{(1)}u)_{N+1/2,j+1/2} = (\mathfrak{C}_x^{(1)}u\mathfrak{C}_t^{(1)}u)_{-N-1/2,j+1/2}\,. \tag{4.10b}$$

The symmetry condition (4.10b) is essential to ensure conservation of a discrete energy. We mention here four possible choices of boundary conditions. While the symmetry condition (4.10b) itself leads to a nonlinear constraint equation to be imposed on the unknowns, one can choose simpler *linear* constraint equations such that (4.10b) is still satisfied, and whose elimination is much easier. *Second choice:* noting that, with the help of the averaging rule (3.7),

[12] The first subscript N in $E^{(\mathrm{I})}_{N,\cdot}$ is fixed; the second subscript, not fixed, is represented by a 'dot'. The 'dot' is used when we do not want to give any particular attention to a subscript. The superscript (I) refers to Algorithm I.

$$\mathfrak{C}_x^{(1)} u_{N+1/2,j+1/2} = \mathfrak{C}_x^{(1)} (\mathfrak{A}_t u)_{N+1/2,j+1/2} = \frac{1}{2}[\mathfrak{C}_x^{(1)} u_{N+1/2,j+1} + \mathfrak{C}_x^{(1)} u_{N+1/2,j}]$$
$$= \frac{1}{2h}[(u_{N+1,j+1} - u_{N,j+1}) + (u_{N+1,j} - u_{N,j})], \tag{4.11}$$

and similarly for $\mathfrak{C}_x^{(1)} u_{-N-1/2,j+1/2}$, we can approximate the boundary conditions $(2.1c)_2$, i.e., $u_x \to 0$ as $x \to \pm\infty$, by setting

$$\mathfrak{C}_x^{(1)} u_{N+1/2,j+1/2} = \mathfrak{C}_x^{(1)} u_{-N-1/2,j+1/2} \equiv 0 \quad \forall j, \tag{4.12a}$$

obtained by letting[13]

$$\begin{aligned} u_{N+1,j+1} - u_{N,j+1} &= -(u_{N+1,j} - u_{N,j}), \\ u_{-N,j+1} - u_{-N-1,j+1} &= -(u_{-N,j} - u_{-N-1,j}). \end{aligned} \tag{4.12b}$$

A *third choice* would be to make

$$\mathfrak{C}_t^{(1)} u_{N+1/2,j+1/2} = \mathfrak{C}_t^{(1)} u_{-N-1/2,j+1/2} \equiv 0 \quad \forall j, \tag{4.13a}$$

to approximate the boundary condition $(2.1c)_1$, i.e., $u_t \to 0$ as $x \to \pm\infty$, with

$$\mathfrak{C}_t^{(1)} u_{\pm N\pm 1/2,j+1/2} = \frac{1}{2k}[(u_{\pm N\pm 1,j+1} + u_{\pm N,j+1}) - (u_{\pm N\pm 1,j} + u_{\pm N,j})], \tag{4.13b}$$

by simply choosing to let

$$u_{N+1,j+1} + u_{N,j+1} = u_{N+1,j} + u_{N,j}, \qquad u_{-N-1,j+1} + u_{-N,j+1} = u_{-N-1,j} + u_{-N,j}. \tag{4.13c}$$

Finally, instead of enforcing zero derivative as in (4.21a) or (4.13a) to approximate the exact boundary conditions (2.1c), and since N is a finite number, a *fourth choice* is to enforce the following *less restrictive* symmetric conditions:

$$\begin{aligned} \mathfrak{C}_x^{(1)} u_{N+1/2,j+1/2} &= \mathfrak{C}_x^{(1)} u_{-N-1/2,j+1/2} \neq 0 \quad \forall j, \\ \mathfrak{C}_t^{(1)} u_{N+1/2,j+1/2} &= \mathfrak{C}_t^{(1)} u_{-N-1/2,j+1/2} \neq 0 \quad \forall j, \end{aligned} \tag{4.14a}$$

to ensure that (4.10b) is satisfied, and which then lead to the *linear* constraint equations

$$(u_{N+1,j+1} - u_{N,j+1}) - (u_{-N,j+1} - u_{-N-1,j+1}) = -(u_{N+1,j} - u_{N,j}) + (u_{-N,j} - u_{-N-1,j}), \tag{4.14b}$$
$$(u_{N+1,j+1} + u_{N,j+1}) - (u_{-N,j+1} + u_{-N-1,j+1}) = (u_{N+1,j} + u_{N,j}) - (u_{-N,j} + u_{-N-1,j}),$$

[13] A set of boundary conditions that also yields (4.12a), easier to implement, but more restrictive, would be $u_{N+1,q} = u_{N,q}$ and $u_{-N-1,q} = u_{-N,q}$, $\forall q$.

Note that the approximated derivatives in (4.14a) are not forced to vanish due to the location of the boundaries at a finite range.[14] It is easy to eliminate (4.14b) from the system of algebraic equations for the unknowns $u_{\cdot,j+1}$ ($j+1$ is fixed) resulting from Algorithm I.

To study the stability of Algorithm I, we consider the following un-normalized linear Klein–Gordon equation (ULKGE):

$$U_{tt} - U_{xx} + \epsilon U = 0\,, \tag{4.15}$$

where $\epsilon \geqslant 0$.[15] One can think of (4.15) as a linearized version of the sine-Gordon equation with a factor ϵ. For stability analysis of numerical algorithms, we consider only equations whose solution does not blow up; it is known that the NLKGE (2.1a) with $G'(u) = -\sin u$ is unstable[16] (e.g., [38]). Hence we do not consider the case with $\epsilon < 0$.

THEOREM 4.4 (Linear stability). Algorithm I as described in (4.1) or (4.5d) is unconditionally stable for the ULKGE (4.15).

PROOF. We will work with the form (4.5d) of Algorithm I, instead of (4.1). With $G(u) = \epsilon(u^2/2)$, (4.5d) takes the form

$$\mathfrak{F}_t^{(2)} u_{i,j} - \frac{1}{4}[\mathfrak{F}_x^{(2)} u_{i,j+1} + 2\mathfrak{F}_x^{(2)} u_{i,j} + \mathfrak{F}_x^{(2)} u_{i,j-1}] + \frac{\epsilon}{2}[u_{i,j+1} + 2u_{i,j} + u_{i,j} + u_{i,j-1}] = 0\,. \tag{4.16a}$$

The method of proof is Von Neumann (Fourier series) analysis, according to which we set (e.g., [25])

$$u_{i,j} = \mathrm{e}^{\mathrm{i}\theta\sqrt{-1}} g^j\,, \tag{4.16b}$$

where $\theta \in [-\pi, \pi]$ is the continuous variable in the transformed space, and g the amplification factor. It follows that

$$\mathfrak{F}_t^{(2)} u_{i,j} = \frac{1}{k^2}\,\mathrm{e}^{\mathrm{i}\theta\sqrt{-1}} g^j [g - 2 + g^{-1}]\,, \tag{4.16c}$$

$$\mathfrak{F}_x^{(2)} u_{i,q} = -\frac{1}{h^2}\,\mathrm{e}^{\mathrm{i}\theta\sqrt{-1}} g^q 4\sin^2\frac{\theta}{2}\,, \qquad q = j+1,\ j,\ j-1\,, \tag{4.16d}$$

$$u_{i,j+1} + 2u_{i,j} + u_{i,j-1} = \mathrm{e}^{\mathrm{i}\theta\sqrt{-1}} g^j [g + 2 + g^{-1}]\,. \tag{4.16e}$$

Define

$$\lambda := k/h\,. \tag{4.16f}$$

[14] Compare (4.12b) and (4.13c) with (4.14b).

[15] For an example of the ULKGE, we refer to [7, p. 247]. By the transformation of variables $\tau = t\sqrt{\epsilon}$ and $\xi = x\sqrt{\epsilon}$, we recover the (normalized) linear Klein–Gordon equation $u_{\tau\tau} - u_{\xi\xi} + u = 0$.

[16] Note the negative sign.

Then substitution of (4.16c,d,e) into (4.16a) yields

$$[g-2+g^{-1}]+\left(\lambda^2\sin^2\frac{\theta}{2}+\frac{\epsilon}{2}\right)[g+2+g^{-1}]=0\,, \tag{4.16g}$$

$$\Rightarrow\quad (g^{1/2}-g^{-1/2})^2=-\frac{4\left(\lambda^2\sin^2\dfrac{\theta}{2}+\dfrac{\epsilon}{2}\right)}{1+\lambda^2\sin^2\dfrac{\theta}{2}+\dfrac{\epsilon}{2}}=:-\alpha\,, \tag{4.16h}$$

$$\Rightarrow\quad g\mp\sqrt{-\alpha}\,g^{1/2}-1=0\,, \tag{4.16i}$$

$$\Rightarrow\quad g^{1/2}=\frac{\pm\sqrt{-\alpha}\pm\sqrt{4-\alpha}}{2}\,. \tag{4.16j}$$

Since $0\leqslant\alpha\leqslant 4$, $\forall\epsilon\geqslant 0$ and $\forall\lambda$, the norm of the complex number $g^{1/2}$ is unitary, i.e., $\|g^{1/2}\|=1$, which is the condition for stability. □

REMARK 4.4. A variant of Algorithm I. Instead of evaluating the approximate to $G'(u_{i,j})$ as in the last bracketed term of (4.5d), we consider using $G'(u_{i,j})$ directly as follows:

$$\mathfrak{G}_t^{(2)}u_{i,j}-\frac{1}{4}[\mathfrak{G}_x^{(2)}u_{i,j+1}+2\mathfrak{G}_x^{(2)}u_{i,j}+\mathfrak{G}_x^{(2)}u_{i,j-1}]+G'(u_{i,j})=0\,. \tag{4.17a}$$

The advantage of doing this is to avoid having to solve a system of nonlinear algebraic equations as arising from (4.1) or (4.5d), because the term $g(u_{i,j+1})$ is no longer present while $G'(u_{i,j})$ is a known quantity at time $t=(j+1)k$. It can be easily seen from the proof of Theorem 4.2 that this variant of Algorithm I is also second order accurate. The algorithm (4.17a) is, on the other hand, only conditionally stable for the ULKGE (4.15), although with a particularity that the stability condition does *not* depend on the spatial grid size h. Indeed, with $G'(u_{i,j})=\epsilon u_{i,j}$, an analysis as in Theorem 4.4 above leads to

$$(g^{1/2}-g^{-1/2})^2=-\frac{4\lambda^2\sin^2\dfrac{\theta}{2}+\epsilon k^2}{1+\lambda^2\sin^2\dfrac{\theta}{2}}=:-\alpha\,. \tag{4.17b}$$

Then the condition for $0\leqslant\alpha\leqslant 4$ is

$$k\leqslant\frac{2}{\sqrt{\epsilon}}\,, \tag{4.17c}$$

which is a very mild stability condition due to the absence of h. As $\epsilon\downarrow 0$, the stability limit on k becomes infinite. In addition, the variant (4.17a) of Algorithm I does *not* satisfy the conservation of energy in the sense of Theorem 4.3, i.e., simplicity can be further achieved at the expense of energy conservation.

4.2. Energy-conserving Algorithm II

We proceed to derive a second algorithm by applying the difference operators to the evolution equation (2.4b) at the mesh point (i, j) instead of at the point $(i, j+\frac{1}{2})$ as in Algorithm I. It will be seen that Algorithm II is somewhat simpler than Algorithm I, but along the same lines as Remark 4.4, simplification is achieved at the detriment of other properties.

4.2.1. Derivation and interpretation

In line with the final note in Remark 4.2, we derive Algorithm II directly from the local conservation law (2.4a), which at point (i, j) gives the following discrete version:

$$\mathfrak{C}_t^{(1)}[\tfrac{1}{2}(\mathfrak{C}_t^{(1)}u)^2 + \tfrac{1}{2}(\mathfrak{C}_x^{(1)}u)^2 + G(u)]_{i,j} - \mathfrak{C}_x^{(1)}[(\mathfrak{C}_x^{(1)}u)(\mathfrak{C}_t^{(1)}u)]_{i,j} = 0\,. \tag{4.18}$$

By applying the general Leibniz rule (3.18), the commutativity (3.12c), and the averaging rule (3.7), the first term in (4.18) becomes

$$\begin{aligned}\mathfrak{C}_t^{(1)}[\tfrac{1}{2}(\mathfrak{C}_t^{(1)}u)^2]_{i,j} &= [\mathfrak{C}_t^{(1)}(\mathfrak{C}_t^{(1)}u)_{i,j}][\mathfrak{A}_t(\mathfrak{C}_t^{(1)}u)_{i,j}] \\ &= (\mathfrak{C}_t^{(2)}u_{i,j})(\mathfrak{C}_t^{(1)}\mathfrak{A}_t u_{i,j}) = (\mathfrak{C}_t^{(2)}u_{i,j})(\mathfrak{C}_t^{(1)}u_{i,j})\,.\end{aligned} \tag{4.19a}$$

Similarly, for the other terms, we obtain

$$\mathfrak{C}_t^{(1)}[\tfrac{1}{2}(\mathfrak{C}_x^{(1)}u)^2]_{i,j} = (\mathfrak{C}_x^{(1)}\mathfrak{C}_t^{(1)}u_{i,j})(\mathfrak{C}_x^{(1)}u_{i,j})\,, \tag{4.19b}$$

$$\begin{aligned}\mathfrak{C}_t^{(1)}G(u)_{i,j} &= \frac{1}{k}\,[G(u_{i,j+1/2}) - G(u_{i,j-1/2})] = \frac{1}{k}\,[G(\mathfrak{A}_t u_{i,j+1/2}) - G(\mathfrak{A}_t u_{i,j-1/2})]\,, \\ &= \frac{1}{k}\,[G(\tfrac{1}{2}\{u_{i,j+1} + u_{i,j}\}) - G(\tfrac{1}{2}\{u_{i,j} + u_{i,j-1}\})]\,,\end{aligned} \tag{4.19c}$$

$$\mathfrak{C}_x^{(1)}(\mathfrak{C}_x^{(1)}u\mathfrak{C}_t^{(1)}u)_{i,j} = (\mathfrak{C}_x^{(2)}u_{i,j})(\mathfrak{C}_t^{1}u_{i,j}) + (\mathfrak{C}_x^{(1)}\mathfrak{C}_t^{(1)}u_{i,j})(\mathfrak{C}_x^{(1)}u_{i,j})\,. \tag{4.19d}$$

Also note that

$$\mathfrak{C}_t^{(1)}u_{i,j} = \frac{1}{k}\,[u_{i,j+1/2} - u_{i,j-1/2}] = \frac{1}{k}\,[\mathfrak{A}_t u_{i,j+1/2} - \mathfrak{A}_t u_{i,j-1/2}] = \frac{1}{2k}\,(u_{i,j+1} - u_{i,j-1})\,. \tag{4.19e}$$

A substitution of (4.19a–d) into (4.18), together with the use of (4.19e), leads to Algorithm II.

THEOREM 4.5 (Algorithm II). By applying the averaging rule (3.7), and the central difference rule (3.8) to approximate $\partial/\partial t$ and $\partial/\partial x$ at the mesh point (i, j), Algorithm II is obtained from the local conservation law (2.4a),

$$\mathfrak{C}_t^{(2)}u_{i,j} - \mathfrak{C}_x^{(2)}u_{i,j} + \frac{G(\tfrac{1}{2}\{u_{i,j+1} + u_{i,j}\}) - G(\tfrac{1}{2}\{u_{i,j} + u_{i,j-1}\})}{\tfrac{1}{2}(u_{i,j+1} - u_{i,j-1})} = 0\,. \tag{4.20}$$

A comparison between (2.1a), (4.5d) and (4.20) reveals the differences between Algorithm I and Algorithm II in the finite difference approximation of the last two terms. In particular, $G'(u_{i,j})$ is approximated by

$$G'(u_{i,j}) \approx \frac{G(\frac{1}{2}\{u_{i,j+1} + u_{i,j}\}) - G(\frac{1}{2}\{u_{i,j} + u_{i,j-1}\})}{\frac{1}{2}(u_{i,j+1} - u_{i,j-1})} . \tag{4.21}$$

Also note that by the nonlinearity of $G(\cdot)$ we have that $G(\mathfrak{A}_t u_{i,j\pm 1/2}) \neq \mathfrak{A}_t G(u_{i,j\pm 1/2})$. A geometric interpretation of Algorithm II is given in Fig. 2.

4.2.2. Accuracy

In view of the second order truncation error in (4.6b) and (4.6c) for the first two terms in (4.20), we only need to consider the truncation in the last term. Since

$$\frac{1}{2}\{u_{i,j\pm 1} + u_{i,j}\} = u_{i,j} \pm \frac{k}{2} u_{i,j}^{(0,1)} + \frac{k^2}{4} u_{i,j}^{(0,2)} + \text{hot} , \tag{4.22a}$$

it follows that

$$G(\tfrac{1}{2}\{u_{i,j\pm 1} + u_{i,j}\}) = G(u_{i,j}) + G'(u_{i,j})\left\{ \pm \frac{k}{2} u_{i,j}^{(0,1)} + \frac{k^2}{4} u_{i,j}^{(0,2)} + \text{hot}\right\} + \frac{1}{2}G''(u_{i,j})\left\{ \pm \frac{k}{2} u_{i,j}^{(0,1)} + \frac{k^2}{4} u_{i,j}^{(0,2)} + \text{hot}\right\}^2 + \text{hot} , \tag{4.22b}$$

and thus

$$G(\tfrac{1}{2}\{u_{i,j+1} + u_{i,j}\}) - G(\tfrac{1}{2}\{u_{i,j} + u_{i,j-1}\}) = kG'(u_{i,j})u_{i,j}^{(0,1)} + O(k^3) . \tag{4.22c}$$

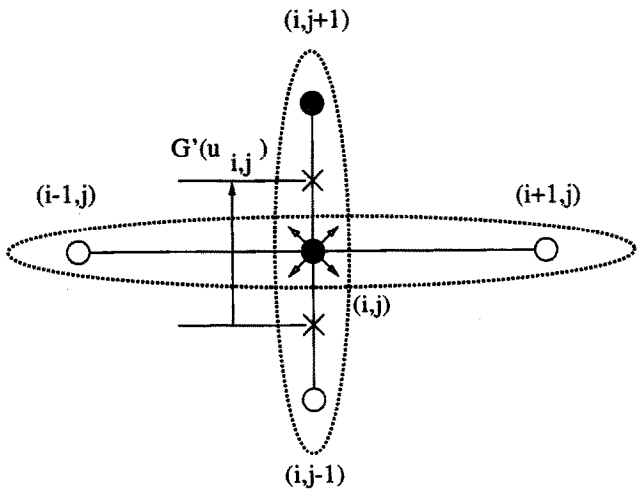

Fig. 2. *Geometry of Algorithm II.* The algorithm is constructed from point (i, j) indicated by the arrowed cross. Taylor series are expanded about the black mesh point (i, j). Only one unknown represented by the gray mesh point at time $t = (j + 1)k$. The two second order central differences linking three mesh points each are indicated by the two dotted-line ellipses: the vertical one in time, and the horizontal one in space. The difference quotient in (4.21) approximating $G'(u_{i,j})$ is represented by the vertical arrow.

On the other hand, it follows from (4.6a) that

$$\tfrac{1}{2}(u_{i,j+1} - u_{i,j-1}) = k u_{i,j}^{(0,1)} + \mathrm{O}(k^3)\,. \tag{4.22d}$$

Then the truncation error in the last term of (4.20) is also of second order,

$$\frac{G(\tfrac{1}{2}\{u_{i,j+1} + u_{i,j}\}) - G(\tfrac{1}{2}\{u_{i,j} + u_{i,j-1}\})}{\tfrac{1}{2}(u_{i,j+1} - u_{i,j-1})} = G'(u_{i,j}) + \mathrm{O}(k^2)\,. \tag{4.22e}$$

THEOREM 4.6 (Order of accuracy). Algorithm II is second order accurate in both space and time.

4.2.3. Energy conservation, boundary conditions, stability

As in Algorithm I, a discrete energy is conserved. Reconsider (4.18) under a different angle:

$$\mathfrak{C}_t^{(1)} \mathscr{E}_{i,j} - \mathfrak{C}_x^{(1)} (\mathfrak{C}_x^{(1)} u \mathfrak{C}_t^{(1)} u)_{i,j} = 0\,, \tag{4.23a}$$

$$\Rightarrow \quad \frac{h}{k}\,[\mathscr{E}_{i,j+1/2} - \mathscr{E}_{i,j-1/2}] - h[(\mathfrak{C}_x^{(1)} u \mathfrak{C}_t^{(1)} u)_{i+1/2,j} - (\mathfrak{C}_x^{(1)} u \mathfrak{C}_t^{(1)} u)_{i-1/2,j}] = 0\,, \tag{4.23b}$$

with

$$\mathscr{E}_{i,j\pm 1/2} = \tfrac{1}{2}(\mathfrak{C}_t^{(1)} u_{i,j\pm 1/2})^2 + \tfrac{1}{2}(\mathfrak{C}_x^{(1)} u_{i,j\pm 1/2})^2 + G(u_{i,j\pm 1/2})\,. \tag{4.23c}$$

The results of applying (3.3) and (3.4) to each term of (4.23c), for the case $(j + \frac{1}{2})$, are recorded below:

$$\mathfrak{C}_t^{(1)} u_{i,j+1/2} = \frac{1}{k}\,[u_{i,j+1} - u_{i,j}]\,, \tag{4.24a}$$

$$\mathfrak{C}_x^{(1)} u_{i,j+1/2} = \frac{1}{4h}\,[(u_{i+1,j} + u_{i+1,j+1}) - (u_{i-1,j} + u_{i-1,j+1})]\,, \tag{4.24b}$$

$$G(u_{i,j+1/2}) = G(\tfrac{1}{2}\{u_{i,j+1} + u_{i,j}\})\,, \tag{4.24c}$$

which together yield the following expression for $\mathscr{E}_{i,j+1/2}$:

$$\mathscr{E}_{i,j+1/2} = \frac{1}{2k^2}\,[u_{i,j+1} - u_{i,j}]^2 + \frac{1}{16h^2}\,[(u_{i+1,j} + u_{i+1,j+1}) - (u_{i-1,j} + u_{i-1,j+1})]^2 + G(\tfrac{1}{2}\{u_{i,j+1} + u_{i,j}\})\,. \tag{4.25a}$$

The expression for $\mathscr{E}_{i,j-1/2}$ can be obtained from (4.25a). Next, upon defining the discrete energy for Algorithm II by

$$E_{N,j\pm 1/2}^{(\mathrm{II})} := h \sum_{i=-N}^{N} \mathscr{E}_{i,j\pm 1/2}\,, \tag{4.25b}$$

summing (4.23b) from $i = -N$ to $i = N$, we arrive at

$$\frac{1}{k}\left[E^{(\mathrm{II})}_{N,j+1/2} - E^{(\mathrm{II})}_{N,j-1/2}\right] - \left[(\mathfrak{G}^{(1)}_x u \mathfrak{G}^{(1)}_t u)_{N+1/2,j} - (\mathfrak{G}^{(1)}_x u \mathfrak{G}^{(1)}_t u)_{-N-1/2,j}\right] = 0\,, \tag{4.26}$$

which leads to the following.

THEOREM 4.7 (Energy conservation). Algorithm II described in (4.20) conserves the discrete energy $E^{(\mathrm{II})}_{N,\cdot}$ defined in (4.25) in the sense that

$$E^{(\mathrm{II})}_{N,j+1/2} = E^{(\mathrm{II})}_{N,j-1/2} \quad \forall j\,, \tag{4.27a}$$

provided that the boundary conditions are chosen to be symmetric such that

$$(\mathfrak{G}^{(1)}_x u \mathfrak{G}^{(1)}_t u)_{N+1/2,j} = (\mathfrak{G}^{(1)}_x u \mathfrak{G}^{(1)}_t u)_{-N-1/2,j}\,. \tag{4.27b}$$

As with the symmetry condition (4.10b), one can count on three ways, with different degrees of restriction, to satisfy the symmetry condition (4.27b)[17], which consist of eliminating appropriate linear constraint equations from the resulting system of algebraic equations. The *first way* is to set

$$\mathfrak{G}^{(1)}_x u_{N+1/2,j} = \mathfrak{G}^{(1)}_x u_{-N-1/2,j} \equiv 0 \quad \forall j\,, \tag{4.28a}$$

by enforcing the boundary conditions

$$u_{N+1,j} = u_{N,j}\,, \quad u_{-N,j} = u_{-N-1,j} \quad \forall j\,. \tag{4.28b}$$

The *second way* is to set

$$\mathfrak{G}^{(1)}_t u_{N+1/2,j} = \mathfrak{G}^{(1)}_t u_{-N-1/2,j} \equiv 0 \quad \forall j\,, \tag{4.29a}$$

by enforcing the boundary conditions

$$\begin{aligned} &u_{N+1,j+1} + u_{N,j+1} = u_{N+1,j-1} + u_{N,j-1}\,,\\ &u_{-N,j+1} + u_{-N-1,j+1} = u_{-N,j-1} + u_{-N-1,j-1}\,. \end{aligned} \tag{4.29b}$$

Finally, a *third* and *least restrictive* way is to set

$$\mathfrak{G}^{(1)}_x u_{N+1/2,j} = \mathfrak{G}^{(1)}_x u_{-N-1/2,j} \neq 0 \quad \forall j\,, \qquad \mathfrak{G}^{(1)}_t u_{N+1/2,j} = \mathfrak{G}^{(1)}_t u_{-N-1/2,j} \neq 0 \quad \forall j\,, \tag{4.30a}$$

by enforcing the boundary conditions

$$\begin{aligned} &(u_{N+1,j} - u_{N,j}) = (u_{-N,j} - u_{-N-1,j}) \quad \forall j\,,\\ &(u_{N+1,j+1} + u_{N,j+1}) - (u_{-N,j+1} + u_{-N-1,j+1})\\ &\quad = (u_{N+1,j-1} + u_{N,j-1}) - (u_{-N,j-1} + u_{-N-1,j-1}) \quad \forall j\,. \end{aligned} \tag{4.30b}$$

[17] Equation (4.27b) is by itself a *nonlinear* constraint equation.

Suppose that all values of $u_{\cdot,j}$ at the time level j had been computed; thus we must have had $(4.30b)_1$ satisfied already. Since $(4.30b)_1$ has to be satisfied for all j, at the (current) time level $(j+1)$ when all unknowns $u_{\cdot,j+1}$ are to be computed, we need to enforce $(4.30b)_1$ at time level $(j+1)$ (and not at time level j, which it is already satisfied) together with $(4.30b)_2$. Next, we examine the stability of Algorithm II and its variants for the ULKGE (4.15).

THEOREM 4.8 (Linear stability). Algorithm II as described in (4.20) is conditionally stable for the ULKGE (4.15), with a stability limit for $\lambda := k/h$ independent of ϵ:

$$\lambda \leqslant 1 . \tag{4.31}$$

PROOF. Substitution of (4.16c,d,e) into (4.20) yields

$$(g^{1/2} - g^{-1/2})^2 = -\frac{4\lambda^2 \sin^2 \frac{\theta}{2} + \epsilon k^2}{1 + \frac{1}{4}\epsilon k^2} =: -\alpha . \tag{4.32a}$$

As already mentioned in the proof of Theorem (4.4), the stability condition requires that

$$0 \leqslant \alpha \leqslant 4 . \tag{4.32b}$$

The first inequality is satisfied since $\epsilon \geqslant 0$, whereas the satisfaction of the second inequality leads to (4.31). □

REMARK 4.5. Variant 1 of Algorithm II. A first variant of (4.20) is to replace the approximation (4.21) to $G'(u_{i,j})$ by $G'(u_{i,j})$ itself:

$$\mathfrak{S}_t^{(2)} u_{i,j} - \mathfrak{S}_x^{(2)} u_{i,j} + G'(u_{i,j}) = 0 . \tag{4.33a}$$

Algorithm (4.33a), also known as the leap-frog algorithm, is second order accurate by using (4.6b,c) and (4.6g). For the ULKGE (4.15) with $G'(u_{i,j}) = \epsilon u_{i,j}$, a Von Neumann analysis leads to

$$(g^{1/2} - g^{-1/2})^2 = -\left(4\lambda^2 \sin^2 \frac{\theta}{2} + \epsilon k^2\right) =: -\alpha , \tag{4.33b}$$

which for $0 \leqslant \alpha \leqslant 4$ requires that

$$\frac{k}{h} \leqslant \frac{1}{\sqrt{1 + \frac{1}{4}\epsilon h^2}} . \tag{4.33c}$$

The stability limit depends on ϵ and h. In particular, it can be seen that λ is strictly less than 1, which corroborates well with the numerical experiments reported in [19, p. 588]: When applying algorithm (4.33a) to the sine-Gordon equation, it was found that the algorithm was unstable for $k = h$, but stable for $k = 0.95h$.[18] Algorithm (4.33a) does not conserve energy in

[18] We did not find the theoretical justification for this result as given in (4.33c) in any of the references that we looked into.

the sense of Theorem 4.7. We refer to Remark 4.4 for a similar variant of Algorithm I, which has a much less restrictive stability limit; compare (4.17c) and (4.33c).

REMARK 4.6. Variant 2 of Algorithm II. The algorithm proposed by Strauss and Vazquez [30] (see also [19, p. 589]) can be thought of as a variant of Algorithm II with an approximation for $G'(u_{i,j})$ different than that of (4.21),

$$\mathfrak{C}_t^{(2)} u_{i,j} - \mathfrak{C}_x^{(2)} u_{i,j} + \frac{G(u_{i,j+1}) - G(u_{i,j-1})}{u_{i,j+1} - u_{i,j-1}} = 0 . \tag{4.34a}$$

In view of (4.6d) and (4.22d), the above approximation to $G'(u_{i,j})$ is also of second order, similar to (4.22e). We obtain the following stability limit of this algorithm for the ULKGE

$$(g^{1/2} - g^{-1/2})^2 = -\frac{4\lambda^2 \sin^2 \frac{\theta}{2} + \epsilon k^2}{1 + \frac{1}{2}\epsilon k^2} =: -\alpha . \tag{4.34b}$$

There are two cases to be considered:

(1) $h \leq 2/\sqrt{\epsilon}$. The stability condition is given as follows:

$$0 \leq \alpha \leq 4 \quad \Rightarrow \quad \lambda := \frac{k}{h} \leq \frac{1}{\sqrt{1 - \frac{1}{4}\epsilon h^2}} \leq \sqrt{2} . \tag{4.34c}$$

(2) $h > 2/\sqrt{\epsilon}$. Clearly, $0 \leq \alpha$. The inequality $\alpha \leq 4$ leads to

$$\frac{k^2}{h^2}\left(\sin^2 \frac{\theta}{2} - \frac{\epsilon h^2}{4}\right) \leq 1 , \tag{4.34d}$$

which is always true since $\sin^2 \frac{1}{2}\theta - \frac{1}{4}\epsilon h^2 \leq 0$. Hence the stability condition $0 \leq \alpha \leq 4$ is satisfied in this case without additional requirement on λ; in other words, the algorithm is always stable for $h > 2/\sqrt{\epsilon}$ and for any k.

The above stability condition is somewhat less restrictive than in the standard wave equation ($\epsilon = 0$). The essential difference between Algorithm II and its variant (4.34a) is that the finite difference discretization in (4.34a) is *not* entirely consistent in the sense that if we replace $G'(u_{i,j})$ by $\mathfrak{C}_u^{(1)} G(u_{i,j})$, we have that

$$\frac{G(u_{i,j+1}) - G(u_{i,j-1})}{u_{i,j+1} - u_{i,j-1}} \neq \mathfrak{C}_u^{(1)} G(u_{i,j}) . \tag{4.35}$$

On the other hand, if we define the central difference operator differently:

$$\Delta^{(1)} f_i := \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}} ,$$

then it is obvious that $\Delta^{(2)} \neq \mathfrak{C}^{(2)}$.

Although one can still manage to obtain some discrete energy as in [30], the relationship between the total system energy and boundary conditions remain obscure. Moreover, the positiveness of the discrete energy proposed in [30] is also unclear. In contrast, the consistency in our formalism leads to discrete algebraic operators that possess algebraic invariants with clear interpretation.

REMARK 4.7. Non-conservative models. To the NLKGE (2.1a), one can add dissipation, damping, and a driving term to form a model that describes the physical behavior of Josephson junctions [31, 39]

$$U_{tt} - U_{xx} + G'(U) + aU_t - bU_{xxt} + c = 0\,. \tag{4.36a}$$

A reliable algorithm in this case should not introduce numerical dissipation, damping, and amplification such that the numerical results deviate to any significant amount from these effects in the physical model. Clearly, such algorithm must possess the properties just cited in the case where $a = b = c = 0$. This observation then suggests that Algorithms I and II can be employed as starting points to construct algorithms for (4.36a). Thus to the right-hand side of (4.5d), or equivalently to the right-hand side of $(4.1)_2$, for Algorithm I, and to the right-hand side of (4.20) for Algorithm II, we add the three terms

$$a\mathfrak{S}_t^{(1)}u_{i,j} - b\mathfrak{S}_x^{(2)}\mathfrak{S}_t^{(1)}u_{i,j} + c\,, \tag{4.36b}$$

with

$$\mathfrak{S}_x^{(2)}\mathfrak{S}_t^{(1)}u_{i,j} = \frac{1}{2h^2k}\left\{(u_{i+1,j+1} - 2u_{i,j+1} + u_{i-1,j+1}) - (u_{i+1,j-1} - 2u_{i,j-1} + u_{i-1,j-1})\right\}, \tag{4.36c}$$

which corresponds to the last three terms in (4.36a) to form two algorithms for (4.36a).

4.3. Momentum-conserving algorithm

4.3.1. Derivation

Instead of starting from the energy conservation law (2.4), one can begin with the conservation of linear momentum (2.8) to derive a desired algorithm possessing a momentum invariant. Applying the same formalism for algorithm derivation as presented above in Section 4.2.1 to the conservation of linear momentum (2.8a) at the mesh point (i, j), we obtain

$$\mathfrak{S}_t^{(2)}u_{i,j} - \mathfrak{S}_x^{(2)}u_{i,j} + \frac{G(\frac{1}{2}\{u_{i+1,j} + u_{i,j}\}) - G(\frac{1}{2}\{u_{i,j} + u_{i-1,j}\})}{\frac{1}{2}(u_{i+1,j} - u_{i-1,j})} = 0\,, \tag{4.37}$$

which is similar to (4.20) except for the approximation of $G'(u_{i,j})$.[19] We refer to (4.37) as Algorithm III, with a geometric interpretation given in Fig. 3.

THEOREM 4.9 (Algorithm III). (i) Algorithm III is consistent with the NLKGE (2.1a), and is also second order accurate, i.e., its truncation error is $O(h^2, k^2)$. *(ii) Algorithm III has an*

[19] A justification of the derivation of (4.37) will be given shortly in the proof of Theorem 4.9.

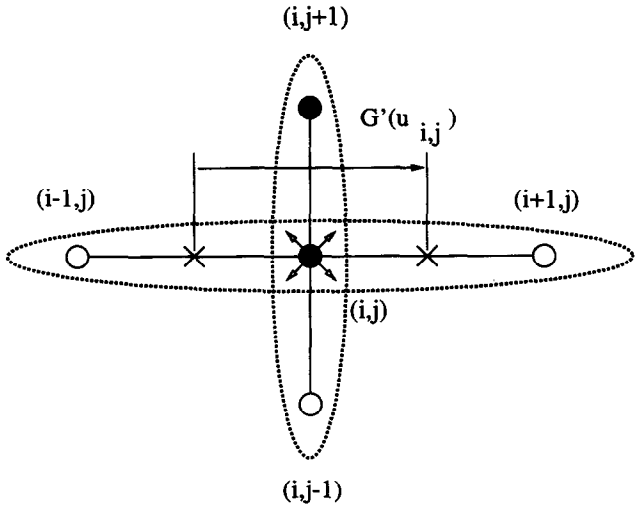

Fig. 3. *Geometry of Algorithm III.* Similar to Algorithm II (Fig. 2), except for the difference quotient (the last term) in (4.37) approximating $G'(u_{i,j})$, which is now represented by the horizontal arrow.

algebraic momentum invariant. (*iii*) *For the ULKGE* (*4.15*), *Algorithm III is conditionally stable, with the criterion*

$$\lambda = \frac{k}{h} \leqslant 1 .$$

PROOF. (i) By (4.6b) and (4.6c), one has that

$$\mathfrak{C}_t^{(2)} U_{i,j} = U_{i,j}^{(0,2)} + \mathrm{O}(k^2) , \qquad \mathfrak{C}_x^{(2)} U_{i,j} = U_{i,j}^{(2,0)} + \mathrm{O}(h^2) . \tag{4.39a}$$

Similar to (4.22e), it is easy to verify that

$$\frac{G(\frac{1}{2}\{U_{i+1,j} + U_{i,j}\}) - G(\frac{1}{2}\{U_{i,j} + U_{i-1,j}\})}{\frac{1}{2}(U_{i+1,j} - U_{i-1,j})} = G'(U_{i,j}) + \mathrm{O}(h^2) . \tag{4.39b}$$

Applying the averaging rule, one obtains

$$U_{i+1,j} - U_{i-1,j} = 2(U_{i+1/2,j} - U_{i-1/2,j}) .$$

Then the use of (4.39a–c) in the definition

$$L_{i,j}^{(\mathrm{III})}(U) := \mathfrak{C}_t^{(2)} U_{i,j} - \mathfrak{C}_x^{(2)} U_{i,j} + \mathfrak{C}_U^{(1)} G(U_{i,j})|_{j=\text{fixed}} , \tag{4.39c}$$

leads to the second order accuracy of Algorithm III, i.e.,

$$\| L_{i,j}^{(\mathrm{III})}(U) \| \sim \mathrm{O}(h^2, k^2) . \tag{4.40}$$

(ii) It is only necessary to show that (4.37) is a finite difference description of the momentum conservation law (2.8a) using the permissible algebraic operators developed in Section 3. Multiply (4.37) by $\mathfrak{E}_x u_{i,j}$:

$$\mathfrak{E}_x u_{i,j}\{\mathfrak{E}_t^{(1)} v_{i,j} - \mathfrak{E}_x^{(2)} u_{i,j} + \mathfrak{E}_u^{(1)} G(u_{i,j})|_{j=\text{fixed}} = 0\} . \tag{4.41a}$$

By the Leibniz rule (3.20), the second term in (4.41a) turns out to be

$$\mathfrak{E}_x^{(1)} u_{i,j} \mathfrak{E}_x^{(2)} u_{i,j} = \mathfrak{E}_x^{(1)}[\tfrac{1}{2}(\mathfrak{E}_x^{(1)} u_{i,j})^2] . \tag{4.41b}$$

Also by combining (3.20) and (3.11), the first term of (4.41a) can be written as

$$\mathfrak{E}_x^{(1)} u_{i,j} \mathfrak{E}_t^{(1)} v_{i,j} = \mathfrak{E}_t^{(1)}(v_{i,j} \mathfrak{E}_x^{(1)} u_{i,j}) - \tfrac{1}{2}\mathfrak{E}_x^{(1)}(v_{i,j})^2 . \tag{4.41c}$$

Finally, by the chain rule (3.21), the third term of (4.41a) is written as

$$\mathfrak{E}_x^{(1)} u_{i,j} \mathfrak{E}_u^{(1)} G(u_{i,j})|_{j=\text{fixed}} = \mathfrak{E}_x^{(1)} G(u_{i,j}) . \tag{4.41d}$$

A substitution of (4.41b,c,d) in (4.41a) leads to

$$\mathfrak{E}_t^{(1)}(v_{i,j} \mathfrak{E}_x^{(1)} u_{i,j}) - \mathfrak{E}_x^{(1)}[\tfrac{1}{2}(v_{i,j})^2 + \tfrac{1}{2}(\mathfrak{E}_x^{(1)} u_{i,j})^2 - G(u_{i,j})] = 0 , \tag{4.42}$$

which is exactly the finite difference discrete version of the continuous local momentum conservation law (2.8a).

(iii) Let $G(u) = \frac{1}{2}\epsilon u^2$; then (4.37) becomes

$$(u_{i,j+1} - 2u_{i,j} + u_{i,j-1}) - \lambda(u_{i+1,j} - 2u_{i,j} + u_{i,j}) + \tfrac{1}{4}\epsilon k^2(u_{i+1,j} + 2u_{i,j} + u_{i-1,j}) = 0 . \tag{4.43}$$

Substituting $u_{i,j} = \mathrm{e}^{\mathrm{i}\theta\sqrt{-1}} g^j$ into (4.43), one obtains

$$(g^{1/2} - g^{-1/2})^2 = -\left[4\lambda \sin^2 \frac{\theta}{2} + \epsilon k^2 \cos^2 \frac{\theta}{2}\right] =: -\alpha(\theta) . \tag{4.44a}$$

By Theorem 4.4, the stability condition reads

$$0 \leqslant \alpha(\theta) \leqslant 4 . \tag{4.44b}$$

The first inequality is automatically satisfied. For the second inequality, since

$$\alpha(\theta) = 4\lambda^2 \sin^2 \tfrac{1}{2}\theta + \epsilon k^2 \cos^2 \tfrac{1}{2}\theta = \tfrac{1}{2}(4\lambda^2 + \epsilon k^2) + (\epsilon k^2 - 4\lambda^2)\cos\theta , \tag{4.45a}$$

it follows that

$$\max\{\alpha(\theta)\} = \tfrac{1}{2}(4\lambda^2 + \epsilon k^2) + \tfrac{1}{2}|4\lambda^2 - \epsilon k^2| . \tag{4.45b}$$

Two cases need to be examined:

(iii.a) $\epsilon k^2 \leq 4\lambda^2$, i.e., $h \leq 2/\sqrt{\epsilon}$. Then $\max\{\alpha(\theta)\} = 4\lambda^2$, which leads to the stability criterion

$$\lambda \leq 1 . \tag{4.45c}$$

(iii.b) $4\lambda^2 \leq \epsilon k^2$, i.e., $2/\sqrt{\epsilon} \leq h$. Then $\max\{\alpha(\theta)\} = \epsilon k^2$, which also leads to

$$4\lambda^2 \leq \epsilon k^2 \leq 4 \quad \Rightarrow \quad \lambda \leq 1 . \tag{4.45d}$$

Thus the stability criterion $\lambda \leq 1$ is valid without any restriction on h. □

REMARK 4.8. A variant of Algorithm III. The following modified leap-frog algorithm, a generalization of the algorithm by Ablowitz et al. [24], can be thought of as a variant of Algorithm III:

$$\mathfrak{S}_t^{(2)} u_{i,j} - \mathfrak{S}_x^{(2)} u_{i,j} + G'(\tfrac{1}{2}\{u_{i-1,j} + u_{i+1,j}\}) = 0 . \tag{4.46a}$$

The only difference between (4.46a) and (4.37) is in the last term, which still retains some similarity. Note that the algorithm by Ablowitz et al. [24] is recovered from (4.46a) by setting $k = h$. Algorithm (4.46a) is second order accurate by a proof similar to that of Theorem 4.6. We now compare the stability properties between algorithm (4.46a) and Algorithm III for the ULKGE (4.15).

The counterpart of (4.44a) for algorithm (4.46a) is

$$(g^{1/2} - g^{-1/2})^2 = -(4\lambda^2 \sin^2 \tfrac{1}{2}\theta + \epsilon k^2 \cos\theta) = 2\lambda^2 + (\epsilon k^2 - 2\lambda^2)\cos\theta =: -\alpha(\theta) . \tag{4.46b}$$

Recall that by Theorem 4.4, the stability condition is $0 \leq \alpha(\theta) \leq 4$. From (4.46b), we have

$$\max\{\alpha(\theta)\} = 2\lambda^2 + |\epsilon k^2 - 2\lambda^2| , \qquad \min\{\alpha(\theta)\} = 2\lambda^2 - |\epsilon k^2 - 2\lambda^2| . \tag{4.46c}$$

In the present situation, the first inequality $0 \leq \alpha$ is no longer always valid. There are three different cases:

(1) $\epsilon k^2 < 2\lambda^2$, i.e., $h \leq \sqrt{2/\epsilon}$. In this case,

$$\max\{\alpha(\theta)\} = 4\lambda^2 - \epsilon k^2 , \qquad \min\{\alpha(\theta)\} = -\epsilon k^2 , \tag{4.46d}$$

leading to the stability criterion

$$0 \leq \alpha \leq 4 \quad \Rightarrow \quad \lambda := \frac{k}{h} \leq \frac{1}{\sqrt{1 - \frac{1}{4}\epsilon h^2}} \leq \sqrt{2} .$$

(2) $2\lambda^2 \leq \epsilon k^2 \leq 4\lambda^2$, i.e., $\sqrt{2/\epsilon} \leq h \leq 2/\sqrt{\epsilon}$. In this case,

$$\max\{\alpha(\theta)\} = \epsilon k^2\,, \qquad \min\{\alpha(\theta)\} = 4\lambda^2 - \epsilon k^2 < 0\,, \tag{4.46e}$$

thereby

$$2\lambda^2 \leqslant \epsilon k^2 \leqslant 4 \quad \Rightarrow \quad \lambda \leqslant \frac{1}{\sqrt{2}}\,.$$

(3) $4\lambda^2 < \epsilon k^2$, i.e., $2/\sqrt{\epsilon} < h$. In this case,

$$\max\{\alpha(\theta)\} = \epsilon k^2\,, \qquad \min\{\alpha(\theta)\} = 4\lambda^2 - \epsilon k^2 < 0\,. \tag{4.46f}$$

Hence

$$\text{if } \epsilon = 0 \text{ (i.e., } h \to 0) \quad \Rightarrow \quad \lambda = 0\,,$$

$$\text{if } \epsilon \neq 0 \text{ (i.e., } h \text{ finite}) \quad \Rightarrow \quad \alpha < 0 \text{ is possible,}$$

which mean that the algorithm is unstable in the present case.

The main observation from the above three cases is that for algorithm (4.46a) the stability criterion on λ depends on the value of h: The finer the spatial discretization (h small) the larger the stability limit on λ.[20] As far as stability is concerned, Algorithm III is more robust in the sense that the stability criterion does not depend on the amplitude of h as shown in (4.45c,d). Note also that algorithm (4.46a) does not conserve momentum in the sense defined in Theorem 4.9. Christiansen and Lomdahl [40] employed a generalization of (4.46a) to the 2-D case; from the above analysis, a superior algorithm could be obtained by generalizing Algorithm III to the 2-D case.

4.3.2. Momentum conservation, boundary conditions

As pointed out earlier in Section 2.2, with the appropriate boundary conditions (2.10b), the momentum integral (2.11) exists. We now demonstrate that Algorithm III can preserve the momentum integral numerically, with a finite difference version of the boundary conditions (2.10b). First, we define

$$\mathcal{M}_{i,j} := v_{i,j}\mathfrak{F}_x^{(1)}u_{i,j} = \frac{1}{k}\,(u_{i,j+1/2} - u_{i,j-1/2})\,\frac{1}{h}\,(u_{i+1/2,j} - u_{i-1/2,j}) \tag{4.47a}$$

and

$$\mathcal{J}_{i,j} := [\tfrac{1}{2}(v_{i,j})^2 + \tfrac{1}{2}(\mathfrak{F}_x^{(1)}u_{i+1/2,j})^2 - G(u_{i,j})]\,. \tag{4.47b}$$

Then (4.42) can be recast as

$$\mathfrak{F}_t^{(1)}\mathcal{M}_{i,j} - \mathfrak{F}_x^{(1)}\mathcal{J}_{i,j} = 0 \tag{4.48a}$$

or

$$\frac{1}{k}\,[\mathcal{M}_{i,j+1/2} - \mathcal{M}_{i,j-1/2}] - \frac{1}{h}\,[\mathcal{J}_{i+1/2,j} - \mathcal{J}_{i-1/2,j}] = 0\,. \tag{4.48b}$$

Summing the index i in (4.48b) from $-N$ to N, we obtain

$$h\sum_{i=-N}^{N}(\mathcal{M}_{i,j+1/2} - \mathcal{M}_{i,j-1/2}) = k\{-\mathcal{J}_{-N-1/2,j} + \mathcal{J}_{N+1/2,j}\}\,. \tag{4.49a}$$

[20] But we only need to have $h \leqslant \sqrt{2/\epsilon}$ to achieve the maximum upper bound on λ for algorithm (4.46a). Thus the maximum time step size k would be $k = 2/\sqrt{\epsilon}$, with $h = \sqrt{2/\epsilon}$.

Denote

$$M^{(\mathrm{III})}_{i,j\pm 1/2} := h \sum_{i=-N}^{N} \mathcal{M}_{i,j\pm 1/2} .$$

Then (4.49a) is read as

$$M^{(\mathrm{III})}_{N,j+1/2} - M^{(\mathrm{III})}_{N,j-1/2} = k\{\mathcal{J}_{i+1/2,j}\}\Big|_{i=-N-1}^{i=N} , \tag{4.49b}$$

which leads to the following.

THEOREM 4.10 (Momentum conservation). Algorithm III described in (4.37) conserves the discrete momentum $M^{(\mathrm{III})}_{N,j}$ defined in (4.49a) in the sense that

$$M^{(\mathrm{III})}_{N,j+1/2} = M^{(\mathrm{III})}_{N,j-1/2} \quad \forall j , \tag{4.50a}$$

provided that the following boundary condition holds:

$$\mathcal{J}_{-N-1/2} = \mathcal{J}_{N+1/2,j} \quad \forall j . \tag{4.50b}$$

Let us have a close look at the expression for $\mathcal{J}_{\pm(N+1/2),j}$,

$$\mathcal{J}_{\pm(N+1/2),j} = \left[\frac{1}{2}\left(\frac{u_{\pm(N+1/2),j+1} - u_{\pm(N+1/2),j-1}}{2k} \right)^2 + \frac{1}{2}\left(\frac{u_{\pm(N+1),j} - u_{\pm(N-1),j}}{h} \right)^2 - G\left(\tfrac{1}{2}\{u_{\pm(N+1),j} + u_{\pm(N-1),j}\}\right) \right] . \tag{4.51}$$

Therefore, a general way to enforce the boundary condition (4.50b) is to set

$$(u_{-(N+1),j+1} + u_{-N,j+1}) - (u_{-(N+1),j-1} + u_{-N,j-1}) = (u_{(N+1),j+1} + u_{N,j+1}) - (u_{(N+1),j-1} + u_{N,j-1}) , \tag{4.52a}$$

$$u_{-(N+1),j} - u_{-N,j} = u_{(N+1),j} - u_{N,j} \quad \forall j , \tag{4.52b}$$

$$G\left(\tfrac{1}{2}\{u_{-(N+1),j} + u_{-N,j}\}\right) = G\left(\tfrac{1}{2}\{u_{N+1,j} + u_{N,j}\}\right) \quad \forall j . \tag{4.52c}$$

REMARK 4.11. Both Algorithm III and its variant (4.46a) are explicit algorithms in the sense that $u_{i,j+1}$ is obtained directly in terms of known solution at time stations j and $j-1$. Besides, (4.46a) has some other computational advantages (parallel computation) (see [24]).

5. Computer implementation

Based on the formalism introduced in the previous sections, several algorithms have been designed which can preserve certain invariants in the NLKGE. In the present section, we demonstrate how to implement these algorithms using the energy-conserving Algorithm I as a

prototype. Let the spatial coordinate line be discretized into $2N$ segments as follows:

$$\{-Nh, (-N+1)h, \ldots, -h, 0, h, \ldots, (N-1)h, Nh\} ,$$

with the integer N chosen such that

$$|u_{\pm Nh, t_{\max}} - u_{\pm(N+i)h, t_{\max}}| < \epsilon ,$$

where i is a given integer, $t_{\max}$ the maximum time up to which the computation will be performed, and ϵ a given tolerance. For convenience, we let

$$\mathcal{N} := \{-N, -N+1, \ldots, -1, 0, 1, \ldots, N-1, N\}$$

and[21]

$$\underline{u}_j := \lfloor u_{-N,j}, u_{-N+1,j}, \ldots, u_{i,j}, \ldots, u_{N-1,j}, u_{N,j} \rfloor \equiv \lfloor u_{i,j} \rfloor \in \mathbb{R}^{1\times(2N+1)} ,$$

$$\underline{v}_j := \lfloor v_{-N,j}, v_{-N+1,j}, \ldots, v_{i,j}, \ldots, v_{N-1,j}, v_{N,j} \rfloor \equiv \lfloor v_{i,j} \rfloor \in \mathbb{R}^{1\times(2N+1)} .$$

Our focus here is Algorithm I as described in (4.1). At time $t = (j+1)k$, it has $2(2N+1)$ unknowns, $\underline{u}_{j+1} = \lfloor u_{i,j+1} \rfloor \in \mathbb{R}^{1\times(2n+1)}$ and $\underline{v}_{j+1} = \lfloor v_{i,j+1} \rfloor \in \mathbb{R}^{1\times(2n+1)}$. One can rewrite (4.1a) as

$$v_{i,j+1} = \frac{2}{k}(u_{i,j+1} - u_{i,j}) - v_{i,j} , \tag{5.1}$$

which when substituted into the second equation of (4.1b) leads to an alternative form of Algorithm I:

$$v_{i,j+1} = \frac{2}{k}(u_{i,j+1} - u_{i,j}) - v_{i,j} , \tag{5.2a}$$

$$-4hrv_{i,j} - \{[u_{i+1,j+1} - (2+4r^2)u_{i,j+1} + u_{i-1,j+1}] + [u_{i+1,j} - (2-4r^2)u_{i,j} + u_{i-1,j}]\}$$
$$+ 2h^2 \frac{G(u_{i,j+1}) - G(u_{i,j})}{u_{i,j+1} - u_{i,j}} = 0 , \tag{5.2b}$$

where $r := 1/\lambda = h/k$. The advantage of (5.2) is that we decoupled $\underline{u}_{j+1}$ from $\underline{v}_{j+1}$ such that (5.2b) can be used to solve for the $(2N+1)$ unknowns in $\underline{u}_{j+1}$. Then (5.2a) is used to obtain v_{j+1}; this update procedure is discussed in Section 5.2.

5.1. *Consistent linearization*

Basically, (5.2b) with $i \in \mathcal{N}$ form a system of nonlinear algebraic equations for the unknowns in $\underline{u}_{j+1} \in \mathbb{R}^{1\times(2N+1)}$. These equations will be solved by the Newton–Raphson procedure; in what follows, we describe the necessary discrete tangent operators obtained via linearization.

[21] We employ the following notation $\underline{u}_j = \lfloor u_{i,j} \rfloor \in \mathbb{R}^{1\times(2N+1)}$, which means that the free index i is to be expanded to cover its range, i.e., $i \in \mathcal{N}$, to form a $1 \times (2N+1)$ row matrix.

We denote the solution for $\underline{u}_{j+1}$ and $\underline{v}_{j+1}$ at iteration (l) of the Newton–Raphson solution procedure by $\underline{u}^{(l)}_{j+1}$ and $\underline{v}^{(l)}_{j+1}$. Now, assume that $\underline{u}^{(l)}_{j+1}$ and $\underline{v}^{(l)}_{j+1}$ are known. Let $\underline{u}^{(l)}_{j+1,\epsilon}$ and $\underline{v}^{(l)}_{j+1,\epsilon}$ be the perturbation of $\underline{u}^{(l)}_{j+1}$ and $\underline{v}^{(l)}_{j+1}$ in the direction $\Delta\underline{u}^{(l)}_{j+1}$ and $\Delta\underline{v}^{(l)}_{j+1}$ defined as follows:

$$\underline{u}^{(l)}_{j+1,\epsilon} := \underline{u}^{(l)}_{j+1} + \epsilon\Delta\underline{u}^{(l)}_{j+1}\,, \qquad \underline{v}^{(l)}_{j+1,\epsilon} := \underline{v}^{(l)}_{j+1} + \epsilon\Delta\underline{v}^{(l)}_{j+1}\,. \tag{5.3}$$

Also for $i \in \mathcal{N}$, we define the residuals $p^{(l)}_{i,j+1}$ and $q^{(l)}_{i,j+1}$ corresponding to (5.2a) and (5.2b), respectively, as[22]

$$p^{(l)}_{i,j+1} := \frac{u^{(l)}_{i,j+1} - u_{i,j}}{k} - \frac{v^{(l)}_{i,j+1} + v_{i,j}}{2}\,, \tag{5.4a}$$

$$\begin{aligned} q^{(l)}_{i,j+1} := {} & -4rhv_{i,j} - \{(u^{(l)}_{i+1,j+1} - (2+4r^2)u^{(l)}_{i,j+1} + u^{(l)}_{i-1,j+1}) \\ & + (u_{i+1,j} - (2-4r^2)u_{i,j} + u_{i-1,j})\} + 2h^2\,\frac{G(u^{(l)}_{i,j+1}) - G(u_{i,j})}{u^{(l)}_{i,j+1} - u_{i,j}}\,. \end{aligned} \tag{5.4b}$$

From the definition of $p^{(l)}_{i,j+1}$ and $q^{(l)}_{i,j+1}$ in (5.4a,b), one can readily introduce the definition of the perturbation $p^{(l)}_{i,j+1,\epsilon}$ and $q^{(l)}_{i,j+1,\epsilon}$ based on (5.3). Thus the linearization of the system (5.4a,b) about the known solution $\underline{u}^{(l)}_{j+1}$ and $\underline{v}^{(l)}_{j+1}$ is obtained as follows. For $i \in \mathcal{N}$,

$$\mathcal{L}p^{(l)}_{i,j+1} = p^{(l)}_{i,j+1} + \frac{\mathrm{d}}{\mathrm{d}\epsilon}\, p^{(l)}_{i,j+1,\epsilon}\bigg|_{\epsilon=0} = p^{(l)}_{i,j+1} + \left(\frac{1}{k}\,\Delta u^{(l)}_{i,j+1} - \frac{1}{2}\Delta v^{(l)}_{i,j+1}\right), \tag{5.5a}$$

$$\begin{aligned} \mathcal{L}q^{(l)}_{i,j+1} &= q^{(l)}_{i,j+1} + \frac{\mathrm{d}}{\mathrm{d}\epsilon}\, q^{(l)}_{i,j+1,\epsilon}\bigg|_{\epsilon=0} \\ &= q^{(l)}_{i,j+1} + \Bigg(-\Delta u^{(l)}_{i+1,j+1} \\ &\quad + \left[2 + 4r^2 + 2h^2\,\frac{G'(u^{(l)}_{i,j+1})(u^{(l)}_{i,j+1} - u_{i,j}) - \{G(u^{(l)}_{i,j+1}) - G(u_{i,j})\}}{(u^{(l)}_{i,j+1} - u_{i,j})^2}\right]\Delta u^{(l)}_{i,j+1} \\ &\quad - \Delta u^{(l)}_{i-1,j+1}\Bigg)\,. \end{aligned} \tag{5.5b}$$

The increments $\Delta u^{(l)}_{i,j+1}$, for $i \in \mathcal{N}$, are obtained by solving the linear system

$$\mathcal{L}q^{(l)}_{i,j+1} = 0\,, \quad \text{for } i \in \mathcal{N}\,. \tag{5.5c}$$

The increments $\Delta v^{(l)}_{i,j+1}$, for $i \in \mathcal{N}$, are obtained from $\Delta u^{(l)}_{i,j+1}$ by solving

$$\mathcal{L}p^{(l)}_{i,j+1} = 0 \quad \text{for } i \in \mathcal{N}\,. \tag{5.5d}$$

[22] Note the presence of $u^{(l)}_{i+1,j+1}$ and $u^{(l)}_{i-1,j+1}$ in $q^{(l)}_{i,j+1}$, making the system of equations $q^{(l)}_{i,j+1}$, for $i \in \mathcal{N}$, coupled (and nonlinear).

An issue that receives less attention in the literature is the consistent linearization of the boundary conditions. It is important to have the linearization and the update of the boundary conditions done correctly, since these procedures affect the formation of the tangent matrix, while the correctness of the whole solution procedure affects the invariant-conserving properties of the algorithm. We present the linearization of the symmetry condition (4.10b), which as mentioned before has three different interpretations regarding the generality of its applications.

The first choice of boundary conditions is (4.12a), or equivalently (4.12b). The linearization (or incremental form) of (4.12b) is

$$\begin{aligned}&(u^{(l)}_{N+1,j+1} - u^{(l)}_{N,j+1}) + (u_{N+1,j} - u_{N,j}) + (\Delta u^{(l)}_{N+1,j+1} - \Delta u^{(l)}_{N,j+1}) = 0\,,\\ &(u^{(l)}_{-N,j+1} - u^{(l)}_{-N-1,j+1}) + (u_{-N,j} - u_{-N-1,j}) + (\Delta u^{(l)}_{-N,j+1} - \Delta u^{(l)}_{-N-1,j+1}) = 0\,.\end{aligned} \tag{5.6a}$$

The second choice of boundary conditions is (4.13a), or equivalently (4.13c), which has the linearized (incremental) form

$$\begin{aligned}&(u^{(l)}_{N+1,j+1} + u^{(l)}_{N,j+1}) - (u_{N+1,j} + u_{N,j}) + (\Delta u^{(l)}_{N+1,j+1} + \Delta u^{(l)}_{N,j+1}) = 0\,,\\ &(u^{(l)}_{-N-1,j+1} + u^{(l)}_{-N,j+1}) - (u_{-N-1,j} + u_{-N,j}) + (\Delta u^{(l)}_{-N-1,j+1} + \Delta u^{(l)}_{-N,j+1}) = 0\,.\end{aligned} \tag{5.6b}$$

More general boundary conditions yielding (4.10) are given in (4.14a), or equivalently (4.14b). Adding and subtracting the two equations in (4.14b), we obtain two simpler equations for enforcing the boundary conditions (4.10):

$$\begin{aligned}&(u_{N+1,j+1} - u_{N,j}) - (u_{-N,j+1} - u_{-N-1,j}) = 0\,,\\ &(u_{N+1,j} - u_{N,j+1}) - (u_{-N,j} - u_{-N-1,j+1}) = 0\,,\end{aligned} \tag{5.7}$$

whose linearized form is

$$\begin{aligned}&(u^{(l)}_{N+1,j+1} - u_{N,j}) - (u^{(l)}_{-N,j+1} - u_{-N-1,j}) + (\Delta u^{(l)}_{N+1,j+1} - \Delta u^{(l)}_{-N,j+1}) = 0\,,\\ &(u_{N+1,j} - u^{(l)}_{N,j+1}) - (u_{-N,j} - u^{(l)}_{-N-1,j+1}) + (\Delta u^{(l)}_{-N-1,j+1} - \Delta u^{(l)}_{N,j+1}) = 0\,.\end{aligned} \tag{5.8}$$

Either one of the equations (5.6a), (5.6b) or (5.8) will be the starting point for the boundary update procedure. In our computer implementation, we adopt (5.7) as boundary conditions and (5.8) as the corresponding update procedure.

5.2. Initial guess and update procedure for Newton iteration

The first step in the Newton–Raphson method is to choose the initial guess (or starting value for iteration $(l) = (0)$). An initial guess which is consistent should satisfy (5.2a), i.e.,

$$\frac{u^{(0)}_{i,j+1} - u_{i,j}}{k} - \frac{v^{(0)}_{i,j+1} - u_{i,j}}{2} = 0 \quad \forall i \in \mathcal{N}\,. \tag{5.9}$$

At the boundary points, the consistency of the initial guess requires that either one of the constraints equations (5.6a), (5.6b) or (5.8) be satisfied for $(l)=(0)$, i.e.,[23]

$$\begin{aligned}&(u^{(0)}_{N+1,j+1}-u^{(0)}_{N,j+1})+(u_{N+1,j}-u_{N,j})=0\,,\\&(u^{(0)}_{-N,j+1}-u^{(0)}_{-N-1,j+1})+(u_{-N,j}-u_{-N-1,j})=0\,.\end{aligned}\tag{5.10a}$$

$$\begin{aligned}&(u^{(0)}_{N+1,j+1}+u^{(0)}_{N,j+1})-(u_{N+1,j}+u_{N,j})=0\,,\\&(u^{(0)}_{-N-1,j+1}+u^{(0)}_{-N,j+1})-(u_{-N-1,j}+u_{-N,j})=0\,.\end{aligned}\tag{5.10b}$$

$$\begin{aligned}&(u^{(0)}_{N+1,j+1}-u_{N,j})-(u^{(0)}_{-N,j+1}-u_{-N-1,j})=0\,,\\&(u_{N+1,j}-u^{(0)}_{N,j+1})-(u_{-N,j}-u^{(0)}_{-N-1,j+1})=0\,.\end{aligned}\tag{5.10c}$$

Equation (5.9) together with (5.10a) or (5.10b) or (5.10c) guarantee a consistent initial guess. Also, one can notice that the value for the velocity at the boundary is not needed in the present algorithm. Since we use (5.7) and (5.8) in our computer implementation, the consistent initial guess for us is therefore (5.10c).

Next, the update formula are defined as

$$u^{(l+1)}_{i,j+1}=u^{(l)}_{i,j+1}+\Delta u^{(l)}_{i,j+1}\,,\qquad v^{(l+1)}_{i,j+1}=v^{(l)}_{i,j+1}+\Delta v^{(l)}_{i,j+1}\,.\tag{5.11}$$

The increments $\Delta u^{(l)}_{i,j+1}$ and $\Delta v^{(l)}_{i,j+1}$ are not, however, independent; they are in fact related through the algorithm and the consistent choice of initial guess. These properties are established in the following propositions.

PROPOSITION 5.1. The incremental velocity $\Delta v^{(l)}_{i,j+1}$ is related to the incremental displacement $\Delta u^{(l)}_{i,j+1}$, $\forall i\in\mathcal{N}$, according to the following relation:

$$\Delta v^{(l)}_{i,j+1}=\frac{2}{k}\,\Delta u^{(l)}_{i,j+1}\,,\quad \text{for } l=0,1,2,\ldots\,.\tag{5.12}$$

PROOF. A substitution of (5.11) into (5.5d), using (5.4a) and (5.5a), yields

$$\frac{u^{(l)}_{i,j+1}-u_{i,j}}{k}-\frac{v^{(l)}_{i,j+1}+v_{i,j}}{2}=0\,,\quad \text{for } l=1,2,3,\ldots\,.\tag{5.13}$$

Moreover, (5.13) is also valid for $l=0$ by a consistent initial guess satisfying (5.9). On the other hand, again by (5.4a) and (5.11),

$$\frac{u^{(k+1)}_{i,j+1}-u_{i,j}}{k}-\frac{v^{(k+1)}_{i,j+1}-v_{i,j}}{2}=\frac{u^{(l)}_{i,j+1}-u_{i,j}}{k}-\frac{v^{(l)}_{i,j+1}-v_{i,j}}{2}+\frac{1}{k}\,\Delta u^{(l)}_{i,j+1}-\frac{1}{2}\,\Delta v^{(l)}_{i,j+1}\,,\tag{5.14}$$

[23] The choice of the constraint equations on the initial guess, i.e., (5.10a) or (5.10b) or (5.10c), should be the same as the choice of the boundary update equations, i.e., (5.6a) or (5.6b) or (5.8).

for $l = 0, 1, 2, 3, \ldots$; thus by (5.13), it is clear that

$$\frac{1}{k}\,\Delta u^{(l)}_{i,j+1} - \frac{1}{2}\,\Delta v^{(l)}_{i,j+1} = 0 \quad \forall k\,. \quad \square \tag{5.15}$$

*PROPOSITION 5.2. Depending on the choice of the boundary condition constraints, i.e., (4.12b), (4.13c) or (5.7), at the l*th *Newton iteration, we have the following corresponding relationships on the incremental displacements*:

$$\Delta u^{(l)}_{N+1,j+1} = \Delta u^{(l)}_{N,j+1}\,, \qquad \Delta u^{(l)}_{-N-1,j+1} = \Delta u^{(l)}_{-N,j+1}\,; \tag{5.16a}$$

$$\Delta u^{(l)}_{N+1,j+1} = -\Delta u^{(l)}_{N,j+1}\,, \qquad \Delta u^{(l)}_{-N-1,j+1} = -\Delta u^{(l)}_{-N,j+1}\,; \tag{5.16b}$$

$$\Delta u^{(l)}_{N+1,j+1} = \Delta u^{(l)}_{-N,j+1}\,, \qquad \Delta u^{(l)}_{-N-1,j+1} = \Delta u^{(l)}_{N,j+1}\,. \tag{5.16c}$$

PROOF. The above is a direct result of the initial guess (5.10a), (5.10b), (5.10c) and the update formulas (5.6a), (5.6b), (5.8), respectively. $\square$

The consistent initial guess (5.9), together with (5.10a) or (5.10b) or (5.10c), form one part of the integral update procedure. Consider (5.9); there are two possible choices of setting the initial guess for the interior points: For $i \in \mathcal{N}$,

$$u^{(0)}_{i,j+1} = u_{i,j}\,, \qquad v^{(0)}_{i,j+1} = -v_{i,j} \tag{5.17a}$$

or

$$u^{(0)}_{i,j+1} = u_{i,j} + v_{i,j}k\,, \qquad v^{(0)}_{i,j+1} = v_{i,j}\,. \tag{5.17b}$$

The choice of (5.17a) or (5.17b) for the interior points will, however, dictate the initial guess for the boundary points by virtue of (5.10a,b,c). In our computer implementation, we use (5.17b) as the initial guess for the interior points.

5.3. *Solution strategy*

Ultimately, to obtain the incremental displacement at iteration l, we are to solve the following linear system, which has a particular structure:

$$\begin{bmatrix} \hat{a}_{-N} & b_1 & 0 & \ldots & \ldots & 0 & c \\ b & \hat{a}_{-N+1} & b & \ldots & \ldots & \ldots & 0 \\ 0 & b & \hat{a}_{-N+2} & b & \ldots & \ldots & \vdots \\ \vdots & \vdots & \cdot & \cdot & \cdot & & \vdots \\ \vdots & \vdots & & \cdot & \cdot & \cdot & \vdots \\ 0 & \ldots & \ldots & \ldots & b & \hat{a}_{N-1} & b \\ c & 0 & \ldots & \ldots & \ldots & b_1 & \hat{a}_N \end{bmatrix} \begin{Bmatrix} \Delta u^{(l)}_{-N,j+1} \\ \Delta u^{(l)}_{-N-1,j+1} \\ \vdots \\ \Delta u^{(l)}_{i,j} \\ \vdots \\ \Delta u^{(l)}_{N-1,j+1} \\ \Delta u^{(l)}_{N,j+1} \end{Bmatrix} = \begin{Bmatrix} \hat{d}_{-N} \\ \hat{d}_{-N+1} \\ \vdots \\ \hat{d}_i \\ \vdots \\ \hat{d}_{N-1} \\ \hat{d}_N \end{Bmatrix}, \tag{5.18}$$

where $b = -1$ and

$$\hat{a}_i := a^{(l)}_{i,j+1} = 4r^2 + 2 + 2h^2 \frac{G'(u^{(l)}_{i,j+1})(u^{(l)}_{i,j+1} - u_{i,j}) - (G(u^{(l)}_{i,j+1}) - G(u_{i,j}))}{(u^{(l)}_{i,j+1} - u_{i,j})^2}, \quad (5.19)$$

$$\hat{d}_i := r^{(l)}_{i,j+1} = 4hrv_{i,j} + (u^{(l)}_{i+1,j+1} - (2+4r^2)u^{(l)}_{i,j+1} + u^{(l)}_{i-1,j+1}) + (u_{i+1,j} - (2-4r^2)u_{i,j} + u_{i-1,j}) - 2h^2 \frac{G(u^{(l)}_{i,j+1}) - G(u_{i,j})}{u^{(l)}_{i,j+1} - u_{i,j}}. \quad (5.20)$$

For the coefficients b_1 and c, there are three different cases, each corresponding to a choice of boundary condition constraints by virtue of (5.16a), (5.16b), (5.16c), respectively:

(1) $b_1 = 2$ and $c = 0$;
(2) $b_1 = 0$ and $c = 0$;
(3) $b_1 = c = b = -1$.

In the first two cases, (5.18) is a typical tri-diagonal system, for which there are several established efficient methods of solution, e.g., the Thomas algorithm (see [36]).

Here, we pay particular attention to the third case, which is slightly different from the standard tri-diagonal system. There are non-zero coefficients (c) in the upper right corner and lower left corner of the matrix in (5.18). The system is, however, symmetric in this case. Using the Gauss elimination procedure, one can easily tridiagonalize the system. The following efficient recursion formulas, which are even simpler than the Thomas algorithm, are found for solving the problem (5.18) with $b_1 = c = b = -1$:

$$\begin{aligned}
&a_{-N} = \hat{a}_{-N}, \qquad c_{-N} = c, \qquad d_{-N} = \hat{d}_{-N}, \\
&a_i = \hat{a}_i - \frac{b^2}{a_{i-1}}, \quad \text{for } i \neq -N, N; \\
&c_i = -\frac{bc_{i-1}}{a_{i-1}}, \quad \text{for } i \neq -N, N-1; \\
&d_i = \hat{d}_i - \frac{d_{i-1}b}{a_{i-1}}, \quad \text{for } i \neq -N, N; \\
&a_N = \hat{a}_N - \sum_{i=-N}^{N-1} \frac{c_i^2}{a_i}, \qquad c_{N-1} = b - \frac{bc_{N-2}}{a_{N-1}}, \qquad d_N = \hat{d}_N - \sum_{i=-N}^{N-1} \frac{d_i c_i}{a_i}.
\end{aligned} \quad (5.21)$$

For completeness, the backsubstitution formulas are also included:

$$\begin{aligned}
&\Delta u^{(l)}_{N,j+1} = \frac{d_N}{a_N}, \qquad \Delta u^{(l)}_{N-1,j+1} = \frac{1}{a_{N-1}}(d_{N-1} - c_{N-1}\Delta u^{(l)}_{N,j+1}), \\
&\Delta u^{(l)}_{i,j+1} = \frac{1}{a_i}(d_i - b\Delta u^{(0)}_{i+1,j+1} - c_i \Delta u^{(l)}_{N,j+1}), \\
&\quad \text{for } i = N-2, N-3, \ldots, -N+1, -N.
\end{aligned} \quad (5.22)$$

6. Numerical results and discussions

We begin by describing the numerical examples solved by using the proposed algorithms in Section 6.1, followed by the discussions on the behaviour of the numerical results in Section 6.2. Some of the examples in Section 6.1 are classical and possess analytical solutions; appropriate references are given whenever possible.

6.1. Numerical examples

EXAMPLE 6.1. Collision of the sine-Gordon solitary waves. In this example, the potential function G takes the form

$$G(U) = 1 - \cos U \,. \tag{6.1}$$

The usual soliton solution for the sine-Gordon equation is

$$U(x, t) = 4 \tan^{-1}\left[\exp\left(\pm \frac{x - \beta t}{\sqrt{1-\beta^2}}\right)\right]. \tag{6.2}$$

Theoretically, the boundary conditions in this case are

$$u(\pm\infty, t) = 0\,, \qquad u_x(\pm\infty, t) = 0\,, \tag{6.3a}$$

which can also be translated into the following equivalent form:

$$u_t(\pm\infty, t) = 0\,, \qquad u_x(\pm\infty, t) = 0\,. \tag{6.3b}$$

We consider the collision of two solitons whose equations are given in (6.1), one with the + sign (kink) and the other with the − sign (antikink). The two solitons have equal but opposite velocities. The exact solution of this collision problem is also known as the kink-antikink solution of the sine-Gordon equation. To set up the initial conditions at $t = 0$, we consider the two kink and antikink solitons to be at a distance x_0 on each side of the origin O of the x axis. Thus the initial conditions employed for the collision of two solitons are

$$u(x, 0) = 4 \tan^{-1}\left[\exp\left(\frac{x + x_0}{\sqrt{1-\beta^2}}\right)\right] + 4 \tan^{-1}\left[\exp\left(\frac{-x + x_0}{\sqrt{1-\beta^2}}\right)\right], \tag{6.4a}$$

$$v(x, 0) = -4\left\{\frac{\dfrac{\beta}{\sqrt{1-\beta^2}}\exp\left(\dfrac{x + x_0}{\sqrt{1-\beta^2}}\right)}{1 + \exp\left(\dfrac{x + x_0}{\sqrt{1-\beta^2}}\right)^2} + \frac{\dfrac{\beta}{\sqrt{1-\beta^2}}\exp\left(\dfrac{-x + x_0}{\sqrt{1-\beta^2}}\right)}{1 + \exp\left(\dfrac{-x + x_0}{\sqrt{1-\beta^2}}\right)^2}\right\}. \tag{6.4b}$$

For the numerical results for $u(x, t)$ presented in Fig. 4, we use $x_0 = 5.0$, $\beta = 0.3$, together

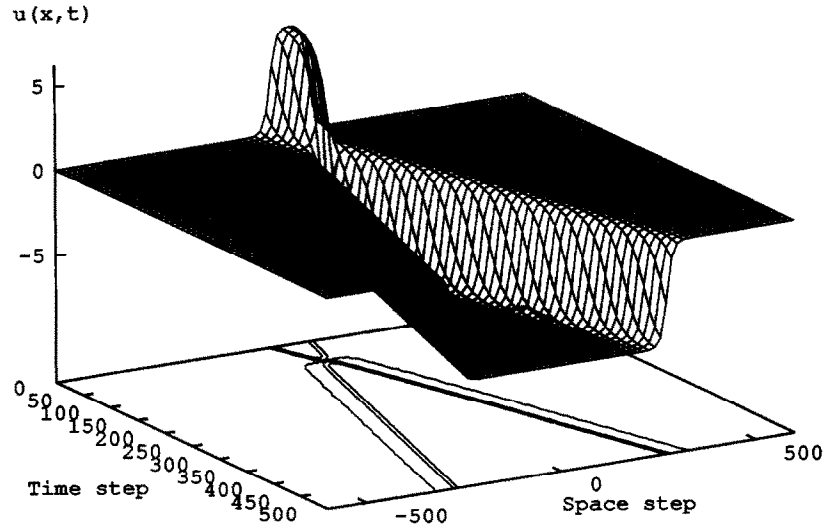

Fig. 4. *Soliton collision for the sine-Gordon equation.* With projected level contours, without hidden-line removal.

with the step sizes $h = 0.1$ and $k = 0.2$. The corresponding velocity $v(x, t)$ is shown in Fig. 5.[24] The above results can be compared with previous numerical results for this problem in [14, 19, 24].

EXAMPLE 6.2. Breather solution of the sine-Gordon equation. Since this is another case for the sine-Gordon equation, the potential function remains unchanged, and so are the boundary conditions. The breather solution is of the form (see [8])

$$U(x, t) = -4 \tan^{-1}\left[\frac{m}{\sqrt{1-m^2}}\,\frac{\sin(t\sqrt{1-m^2}+c_2)}{\cosh(mx+c_1)}\right]. \tag{6.5}$$

The initial velocity $v(x, 0) = u_t(x, 0)$ can be easily obtained from (6.5). In the present example, we use $m = 0.5$, $c_1 = 0$ and $c_2 = -10\sqrt{1-m^2}$. The step sizes used in this example are $h = 0.1$ and $k = 0.1$. The results can be compared with those presented in [19].

EXAMPLE 6.3. Collision of the ϕ^4 solitary waves. The potential function for this example is

$$G(u) = -\tfrac{1}{2}m^2u^2 + \tfrac{1}{4}\gamma u^4\,. \tag{6.6}$$

The boundary conditions also have different form,

$$u(\pm\infty, t) = -\sqrt{\frac{m^2}{2}}\,, \qquad u_x(\pm\infty, t) = 0\,. \tag{6.7}$$

[24] In fact, the negative of the velocity is plotted in Fig. 5 to show clearly the decrease in velocity at collision. Further, a refined mesh with $h = 0.05$ and $k = 0.05$ was used here to obtain more details.

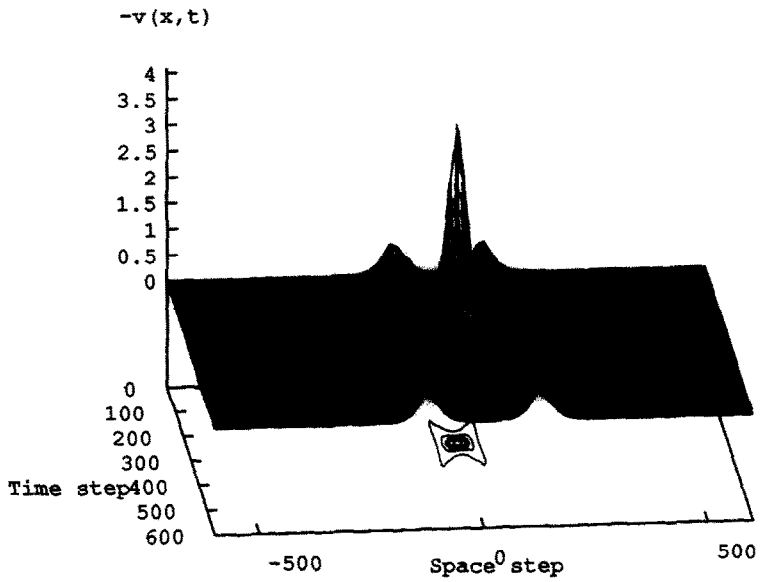

Fig. 5. *Negative velocity profile, i.e., $-v(x, t)$, for the sine-Gordon kink-antikink pair.* With projected level contours, without hidden-line removal.

Under the above conditions, a soliton-like solution is (see [41])

$$U(x, t) = \frac{m}{\sqrt{\gamma}} \tanh\left(\pm \frac{m(x - \beta t)}{\sqrt{2(1 - \beta^2)}} \right). \tag{6.8}$$

Similar to the construction of the collision of two kink and antikink solitons in Example 6.1, the initial conditions employed in the present example are

$$U(x, 0) = \frac{m}{\sqrt{\gamma}} \left[\tanh \frac{m(x + x_0)}{\sqrt{2(1 - \beta^2)}} + \tanh \frac{m(-x + x_0)}{\sqrt{2(1 - \beta^2)}} - 1 \right]. \tag{6.9a}$$

$$V(x, 0) = -\frac{m^2}{\sqrt{\gamma}} \frac{\beta}{\sqrt{2(1 - \beta^2)}} \left[2 - \tanh \frac{m(x + x_0)^2}{\sqrt{2(1 - \beta^2)}} - \tanh \frac{m(-x + x_0)^2}{\sqrt{2(1 - \beta^2)}} \right]. \tag{6.9b}$$

The computed displacement $u(x, t)$ shown in Fig. 7 was produced with $\gamma = 1/\pi^2$, $m = 1$, $x_0 = 3$, $\beta = 0.2$, together with the step sizes $h = 0.1$ and $k = 0.1$. The above results can be compared to those obtained in [42].

EXAMPLE 6.4. Solitary waves of the double sine-Gordon equation with positive velocity (+ve) *and their collision.* The potential function for the double sine-Gordon equation with positive velocity is

$$G(U) = 2 - \cos U - \cos(\tfrac{1}{2}U), \tag{6.10}$$

whereas the boundary conditions are the same as in the sine-Gordon equation. The soliton

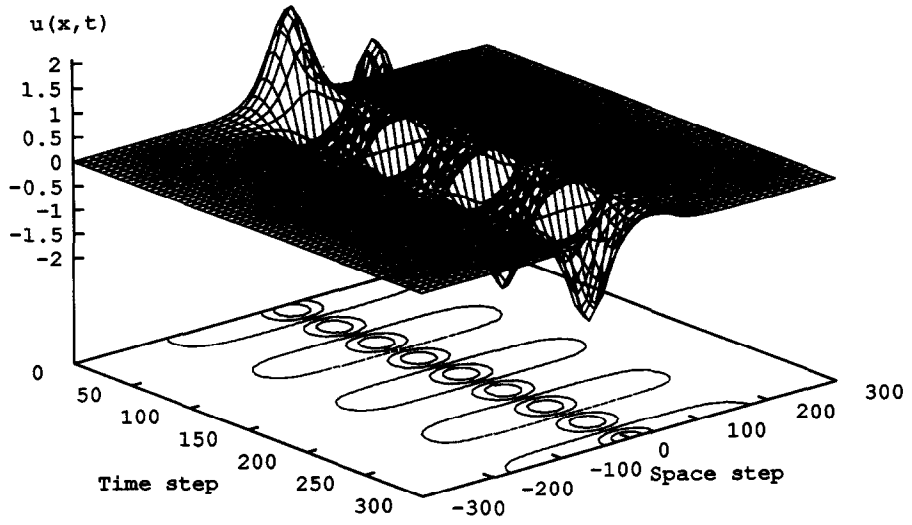

Fig. 6. *A stationary sine-Gordon breather solution.* With projected level contours, without hidden-line removal.

solution is of the following form (see [19]):

$$U(x,t) = -4\tan^{-1}\left(\frac{\sqrt{5}}{\sinh\theta}\right), \tag{6.11a}$$

$$\theta = \tfrac{1}{2}\sqrt{5}(x-\beta t)(1-\beta^2)^{-1/2}. \tag{6.11b}$$

The solutions in the previous examples are all symmetric. Figure 8 presents a non-symmetric solution called the single 'wobbling' 4π kink solution related to (6.11a,b) using as initial conditions

$$u(x,0) = 4\tan^{-1}(\exp(\theta+\delta)) + 4\tan^{-1}(\exp(\theta-\delta)) \tag{6.11c}$$

and

$$v(x,0) = 2\omega\,\mathrm{sech}(\theta+\delta) + 2\omega\,\mathrm{sech}(\theta-\delta), \tag{6.11d}$$

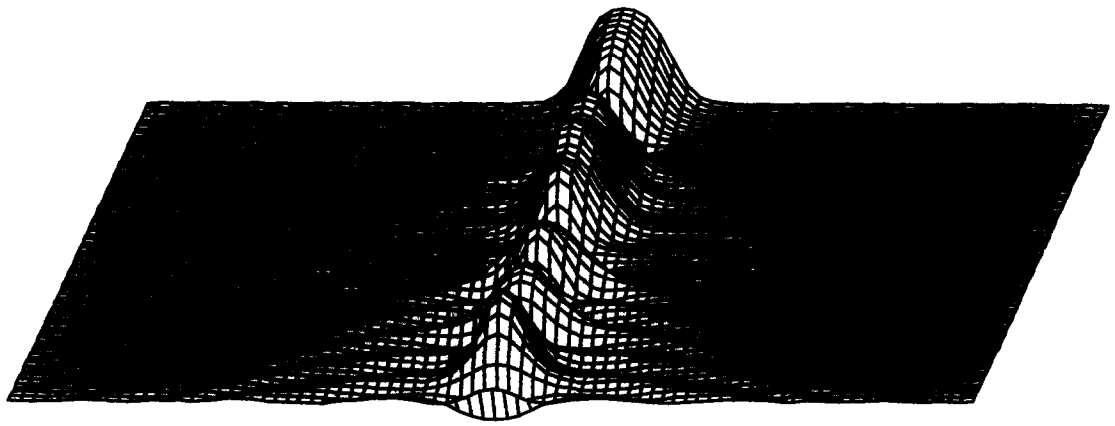

Fig. 7. *Kink-antikink collision for the ϕ^4 model.* With hidden-line removal.

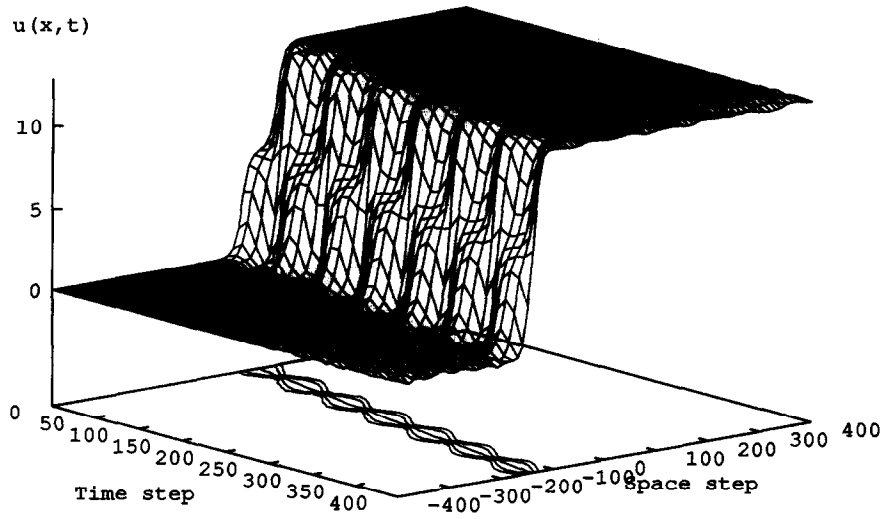

Fig. 8. The single 'wobbling' 4π kink of the double sine-Gordon equation with +ve sign.

where $\delta = \ln(\sqrt{5}+2) + x_0$, the parameter θ is as defined in (6.11b), and $\omega = \frac{1}{2}\sqrt{5}\beta(1-\beta^2)^{-1/2}$. Figure 8 is obtained with the following choice of parameters: $\beta = 0.3$, $x_0 = 2.5$, together with the step sizes $h = 0.1$ and $k = 0.1$.

Ablowitz et al. [24] also studied the same problem but with a transformed version, in which the potential function takes the form

$$G(U) = 2 - \cos U - \cos(\gamma U), \quad \gamma = 2. \tag{6.12}$$

The corresponding soliton solution, which is very similar to (6.11a), is as follows:

$$U(x,t) = 2\tan^{-1}\left(\frac{\sqrt{2\gamma+1}}{\sinh\dfrac{\sqrt{2\gamma+1}(x-\beta t)}{\sqrt{1-\beta^2}}}\right). \tag{6.13}$$

Now to obtain the solution for the collision of two kink/antikink solitons of the form (6.13), the initial conditions at $t = 0$ can be derived following the same procedure explained in Example 6.1. To produce Fig. 9, we use $\gamma = 2$, $\beta = 0.3$, with $x_0 = 5.0$ for the initial conditions (cf. expression (6.4a)), and the step sizes $h = 0.1$ and $k = 0.2$. The velocity profile $v(x,t)$ is presented in Fig. 10, using a refined mesh with $h = 0.05$ and $k = 0.05$. A comparison of Fig. 5 and Fig. 10 reveals the distinct velocity profiles for the two collisions in Example 6.1 and Example 6.4, whereas the displacement profiles are similar (Figs. 4 and 9). The results presented in Fig. 8–10 can be compared with previous results obtained in [43, 44].

EXAMPLE 6.5. Double sine-Gordon equation with negative velocity (−ve). The governing equation for this type of problem takes the form

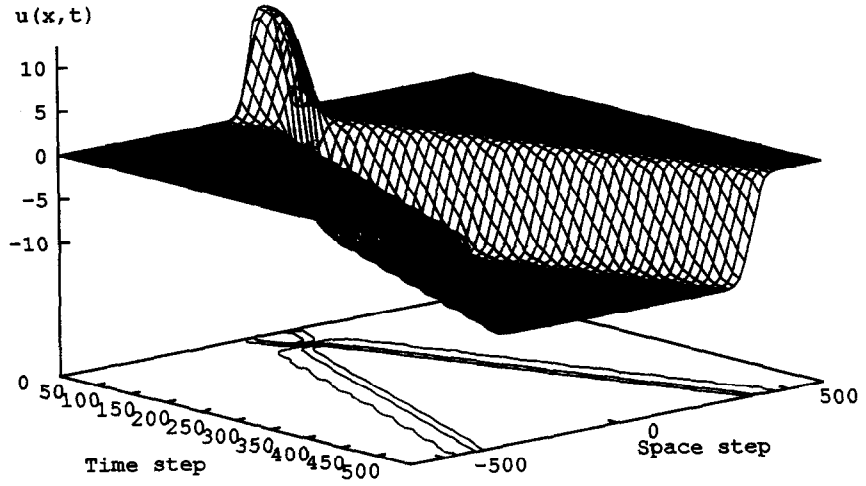

Fig. 9. Soliton collision for the double sine-Gordon equation with +ve sign.

$$U_{tt} - U_{xx} - (\sin U + \tfrac{1}{2} \sin \tfrac{1}{2} U) = 0 , \tag{6.14a}$$

with the corresponding potential function

$$G(U) = 2\left(\cos \frac{U}{2} + \frac{1}{4}\right)^2 . \tag{6.14b}$$

There are two types of soliton (kink) solution for this equation. One is called the $(4\pi - \delta)$ kink, and the other the 2δ kink.

(a) *The* $(4\pi - \delta)$ *kink.* The boundary conditions for the $(4\pi - \delta)$ kink are

$$\begin{aligned} &U(+\infty, t) = \delta \quad \text{and} \quad U(-\infty, t) = 4\pi - \delta , \\ &U_x(\pm\infty, t) = 0 , \end{aligned} \tag{6.15}$$

with the exact soliton (kink) solution

$$U(x, t) = 2\pi + 4 \tan^{-1} (\sqrt{\tfrac{3}{5}} \tanh \tfrac{1}{2} \theta) , \tag{6.16a}$$

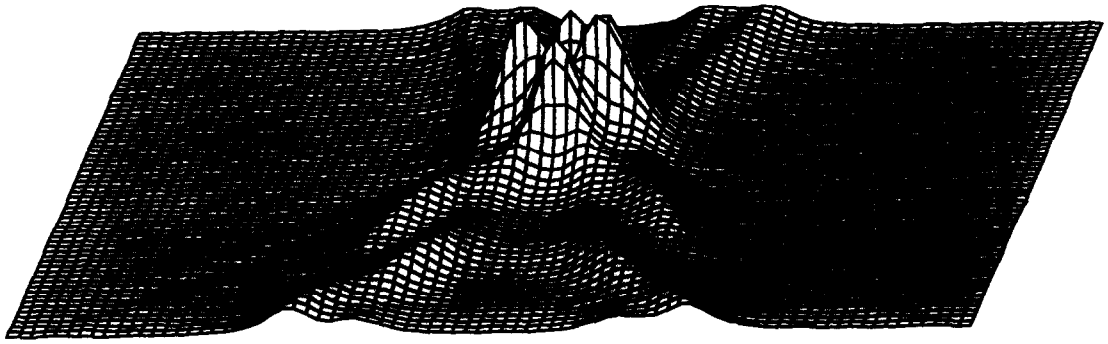

Fig. 10. Negative of velocity profile, i.e., $-v(x, t)$, for the double sine-Gordon kink/antikink collision.

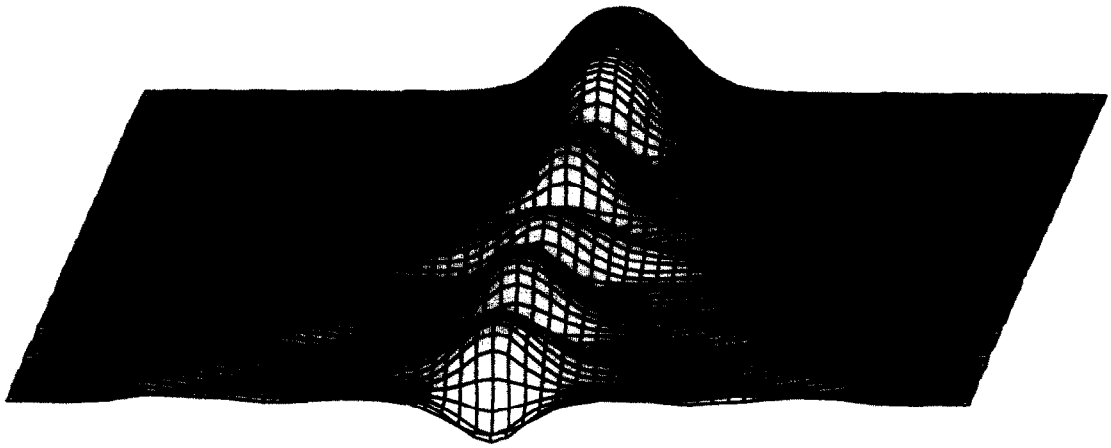

Fig. 11. The long-lived breather-like state of the double sine-Gordon equation with −ve sign.

$$\theta = \chi(x - \beta t) \tag{6.16b}$$

$$\chi = \sqrt{\tfrac{15}{16}}\,(1-\beta^2)^{-1/2}\,. \tag{6.16c}$$

The initial conditions for the collision of two kink/antikink solitons in Figs. 11–13, obtained following the same procedure explained in Example 6.1, are

$$u(x,0) = 4[\tan^{-1}(\sqrt{\tfrac{3}{5}}\tanh\{\tfrac{1}{2}\chi(x+x_0)\})\tan^{-1}(\sqrt{\tfrac{3}{5}}\tanh\{\tfrac{1}{2}\chi(-x+x_0)\})] \tag{6.16d}$$

and

$$v(x,0) = 2\sqrt{\tfrac{3}{5}}\,\chi\beta\left[\frac{1}{(1+\frac{3}{5}\tanh(\frac{1}{2}\chi(x+x_0))^2)\cosh(\chi(x+x_0))^2} + \frac{1}{(1+\frac{3}{5}\tanh(\frac{1}{2}\chi(x-x_0))^2)\cosh(\chi(x-x_0))^2}\right]. \tag{6.16e}$$

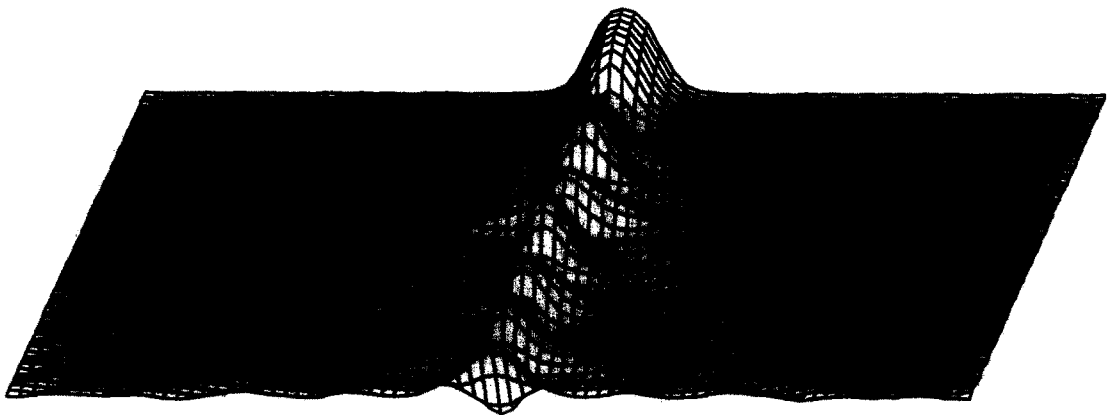

Fig. 12. Behaviour of the $(4\pi - 2\delta)$ kink/antikink collision for the double sine-Gordon with −ve sign: $\beta = 0.1$.

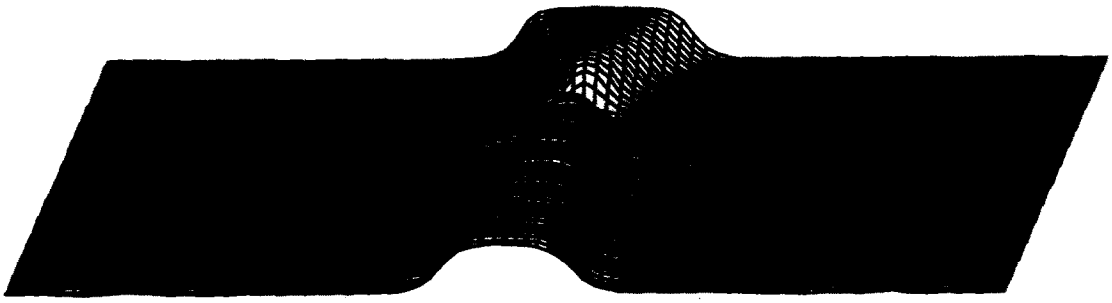

Fig. 13. The collision of two $(4\pi - 2\delta)$ kink/antikink solitons of the double sine-Gordon equation with $-$ve sign: $\beta = 0.36$.

In Fig. 11, a long-lived breather-like solution of (6.14a) is given, using $x_0 = 3$ and $\beta = 0$, with step sizes $h = 0.05$, $k = 0.05$. Figure 12 is obtained using the parameters $x_0 = 3$, $\beta = 0.1$, $h = 0.1$ and $k = 0.1$. Figure 13 is obtained with a larger parameter $\beta = 0.36$; the other parameters are the same as those for Fig. 12. The values for β in the figure captions of Figs. 12 and 13 are $\beta = 0.1$ and $\beta = 0.36$, respectively.

(b) *The 2δ kink.* For the 2δ kink case, the governing equation and potential function are the same as for the $(4\pi - \delta)$ kink, with the only difference in the boundary conditions

$$U(+\infty, t) = \delta \quad \text{and} \quad U(-\infty, t) = -\delta\,, \qquad U_x(+\infty, t) = 0\,. \tag{6.17}$$

The corresponding kink solution is

$$U(x, t) = 4 \tan^{-1} (\sqrt{\tfrac{3}{5}} \tanh \tfrac{1}{2}\theta)\,, \tag{6.18a}$$

$$\theta = \chi(x - \beta t) \tag{6.18b}$$

$$\chi = \sqrt{\tfrac{15}{16}}\, (1 - \beta^2)^{-1/2}\,. \tag{6.18c}$$

The expressions for the initial conditions are very similar to (6.16d,e), but with the factor $\frac{5}{3}$ instead of factor $\frac{3}{5}$. Figure 14 is obtained with $\beta = 0.36$, $x_0 = 6$ and the step sizes $h = 0.1$, $k = 0.1$.

6.2. *Discussions on the invariant properties*

The above results are obtained using Algorithm I. Along with the computation for the displacement $u(x, t)$ and velocity $v(x, t)$, we monitor the change in the total energy, linear momentum and angular momentum. For Example 6.1, the simulation of elastic collision of sine-Gordon solitary waves, we compute the numerical solution by using the spatial step size $h = 0.1$ and with three different time step sizes $\lambda = k/h = 0.5$, 1.0 and 2.0. The computed results are virtually the same for the three cases regarding the displacement profile, the velocity profile and the system's total energy at each time step. We record below the numerical

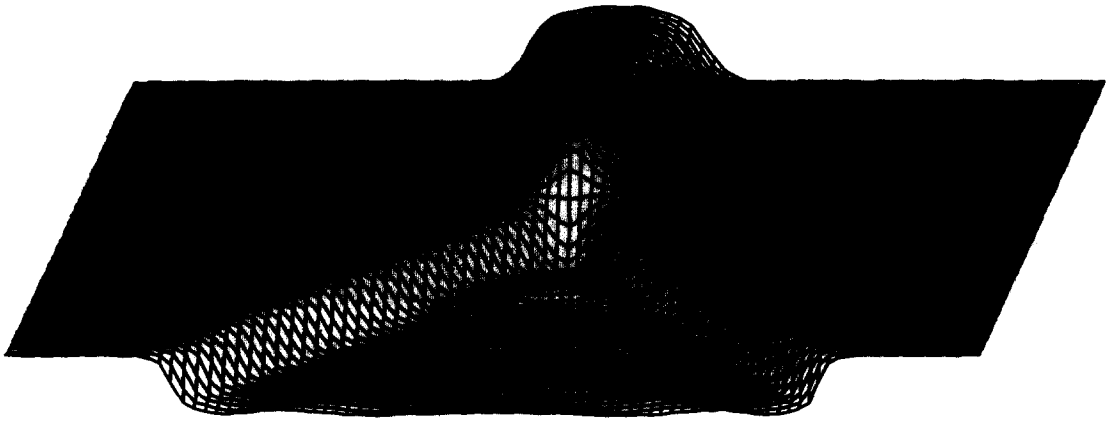

Fig. 14. Collision of two 2δ kink/antikink solitons of the double sine-Gordon equation with $-$ve sign: $\beta = 0.36$.

results obtained for the total discrete energy (Table 1), the momentum (Table 2) and the angular momentum (Table 3). From Table 1, it can be seen that the proposed Algorithm I can preserve the total energy accurately up to more than 10 digits with a rather coarse mesh. From Tables 2 and 3, one can also notice that the fluctuation of linear momentum and the angular momentum are also very small when using Algorithm I. The numerical values in Tables 1, 2

Table 1
Collision of sine-Gordon solitons: total discrete energy versus time

Time	Energy
0.000000000000	16.7691395594
0.800000000000	16.7691395594
1.600000000000	16.7691395594
2.400000000000	16.7691395594
3.200000000000	16.7691395594
4.000000000000	16.7691395594
4.800000000000	16.7691395594
5.600000000000	16.7691395594
6.400000000000	16.7691395594
7.200000000000	16.7691395594
⋮	⋮
71.200000000000	16.7691395595
72.000000000000	16.7691395595
72.800000000000	16.7691395595
73.600000000000	16.7691395595
74.400000000000	16.7691395595
75.200000000000	16.7691395595
76.000000000000	16.7691395595
76.800000000000	16.7691395595
77.600000000000	16.7691395595
78.400000000000	16.7691395595
78.600000000000	16.7691395595
78.800000000000	16.7691395595
80.000000000000	16.7691395595

Table 2
Collision of sine-Gordon solitons: linear momentum versus time

Time	Momentum
0.0000000000000	0.0000000000
0.8000000000000	0.0000000000
1.6000000000000	0.0000000000
2.4000000000000	0.0000000000
3.2000000000000	0.0000000000
4.0000000000000	0.0000000000
4.8000000000000	0.0000000000
5.6000000000000	0.0000000000
6.4000000000000	0.0000000000
7.2000000000000	0.0000000000
8.0000000000000	0.0000000000
8.8000000000000	0.0000000000
9.6000000000000	0.0000000000
10.4000000000000	0.0000000000
11.2000000000000	0.0000000000
⋮	⋮
70.4000000000000	0.0000000005
71.2000000000000	−0.0000000007
72.0000000000000	−0.0000000011
72.8000000000000	0.0000000000
73.6000000000000	0.0000000006
74.0000000000000	0.0000000002
75.2000000000000	0.0000000003
76.0000000000000	0.0000000003
76.8000000000000	−0.0000000000
77.6000000000000	−0.0000000010
78.4000000000000	−0.0000000007
79.2000000000000	−0.0000000000
80.0000000000000	0.0000000007

Table 3
Collision of sine-Gordon solitons: angular momentum versus time

Time	Angular momentum
0.0000000000000	0.0000000000
0.8000000000000	−0.0000000000
1.6000000000000	0.0000000000
2.4000000000000	0.0000000000
3.2000000000000	−0.0000000000
4.0000000000000	−0.0000000000
4.8000000000000	−0.0000000036
5.6000000000000	−0.0000000036
6.4000000000000	−0.0000000036
7.2000000000000	−0.0000000036
8.0000000000000	−0.0000000036
8.8000000000000	−0.0000000036
9.6000000000000	−0.0000000036
10.4000000000000	−0.0000000036
⋮	⋮
68.8000000000000	−0.0000000073
69.6000000000000	0.0000000507
70.4000000000000	0.0000000362
71.2000000000000	−0.0000000561
72.0000000000000	−0.0000000823
72.8000000000000	0.0000000058
73.6000000000000	0.0000000450
74.4000000000000	0.0000000156
75.2000000000000	0.0000000214
76.0000000000000	0.0000000174
76.8000000000000	−0.0000000078
77.6000000000000	−0.0000000873
78.4000000000000	−0.0000000598
79.2000000000000	−0.0000000009
80.0000000000000	0.0000000583

and 3 are from the first 80 computation time units of the computed solution in Example 6.1. Recall that the step sizes employed in this example are $h = 0.1$ and $k = 0.2$, which correspond to $\lambda = k/h = 2 > 1$, testifying to the robustness of the proposed algorithm. Similar remarks can be made for the remaining examples.

7. Closure

We have presented a formalism to derive second-order invariant-conserving algorithms for the nonlinear Klein–Gordon equation. Three algorithms are derived based on the proposed formalism; they conserve in discrete form either the total energy or the momentum. A geometric interpretation of these algorithms is given. Their stability and accuracy are investigated in some detail. We address the different choices of boundary conditions for the discretized problem. An efficient solution procedure has been discussed in detail for the

computer implementation of the proposed algorithms. An extension of the proposed algorithms to non-conservative problems that have as basis the nonlinear Klein–Gordon equation is also given. Finally, we have presented several numerical examples, which include collisions of solitary waves, to demonstrate the conservation and robustness properties of the proposed algorithms. In a follow-up work [45], we will present a proof of the exact algebraic conservation of the discrete invariants in the algorithms presented here, the nonlinear stability aspects of these algorithms, together with detailed discussions on some previously published algorithms for the NLKGE. Some improvements on the results reported here will also be presented.

Acknowledgment

The research is supported by a grant from the National Science Foundation and from the SRAP program of the Florida Space Grant Consortium. This support is gratefully acknowledged.

References

[1] E. Infeld, and G. Rowlands, Nonlinear Waves, Solitons and Chaos (Cambridge Univ. Press, New York, 1990).
[2] J.J. Liouville, Mathématiques Pures et Appliquées (Paris) 18 (1853) 71–72.
[3] R.F. Dashen, B. Hasslacher and A. Neuveu, Nonperturbative methods and extended hadron models in field theory II. Two-dimensional model and extended hadrons, Phys. Rev. D 10 (1974) 4130–4138.
[4] J.C. Eilbeck, Numerical studies of solitons, in: A.R. Bishop and T. Schneider, eds., Solitons and Condensed Matter Physics (Springer, New York, 1978) 28–43.
[5] A.C. Scott, Active and Nonlinear Wave Propagation in Electronics (Wiley/Interscience, New York, 1970).
[6] J.D. Gibbon, I.N. James and I.M. Moroz, The sine-Gordon equation as a model for a rapidly rotating baroclinic fluid, Phys. Scripta 20 (1979) 402–408.
[7] A. Barone, F. Esposito, C.J. Magee and A.C. Scott, Theory and applications of the sine-Gordon equation, Riv. Nuovo Cimento 1 (1971) 227–267.
[8] G.L. Lamb, Jr., Elements of Soliton Theory (Wiley, New York, 1980).
[9] A.P. Fordy, ed., Soliton Theory: A Survey of Results (Manchester Univ. Press, 1990).
[10] G.B. Whitham, Linear and Nonlinear Waves (Wiley, New York, 1974).
[11] N.J. Zabusky and M.D. Kruskal, Phys. Rev. Lett. 15 (1965) 240–243.
[12] M. Roseau, Vibrations in Mechanical Systems: Analytical Methods and Applications (Springer, Berlin, 1987).
[13] C.S. Gardner, J.M. Greene, M.D. Kruskal and R.M. Miura, Method for solving the Korteweg–De Vries equation, Phys. Rev. Lett. 19 (1967) 1095–1097.
[14] G.L. Lamb, Jr., Higher conservation laws in ultrashort optical pulse propagation, Phys. Lett. 32A (1970) 251–252.
[15] H. Steudel, Noether's theorem and higher conservation laws in ultrashort pulse propagation, Ann. Phys. 7 Folge, Band 32, Heft 3 (1975) S. 217–226.
[16] M.J. Ablowitz, D.J. Kaup, A.C. Newell and H. Segur, Method for solving the sine-Gordon equation, Phys. Rev. Lett. 30 (1973) 1262–1264.
[17] L.D. Fadeev, A Hamiltonian interpretation of the inverse scattering method, in: R.K. Bullough and P.J. Caudrey, eds., Solitons (Springer, New York, 1980).
[18] P.W. Kitchenside, P.J. Caudrey and R.K. Bullough, Soliton-like spin waves in ^{3}He B, Phys. Scripta 20 (1979) 673–680.

[19] R.K. Dodd, J.C. Eilbeck, J.D. Gibbon and H.C. Morris, Solitons and Nonlinear Wave Equations (Academic Press, London, 1982).
[20] P.D. Lax, Integrals of nonlinear equations of evolution and solitary waves, Comm. Pure Appl. Math. 21 (1968) 467–490.
[21] J. Argyris and M. Haase, An engineer's guide to soliton phenomena: Application of the finite element method, Comput. Methods Appl. Mech. Engrg. 61 (1987) 71–122.
[22] L.R.T. Gardner, G.A. Gardner and A.H.A. Ali, Simulations of solitons using quadratic spline finite elements, Comput. Methods Appl. Mech. Engrg. 92 (1991) 231–243.
[23] G.F. Carey and Y. Shen, Approximations for the KdV equation by least squares finite elements, Comput. Methods Appl. Mech. Engrg. 93 (1991) 1–11.
[24] M.J. Ablowitz, M.D. Kruskal and J.F. Ladik, Solitary wave collisions, SIAM J. Appl. Math. 36 (1979) 428–437.
[25] J.C. Strikwerda, Finite Difference Schemes and Partial Differential Equations (Wadsworth & Brooks/Cole, California, 1989).
[26] T.J.R. Hughes, T.K. Caughey and W.K. Liu, Finite element methods for nonlinear elastodynamics which conserve energy, ASME J. Appl. Mech. 45 (1978) 366–370.
[27] D. Greenspan, Conservative numerical methods for $\ddot{x} = f(x)$, J. Comput Phys. 56 (1984) 28–41.
[28] Y.S. Chin and C. Qin, Explicit energy-conserving schemes for the three-body problem, J. Comput. Phys. 83 (1989) 485–493.
[29] J.C. Simo and T. Honein, Variational formulation, discrete conservation laws, and path-domain independent integrals for elasto-viscoplasticity, ASME J. Appl. Mech. 57 (1990) 488–497.
[30] W. Strauss and L. Vazquez, Numerical solution of a nonlinear Klein–Gordon equation, J. Comput. Phys. 28 (1978) 271–278.
[31] P.S. Lomdahl, O.H. Soerensen, P.L. Christiansen, A.C. Scott and J.C. Eilbeck, Bunched multi-solitons in Josephson tunnel junctions, Phys. Rev. B 24 (1981) 7460–7462.
[32] J.C. Luke, A note on particle-like solutions for a nonlinear complex-valued Klein–Gordon equation, SIAM J. Appl. Math. 51 (1991) 1194–1204.
[33] C.S. Morawetz, Time decay for the nonlinear Klein–Gordon equation, Proc. Roy. Soc. London Ser. A 306 (1968) 291–296.
[34] C.S. Morawetz, Notes on Time Decay and Scattering for Some Hyperbolic Problems (Soc. Ind. Appl. Math., Philadelphia, PA, 1975).
[35] R.D. Richtmyer and K.W. Morton, Difference Methods for Initial Value Problems (Wiley, New York, 1967).
[36] W.F. Ames, Numerical Methods for Partial Differential Equations, 2nd Edition (Academic Press, New York, 1977).
[37] G.D. Smith, Numerical Solution of Partial Differential Equations: Finite Difference Methods, 3rd Edition (Oxford Univ. Press, New York, 1985).
[38] P.W. Kitchenside, R.K. Bullough and P.J. Caudrey, Creation of spin waves in ^{3}He B, in: A.R. Bishop and T. Schneider, eds. Solitons and Condensed Matter Physics (Springer, New York, 1978) 291–296.
[39] P.L. Christiansen, P.S. Lomdahl, A.C. Scott, O.H. Soerensen and J.C. Eilbeck, Internal dynamics of long Josephson junction oscillators, Appl. Phys. Lett. 39 (1981) 108–110.
[40] P.L. Christiansen and P.S. Lomdahl, Numerical study of 2 + 1 dimensional sine-Gordon solitons, Physica D 2 (1981) 482–494.
[41] A.E. Kudryavtsev, Soliton-like solutions for a Higgs scalar field, JETP Lett. 22 (1975) 82–83.
[42] C.A. Wingate, Numerical search for a ϕ^4 breather mode, SIAM J. Appl. Math. 43 (1983) 120–140.
[43] R.K. Bullough, P.J. Caudrey and H.M. Gibbs, The double sine-Gordon equations: A physically applicable system of equations, in: R.K. Bullough and P.J. Caudrey, eds., Solitons (Springer, New York, 1980).
[44] R.K. Bullough and P.J. Caudrey, The multiple sine-Gordon equations in non-linear optics and in liquid ^{3}He, in: F. Calogero, ed., Nonlinear Evolution Equations Solvable by the Spectral Transform (Pitman, Boston, 1978).
[45] S. Li and L. Vu-Quoc, Finite-difference-calculus invariant structure of a class of algorithms for the nonlinear Klein–Gordon equation, Report No. AeMES-TR-92-2-02, Aerospace Engineering, Mechanics & Engineering Science, University of Florida, 1992; SIAM J. Numer. Anal. submitted.
[46] F. Calogero, Nonlinear Evolution Equations Solvable by the Spectral Transform (Pittman, London, 1978).