JOURNAL OF COMPUTATIONAL PHYSICS **145,** 359–381 (1998)
ARTICLE NO. CP986013

Numerical Integration of $(2+1)$ Dimensional PDEs for S^2 Valued Functions

B. Piette and W. J. Zakrzewski

Department of Mathematical Sciences, University of Durham, Durham, DH1 3LE, England
E-mail: B.M.A.G.Piette@uk.ac.durham and W.J.Zakrzewski@uk.ac.durham

Received October 23, 1997; revised April 17, 1998

We investigate the accuracy of various numerical methods used to simulate the time evolution of partial differential equations for functions valued in S^2. We use three different methods to describe the fields: a unit length vector, the polar angles, and a complex field. We derive some nonlinear finite difference operators and we compare the different methods used in simulations. For the time integration, we employ both the 4th order Runge–Kutta and the leapfrog methods. © 1998 Academic Press

1. INTRODUCTION

Classical field theories valued in non-flat manifolds arise very often in the description of various phenomena in different areas of physics. In elementary particle physics, for example, the Yang Mills theories, the Skyrme model, and the monopoles, all involve fields valued in non-Abelian Lie groups (typically $SU(n)$). The fact that the fields are valued in topologically non-trivial manifolds leads to the existence of non-trivial stable solutions which, in turn, are good candidates to describe elementary particles.

In solid states physics, ferro-magnetic materials and liquid crystals are often described by a unit vector which corresponds to, respectively, the local orientation of the magnetisation or the orientation of molecules in liquid crystals. The space of configurations is then a two dimensional sphere (ferro-magnets) or a projective space RP^2 when the vector has a direction but no orientation (some liquid crystals).

Cosmological strings are also described by nonlinear sigma models which take values in a 2 dimensional sphere. Other applications involve the quantum Hall effect and solitonic structures in ferromagnets and antiferromagnets, *etc.*, which all involve, in their mathematical description, topologically non-trivial curved manifolds.

Most of such models possess extended structure solutions whose dynamics are responsible for the physical phenomena. To study dynamics one has to solve the classical equations of motion, which, due to the topological non-triviality of the target manifold, are

0021-9991/98 $25.00

non-linear. So, choosing some coordinate variables on the target manifold, the problem reduces to having to solve a set of coupled nonlinear partial differential equations, often also with algebraic constraints. Moreover, sometimes one trades the constraints for having to deal with fields which can (and do) take arbitrarily large values or possess coordinate singularities.

Luckily, in most cases the manifold can be described in several different but equivalent ways. This can be exploited when solving the equations. As the equations can, virtually never, be solved analytically we have to solve them numerically. This is particularly true when we are interested in time dependent problems.

As we will see, the different descriptions often have very different properties; moreover, the number of functions used can depend on the description used. This has important consequences for the amount of memory and time required to solve the equations numerically.

Consider, for example, a model valued in the two dimensional sphere S^2. One can describe it by a 3 component real vector field $\phi = (\phi_1, \phi_2, \phi_3)$ normalised to 1:

$$\phi \cdot \phi = 1. \tag{1.1}$$

The evolution of the model will then be described by 3 equations (one for each component of ϕ) together with (1.1). The advantage of this formulation is that each function (component of ϕ) takes values in the interval $[-1, 1]$. The main problem, however, is to make sure that the evolving fields satisfy the constraint. Notice, that one of the consequences of (1.1) is

$$\phi \cdot \frac{\partial \phi}{\partial u} = 0, \tag{1.2}$$

where u is any of the space time coordinates $t, x, \ldots,$ together with further conditions which follow from (1.2) by its differentiation.

To avoid the problem of constraints one can parametrise the sphere by introducing a set of polar angles on it, *e.g.*, θ and φ related to ϕ by

$$\begin{aligned}\phi_1 &= \cos(\theta)\cos(\varphi)\\ \phi_2 &= \cos(\theta)\sin(\varphi)\\ \phi_3 &= \sin(\theta).\end{aligned}$$

This description of the model involves only 2 fields without any algebraic constraint. However, this description has its drawback in that this formulation has a coordinate singularity, as for $\theta = 0$, φ is not defined. This, in turn, implies that, in general, the equations of the model are singular in the fields, which can make the numerical methods unstable.

A third natural way to describe a sphere involves a conformal projection of the sphere onto the complex plane. Defining w as such a complex field we can take

$$w = \frac{\phi_1 + i\phi_3}{1 - \phi_3}. \tag{1.3}$$

Again, this description involves only 2 real functions with no algebraic constraint. This time, however, the problem is that the field w is infinite at the north pole of the sphere

($\phi_3 = 1$). Moreover, for values of ϕ with ϕ_3 close to 1, w becomes very large, which makes most numerical methods difficult to implement.

The most common manifolds encountered in classical field theories are Riemannian symmetric spaces, and in particular, the spheres S^n (Sigma models, cosmic strings), unitary groups $SU(N)$ (Monopoles, Yang Mills fields, fields of the Skyrme model), as well as the projective planes $\mathbb{C}P^n$ (sigma models) and $\mathbb{R}P^n$ (liquid crystals). Each of them can be described by a real or a complex matrix satisfying a set of algebraic constraints [1], but can also be described by a set of parameters equal to the dimension of the manifold.

Over the past few years, we have performed many numerical simulations studying various extended structures in many $(2+1)$ dimensional models using both relativistic and Landau–Lifschitz type dynamics. Most of our studies involved fields valued in S^2. Sometimes the equations of motion involved only the simplest σ model terms, at other times additional terms were added (such as Skyrme and various potential terms). In our studies we used various formulations of our models, always paying special attention to the reliability of the derived results. Of course, all numerical methods involve numerical errors; so in our simulations we have always tried to reduce such errors to a level at which we can trust the general features of the studied phenomena. So, for some phenomena (like the 90° scattering [2]), relatively small lattices have already been sufficient, for some others (like the rate of soliton "shrinking" in the pure S^2 model [3]) we needed large lattices and sophisticated multi-grid methods.

Having gained experience from some simulations we have decided to compare our methods (for our types of fields) to decide which methods to use in future, and how efficient they are (both from the computer memory point of view and the time required for a given simulation). In each case we make sure that errors are very small, but then there is no point in using a method which reduces a negligible error even further at a price of lengthening significantly the CPU time of the simulation. At the same time, if the CPU time or the memory requirements are not increased there is no "harm" in using a more accurate method. In fact, for $(2+1)$ dimensional model simulations, given the present state of computer technology, most methods work quite well and produce reliable results in reasonable times. Some simulations of $(3+1)$ dimensional models have also been performed [4] but they require much more computing power hence the need to find accurate and economical methods of numerical integration. We thus hope that the results of this paper, in addition to providing us with an insight into the accuracy of simulations in $(2+1)$ dimensions, will also be useful when thinking of simulating the dynamics of extended structures in $(3+1)$ dimensions.

Thus, in this paper we compare different methods of studying the dynamics of soliton-like structures of the $(2+1)$ dimensional S^2 sigma model. We use 3 different formulations of the model. For each formulation, we tackle the problems created by the field description and develop a stable method to solve the equations satisfied by the fields. We compare the results of all methods, focusing our attention on the accuracy of the results and the computing requirements.

2. THE S^2 SIGMA MODEL

We will concentrate our attention on the numerical investigations of the extended structures and their scattering properties in the $(2+1)$ dimensional sigma model. This model is defined on the two dimensional plane (locally parametrised by x and y) and the field takes its value in the sphere S^2. The model itself is defined by the Lagrangian,

$$L = \int dx\, dy \left(\frac{1}{2}\left(\frac{\partial \phi}{\partial t} \cdot \frac{\partial \phi}{\partial t} - \frac{\partial \phi}{\partial x} \cdot \frac{\partial \phi}{\partial x} - \frac{\partial \phi}{\partial y} \cdot \frac{\partial \phi}{\partial y} \right) - V(\phi) \right.$$
$$- \frac{K^2}{2}\left[\left|\frac{\partial \phi}{\partial x}\right|^2 \left|\frac{\partial \phi}{\partial y}\right|^2 - \left|\frac{\partial \phi}{\partial t}\right|^2 \left(\left|\frac{\partial \phi}{\partial x}\right|^2 + \left|\frac{\partial \phi}{\partial y}\right|^2 \right) - \left(\frac{\partial \phi}{\partial x} \cdot \frac{\partial \phi}{\partial y} \right)^2 \right.$$
$$\left.\left. + \left(\frac{\partial \phi}{\partial t} \cdot \frac{\partial \phi}{\partial x} \right)^2 + \left(\frac{\partial \phi}{\partial t} \cdot \frac{\partial \phi}{\partial y} \right)^2 \right] \right), \tag{2.1}$$

where ϕ satisfies $\phi \cdot \phi = 1$. The first term of (2.1) describes the pure S^2 sigma model ($K = 0$, $V = 0$). Its extended static solutions are unstable (they blow up in a finite time) [2, 3]. For this reason, to stabilise these solutions, the last two terms were introduced [5–7]. The term proportional to K^2 is called the Skyrme term and V is a potential term. Their combined effect is to set the scale and so to fix the size of the static solitonic solutions thus preventing the solitons from expanding or shrinking.

From (2.1) we can derive the following expression for the total energy of a given field configuration:

$$E = \int dx\, dy \left\{ \left(\frac{\partial \phi}{\partial t} \cdot \frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} \cdot \frac{\partial \phi}{\partial x} + \frac{\partial \phi}{\partial y} \cdot \frac{\partial \phi}{\partial y} \right) + V(\phi) + \frac{K^2}{2}\left[\left|\frac{\partial \phi}{\partial x}\right|^2 \left|\frac{\partial \phi}{\partial y}\right|^2 \right.\right.$$
$$\left.\left. - \left(\frac{\partial \phi}{\partial x} \cdot \frac{\partial \phi}{\partial y} \right)^2 + \left|\frac{\partial \phi}{\partial t}\right|^2 \left(\frac{\partial \phi}{\partial x} \cdot \frac{\partial \phi}{\partial x} + \frac{\partial \phi}{\partial y} \cdot \frac{\partial \phi}{\partial y} \right) - \left(\frac{\partial \phi}{\partial t} \cdot \frac{\partial \phi}{\partial x} \right)^2 - \left(\frac{\partial \phi}{\partial t} \cdot \frac{\partial \phi}{\partial y} \right)^2 \right] \right\}. \tag{2.2}$$

The equations of motion of the model can easily be derived using the Euler Lagrange equations where, to take the constraint $\phi \cdot \phi = 1$ into account, we can use a Lagrange multiplier.

Eliminating this multiplier we find that the equation of motion is given by

$$\frac{\partial^2 \phi}{\partial t^2} = B\big\{ (\mathbb{1} - P)\phi_{ii} - \phi|\phi_t|^2 - (\mathbb{1} - P)\nabla_\phi V(\phi) + K^2\big[-\phi|\phi_t|^2(\phi_i \cdot \phi_i)$$
$$+ (\mathbb{1} - P)\big(2\phi_{it}(\phi_i \cdot \phi_t) + \phi_{ii}(\phi_j \cdot \phi_j) - \phi_{ij}(\phi_i \cdot \phi_j) - |\phi_t|^2(\phi_i \cdot \phi_i)\big)$$
$$+ \phi_t[(\phi_t \cdot \phi_{ii}) - (\phi_i \cdot \phi_{it})] + \phi_i(\phi_j \cdot \phi_{ij}) - \phi_i(\phi_i \cdot \phi_{jj}) - \phi_i(\phi_t \cdot \phi_{it})\big]\big\} \tag{2.3}$$

taken together with the constraint $\phi \cdot \phi = 1$. In (2.3) we have used the notation $f_u = \frac{\partial f}{\partial u}$, $f_{ii} = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$, $f_i g_i = \frac{\partial f}{\partial x}\frac{\partial g}{\partial x} + \frac{\partial f}{\partial y}\frac{\partial g}{\partial y}$, and $f_i g_j h_{ij} = \frac{\partial f}{\partial x}\frac{\partial g}{\partial x}\frac{\partial^2 h}{\partial x^2} + \frac{\partial f}{\partial y}\frac{\partial g}{\partial y}\frac{\partial^2 h}{\partial y^2} + \frac{\partial f}{\partial y}\frac{\partial g}{\partial x}\frac{\partial^2 h}{\partial y \partial x} + \frac{\partial f}{\partial x}\frac{\partial g}{\partial y}\frac{\partial^2 h}{\partial x \partial y}$. Moreover, $P = \phi\phi^t$ and B is the 3×3 matrix given by

$$B = \frac{1}{D}\left([1 + G]\mathbb{1} + (1 + G)K^2\left(\frac{\partial \phi}{\partial x}\frac{\partial \phi^t}{\partial x} + \frac{\partial \phi}{\partial y}\frac{\partial \phi^t}{\partial y} \right) \right.$$
$$\left. + K^4\left(\frac{\partial \phi}{\partial x} \times \frac{\partial \phi}{\partial y} \right)\left(\frac{\partial \phi}{\partial x} \times \frac{\partial \phi}{\partial y} \right)^t \right)$$

$$D = (1 + G)^2 + (1 + G)K^4\left(\left|\frac{\partial \phi}{\partial x}\right|^2 \left|\frac{\partial \phi}{\partial y}\right|^2 - \left(\frac{\partial \phi}{\partial x} \cdot \frac{\partial \phi}{\partial y} \right)^2 \right),$$

with $G = K^2|\phi_i|^2$.

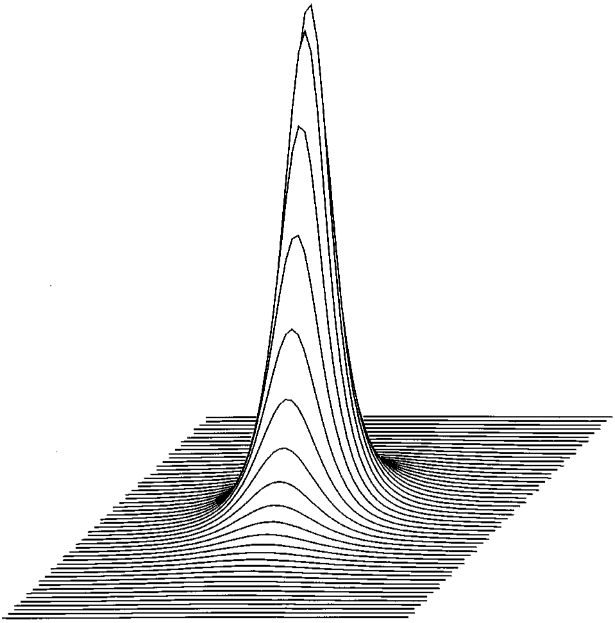

FIG. 1. S^2 soliton energy density: $w = (x + iy)$.

Before concentrating on the methods used to solve these equations, let us say a few words about the known solutions of (2.3). Static, finite energy, solutions of the pure S^2 model ($K = V = 0$) are all known [8, 9]. They are given by holomorphic or anti-holomorphic functions: $w = f(x + iy)$ or $w = f(x - iy)$. To have finite energy solutions the function f must be a rational function and the energy is then proportional to the highest degree in the denominator or the numerator. An important property of these solutions is that they are topologically stable. This means that the solutions cannot decay and can be thus considered as models of solitons. The simplest static solution is given by

$$w = \lambda(x + iy - a), \tag{2.4}$$

where λ and a are 2 arbitrary complex parameters. Rather than looking at the shape of the function w itself it is more appropriate to look at the energy density for the solution. Figure 1 shows this density for (2.4). It can be shown that a in (2.4) fixes the position of the soliton. On the other hand, λ fixes its size. As (2.4) has the same total energy (the volume under the surface in Fig. 1) for all values of λ and a it is easy to check that when λ is large the soliton is spiky and well localised, and when λ is small, the soliton is flat and spread out.

In this paper we will describe the simulations of the model (2.1) for 2 different potentials: $V_H(\phi) = \frac{\theta}{2}(1 + \phi_3)^4$ and $V_B(\phi) = \theta(1 - \phi_3)$, where ϕ_3 denotes the third component of the vector ϕ. The first model has harmonic static solutions, hence the H subscript for the potential. This model was studied in detail in [5] and its simplest static solution is given by $w = \frac{\theta^{1/4}}{K^{1/2}}(x + iy - a)$. The second model was studied in [6, 7] and is usually referred to as the baby Skyrme model. Its static solutions are not known analytically but their construction (for one soliton or many solitons on top of each other) reduces to having to solve an ordinary differential equation whose solutions decay exponentially at infinity [6].

Before we discuss the numerical integration of these models let us present their description in terms of polar angles on the sphere and the stereographic projection of the sphere onto the 2 dimensional plane.

3. THE POLAR ANGLE FORMULATION

Instead of using a three component vector to describe the sphere we can use the polar coordinates on S^2. The advantage of such a description is that it involves two fields instead of three but, on the other hand, we know that the equation of motion has coordinate singularities. To avoid the numerical problems that this generates we can use an idea from differential geometry, namely to divide the sphere into regions and employ a different system of coordinates (maps) on each region. Such maps would have no singularities in their own domains and both of them can be extended to allow for an overlap between the two domains.

Our potentials are functions of ϕ_3 only. This means that the models we are interested in are invariant with respect to rotations around the ϕ_3 axis. To preserve this symmetry in our systems of coordinates we cut the sphere "vertically" into two overlapping bands parallel, respectively, to the (ϕ_1, ϕ_3) and (ϕ_2, ϕ_3) planes and use, respectively, the pairs of angles (θ_1, φ_1) and (θ_2, φ_2) with $0 \leq \theta_{1,2} < \pi$ and $0 \leq \varphi_{1,2} < 2\pi$:

$$\begin{aligned} \phi_1 &= \cos(\theta_1) \\ \phi_2 &= \sin(\theta_1)\cos(\varphi_1) \\ \phi_3 &= \sin(\theta_1)\sin(\varphi_1) \end{aligned} \tag{3.1}$$

$$\begin{aligned} \phi_1 &= -\sin(\theta_2)\cos(\varphi_2) \\ \phi_2 &= \cos(\theta_2) \\ \phi_3 &= \sin(\theta_2)\sin(\varphi_2). \end{aligned} \tag{3.2}$$

When we want to use the two different maps on the same grid, we must be able to transform the coordinates of any point from one map to the other. Assuming that the values of the functions acos and asin lie in the range $[0, \pi]$ and $[-\pi, \pi]$, respectively, we can use the following relations:

$$\begin{aligned} \theta_2 &= \mathrm{acos}(\sin(\theta_1)\cos(\varphi_1)) \\ \theta_1 &= \mathrm{acos}(-\sin(\theta_2)\cos(\varphi_2)) \\ \varphi_2 &= \mathrm{asin}\left(\frac{\sin(\theta_1)\sin(\varphi_1)}{(1-\sin^2(\theta_1)\cos^2(\varphi_1))^{1/2}}\right), \qquad \theta_1 > \frac{\pi}{2} \\ &= \pi - \mathrm{asin}\left(\frac{\sin(\theta_1)\sin(\varphi_1)}{(1-\sin^2(\theta_1)\cos^2(\varphi_1))^{1/2}}\right), \qquad \theta_1 < \frac{\pi}{2} \\ \varphi_1 &= \mathrm{asin}\left(\frac{\sin(\theta_2)\sin(\varphi_1)}{(1-\sin^2(\theta_2)\cos^2(\varphi_2))^{1/2}}\right), \qquad \theta_2 < \frac{\pi}{2} \\ &= \pi - \mathrm{asin}\left(\frac{\sin(\theta_2)\sin(\varphi_2)}{(1-\sin^2(\theta_2)\cos^2(\varphi_2))^{1/2}}\right), \qquad \theta_2 > \frac{\pi}{2}. \end{aligned} \tag{3.3}$$

The Lagrangian density can easily be computed using (2.1) and (3.1) or (3.2). It takes exactly the same form for the two maps,

$$
\begin{aligned}
L = \int dx\, dy \Big\{ & \frac{1}{2}\big(\theta_t^2 - \theta_i^2\big) + \sin^2(\theta)\big(\varphi_t^2 - \varphi_i^2\big)\big) - V(\theta,\phi) \\
& - \frac{K^2}{2}\sin^2(\theta)\big(\theta_t^2\varphi_i^2 + \varphi_t^2\theta_i^2 - \varphi_i^2\theta_j^2 - 2\theta_t\varphi_t\theta_i\varphi_i + (\theta_i\varphi_i)^2\big)\Big\},
\end{aligned}
\tag{3.4}
$$

where for V we take one of the two potentials described before,

$$
\begin{aligned}
V_H(\theta,\varphi) &= \frac{\theta}{2}(1 + \sin(\theta)\sin(\varphi))^4 \\
V_B(\theta,\varphi) &= \theta(1 - \sin(\theta)\sin(\varphi)).
\end{aligned}
\tag{3.5}
$$

The energy is then given by

$$
\begin{aligned}
E = \int dx\, dy \Big\{ & \frac{1}{2}\big(\big(\theta_t^2 + \theta_i^2\big) + \sin^2(\theta)\big(\varphi_t^2 + \varphi_i^2\big)\big) + V(\theta,\phi) \\
& + \frac{K^2}{2}\sin^2(\theta)\big(\theta_t^2\varphi_i^2 + \varphi_t^2\theta_i^2 + \varphi_i^2\theta_j^2 - 2\theta_t\varphi_t\theta_i\varphi_i - (\theta_i\varphi_i)^2\big)\Big\},
\end{aligned}
\tag{3.6}
$$

and the equation of motion, again, takes the same form for the 2 maps,

$$
\begin{aligned}
\theta_{tt} &= \frac{1}{D}\big(\big(1 + K^2\theta_i^2\big)F_\theta + K^2\sin^2(\theta)\theta_i\varphi_i F_\varphi\big) \\
\varphi_{tt} &= \frac{1}{D}\big(\big(1 + K^2\sin^2(\theta)\varphi_i^2\big)F_\varphi + K^2\theta_i\varphi_i F_\theta\big),
\end{aligned}
\tag{3.7}
$$

where

$$
\begin{aligned}
D &= \big(1 + K^2\sin^2(\theta)\varphi_i^2\big)\big(1 + K^2\theta_i^2\big) - K^4(\theta_i\varphi_i)^2\sin^2(\theta), \\
F_\theta &= \theta_{ii} + \sin(\theta)\cos(\theta)\big(\varphi_t^2 - \varphi_i^2\big) - \frac{\partial V}{\partial\theta} + K^2\sin(\theta)\cos(\theta)\big[\theta_i^2\big(\varphi_j^2 - \varphi_t^2\big) - \theta_t^2\phi_i^2 \\
&\quad - (\theta_i\varphi_i)^2\big) + 2\theta_t\varphi_t\theta_i\varphi_i\big] + K^2\sin^2(\theta)\big[\theta_{ii}\big(\varphi_j^2 - \varphi_t^2\big) - \varphi_{ii}(\theta_t\varphi_t - \theta_i\varphi_i) + \varphi_{ij}\theta_i\theta_j \\
&\quad - \theta_{ij}\varphi_i\varphi_j + 2\theta_{ti}\varphi_t\varphi_i - \varphi_{ti}(\theta_t\varphi_i + \theta_i\varphi_t)\big], \\
F_\varphi &= \varphi_{ii} - 2\,\mathrm{cotg}(\theta)(\theta_t\varphi_t - \theta_i\varphi_i) - \frac{1}{\sin^2(\theta)}\frac{\partial V}{\partial\varphi} - K^2\big[\theta_{ii}\big(\theta_j\varphi_j - \theta_t\varphi_t^2\big) \\
&\quad + \varphi_{ii}\big(\theta_t^2 - \theta_i^2\big) - 2\varphi_{ti}\theta_t\theta_i + \theta_{ti}(\theta_t\varphi_i + \theta_i\varphi_t) + \varphi_{ij}\theta_i\theta_j - \theta_{ij}\theta_i\varphi_j\big].
\end{aligned}
$$

4. THE CONFORMAL PROJECTION ONTO C^2

Our third method involves projecting the sphere onto the complex plane using the conformal projection (1.3). The field is then described by the complex field w. The only singularity of w corresponds to the north pole of the sphere: $\phi_3 = 1$. At this point w becomes infinite. We can forsee that this will lead to problems for the numerical integration as, close to this value, w can become arbitrarily large making the numerical method unstable and unreliable. To avoid this problem, we can again use two different maps. This time we can take w and it inverse $u = 1/w$ for, respectively, the southern and the northern hemispheres of S^2. Formally, the Lagrangian and the equation of motion are identical for w and u as long as we define $V(|u|^2) = V(1/|w|^2)$. We have

$$L = \int dx\,dy \Bigg\{ 2\frac{\left(|w_t|^2 - |w_i|^2\right)}{(1+|w|^2)^2} - V(|w|^2) - \frac{K^2}{(1+|w|^2)^4}\big(8|w_t|^2|w_i|^2 \\ - 4|w_i|^2|w_j|^2 - 2(w_t\bar{w}_i + w_i\bar{w}_t)^2 + (w_i\bar{w}_j + w_j\bar{w}_i)^2\big) \Bigg\}, \tag{4.1}$$

$$E = \int dx\,dy \Bigg\{ 2\frac{\left(|w_t|^2 + |w_i|^2\right)}{(1+|w|^2)^2} + V(|w|^2) + \frac{K^2}{(1+|w|^2)^4}\big(8|w_t|^2|w_i|^2 \\ + 4|w_i|^2|w_j|^2 - 2(w_t\bar{w}_i + w_i\bar{w}_t) - (w_i\bar{w}_j + w_j\bar{w}_i)\big) \Bigg\}, \tag{4.2}$$

where

$$\begin{aligned} V_H(|w|^2) &= \frac{\theta}{2}\left(\frac{2|w|^2}{1+|w|^2}\right)^4, \\ V_B(|w|^2) &= \frac{2\theta}{1+|w|^2}. \end{aligned} \tag{4.3}$$

The equation of motion is now given by

$$w_{tt} = \frac{Fa - \bar{F}B}{a^2 - |B|^2}, \tag{4.4}$$

where

$$\begin{aligned} a &= 1 + \frac{2K^2|w_i|^2}{(1+|w|^2)^2}, \\ B &= -\frac{2K^2 w_i^2}{(1+|w|^2)^2}, \\ F &= w_{ii} + 2\left(w_t^2 - w_i^2\right)\frac{\bar{w}}{1+|w|^2} + \frac{4K^2 w}{(1+|w|^2)^3}\big[\bar{w}_i^2 w_j^2 - \bar{w}_t^2 w_i^2 - \bar{w}_t^2 w_i^2 - |w_i|^4 \\ &\quad + 2|w_t|^2|w_i|^2\big] - \frac{2K^2}{(1+|w|^2)^2}\big[\bar{w}_{ii}w_j^2 - \bar{w}_{ii}w_t^2 + w_{ij}w_i\bar{w}_j - w_{ti}(w_t\bar{w}_i + w_i\bar{w}_t) \\ &\quad - \bar{w}_{ij}w_i w_j + 2\bar{w}_{ti}w_t w_i + w_{ii}\left(|w_t|^2 - |w_j|^2\right)\big] - \frac{w}{2}(1+|w|^2)^2\frac{\partial V}{\partial |w|^2}. \end{aligned}$$

Next we will use the methods we have just described to solve (2.3) and to compare their results. In each case, we have to solve a set of second order partial differential equations. To perform the time integration of the equations we will use both the 4th order Runge–Kutta and the leapfrog methods. We will compare both methods and will describe their advantages and disadvantages. To evaluate the spatial derivatives we will use finite difference operators. The equations we want to solve are nonlinear in the derivatives but they are invariant under spatial rotations. To take advantage of this property we will introduce, what we will call, the isotropic nonlinear finite difference operators. Then we will describe how one can reduce the problems associated with the numerical integration of (2.3) and finally, we will compare the results produced by the different methods.

5. NONLINEAR FINITE DIFFERENCE OPERATORS

To evaluate the spatial derivatives we use finite difference operators. It is easy to note that (2.1) is rotationally invariant and, as a consequence, all the nonlinear terms involving derivatives in the equations of motion (2.3), (3.7), and (4.4), are of 3 different types: the Laplacian $\Delta(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$, scalar products of gradients $f_i g_i = \frac{\partial f}{\partial x}\frac{\partial g}{\partial x} + \frac{\partial f}{\partial y}\frac{\partial g}{\partial y}$, and operators of the form $f_i g_j h_{ij}$ where the sums over i and j are implicit. The simplest method to evaluate these expressions involves computing each derivative separately and then combining them together to form the desired operator. For example, to evaluate the Laplacian, we can compute $\frac{\partial^2 f}{\partial x^2}$ and $\frac{\partial^2 f}{\partial y^2}$ and then add them together. If we label each point of the two dimensional regular square grid with 2 indices so that $f(x, y) = f(n, m)$, and call dx the distance between two adjacent lattice points, we have $\frac{\partial^2 f}{\partial x^2}(x, y) = 1/dx^2(f(n+1, m) + f(n-1, m) - 2f(n, m)) + O(dx^2)$ and $\frac{\partial^2 f}{\partial y^2}(x, y) = 1/dx^2(f(n, m+1) + f(n, m-1) - 2f(n, m)) + O(dx^2)$ leading to the well known 5 point Laplacian.

Having performed some simulations we have found that we cannot use the 5 point Laplacian. The grid effects are strong and destroy the rotational symmetry of various quantities, such as the energy or the topological charge densities. This, in turn, affects the evolution of the solitons. The effects are small but non-negligible. So, we have to go beyond the 5 point Laplacian. However, there are many finite difference operators which in the limit of small lattice spacings converge to the Laplace operator. An obvious improvement on the 5 point Laplacian will involve a 9 point one, i.e., a laplacian of the type

$$\begin{aligned} DL_{gen.9}(f) = (-(4a+4b)\,f_{i,j} + a(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}) \\ + b(f_{i+1,j+1} + f_{i-1,j+1} + f_{i+1,j-1} + f_{i-1,j-1}))\frac{1}{(a+2b)\,dx^2} \end{aligned} \tag{5.1}$$

for a reasonable choice of a and b. By trial and error we have found that b has to be a reasonable fraction of a (somewhere between 0.1 and 0.7). Of these $a = 4$, $b = 1$ is particularly convenient, *i.e.*,

$$\begin{aligned} DL_9(f) = (-20\,f_{i,j} + 4(f_{i+1,j} + f_{i-1,j} + f_{i,j+1} + f_{i,j-1}) + f_{i+1,j+1} + f_{i-1,j+1} \\ + f_{i+1,j-1} + f_{i-1,j-1})\frac{1}{6\,dx^2}, \end{aligned} \tag{5.2}$$

as then its first correction term, like the Laplacian itself, is also invariant under rotations (isotropic) and vanishes for harmonic functions:

$$DL_9(f) = \Delta(f) + \frac{dx^2}{12}\Delta^2(f) + O(dx^4).$$

So, up to the first correction term we see that $DL_9(f)$ does not break the symmetries of (2.3). Moreover, when f is a harmonic function, like the static solution (2.4), the first correction term vanishes.

Is it possible to derive nonlinear differential operators with similar properties for the other 2 types of operators we have to evaluate? To evaluate the scalar product of gradients the simplest approach would involve evaluating the first derivatives $\frac{\partial f}{\partial x}$ and $\frac{\partial f}{\partial y}$, using say, the symmetric difference operators and then using these results to calculate the scalar product. However, instead of computing this operator along the lines of the lattice, we can also evaluate it along the diagonals or even better take a linear combination of the transverse and diagonal operators with the appropriate coefficients.

Defining the diagonal coordinates $u = x + y$ and $v = x - y$, and introducing the following finite difference operators on the grid,

$$\begin{aligned} D_x f &= \frac{1}{2\,dx}(f(n+1,m) - f(n-1,m)),\\ D_y f &= \frac{1}{2\,dx}(f(n,m+1) - f(n,m-1)),\\ D_u f &= \frac{1}{4\,dx}(f(n+1,m+1) - f(n-1,m-1)),\\ D_v f &= \frac{1}{4\,dx}(f(n+1,m-1) - f(n-1,m+1)) \end{aligned} \tag{5.3}$$

we find that a good choice corresponds to

$$\begin{aligned} DG(f,g) &= \frac{2}{3}(D_x f D_x g + D_y f D_y g + D_u f D_u g + D_v f D_v g)\\ &= \frac{\partial f}{\partial x}\frac{\partial g}{\partial x} + \frac{\partial f}{\partial y}\frac{\partial g}{\partial y} + dx^2(f_x \Delta g_x + f_y \Delta g_y + g_x \Delta f_x + g_y \Delta f_y) + O(dx^4) \end{aligned}$$

which is isotropic and harmonic up to the second order in dx.

For the third type of operators we can follow a similar method. First we define the second order finite difference operators

$$\begin{aligned} D_{xx} f &= \frac{1}{dx^2}(f(n+1,m) + f(n-1,m) - 2f(n,m)),\\ D_{yy} f &= \frac{1}{dx^2}(f(n,m+1) + f(n,m-1) - 2f(n,m)),\\ D_{xy} f &= \frac{1}{4\,dx^2}(f(n+1,m+1) + f(n-1,m-1)\\ &\quad - f(n+1,m-1) - f(n-1,m+1)),\\ D_{uu} f &= \frac{1}{4\,dx^2}(f(n+1,m+1) + f(n-1,m-1) - 2f(n,m)),\\ D_{vv} f &= \frac{1}{4\,dx^2}(f(n+1,m-1) + f(n-1,m+1) - 2f(n,m)),\\ D_{uv} f &= \frac{1}{4\,dx^2}(f(n+1,m) + f(n-1,m) - f(n,m+1) - f(n,m-1)). \end{aligned} \tag{5.4}$$

and find that

$$\begin{aligned} DT(f,g,h) = {}& \frac{2}{3}(D_x f D_x g D_{xx} h + D_y f D_y g D_{yy} h + (D_x f D_y g + D_y f D_x g) D_{xy} h\\ &+ 2(D_u f D_u g D_{uu} h + D_v f D_v g D_{vv} h + (D_u f D_v g + D_v f D_u g) D_{uv} h))\\ = {}& f_i g_j h_{ij} + \frac{dx^2}{12}(f_k g_l h_{iikl} + (f_x g_y + f_y g_x) h_{xyii})\\ &+ \frac{dx^2}{6}(f_i h_{ij} g_{kkj} + g_i h_{ij} f_{kkj}). \end{aligned}$$

For this choice we see that the first correction term also vanishes for harmonic functions but, this time, the extra term is not invariant under rotations.

6. IMPLEMENTING THE INTEGRATION METHODS

Using ϕ

The easiest way to study the time evolution of extended structures of the S^2 sigma model involves using the ϕ formulation. This is because this formulation does not require the use of different maps to describe the field. If we look at (2.3), we see that it is given by a set of 3 second order hyperbolic equations which can be solved as an initial value problem. For the time evolution we have used the 4th order Runge–Kutta or the leapfrog methods. Both are simple to implement and we will compare their relative accuracy. The main difficulty with the ϕ formulation is to keep the length of the vector ϕ equal to 1 (to satisfy (1.1)). If we start with an initial condition $\phi(t=0)$ and $\frac{\partial\phi}{\partial t}(t=0)$ satisfying (1.1) and (1.2), we can hope that the time integration procedure will preserve this property. In practice it nearly does and after a few integration steps the normalisation of the field is still close to unity. However, as the equation of motion holds only when the ϕ field is properly normalised this small error quickly exponentiates and the numerical procedure becomes unreliable; in practice ignoring the constraint introduces instability into the method and it does not take long before $(\phi\cdot\phi)$ explodes. A simple solution of this problem is to normalise the field ϕ to 1 and project $\frac{\partial\phi}{\partial t}$ onto the plane orthogonal to ϕ after every integration steps. This method works reasonably well and was used in [5] and some other works [6, 7].

Another method involves keeping the Lagrange multiplier (which imposes the constraint $\phi\cdot\phi=1$) in the equation of motion and arranging for its effect to be such that the constraint is satisfied. This corresponds to projecting the vector ϕ onto the sphere but using a different projection scheme.

As we will see, our method of dealing with the constraint can be improved further and some of the methods we analyse here will give more accurate results. For some problems the accuracy of the normalisation method is sufficient; some others will require the use of our more sophisticated approaches.

To evaluate the spatial derivatives we can use finite difference operators, evaluating the derivatives of each component of ϕ independently. This simple approach has the disadvantage that it ignores the relations that exist between the different components of ϕ because of (1.1). Indeed, one way to derive finite difference operators is to expand the fields in a power series and match the obtained expressions with the values taken by the field on the lattice. Thus we could put, for the field close to some lattice point,

$$\Psi = a + bx + cy + dx^2 + ey^2 + fxy + \dots,$$

where a, b, c, d, e, and f are vectors whose values are fixed by the value of the field at it and some adjacent lattice points. Performing such a construction for each lattice point we see that our fields are correctly normalised at each point of the lattice used for its construction, but not in between them. Thus (1.1) and (1.2) are not satisfied everywhere in the plane. This can be put right by defining $\phi = \Psi/(\Psi\cdot\Psi)^{1/2}$. Then ϕ is correctly normalised everywhere and all its derivatives can be expressed in terms of the derivatives of Ψ which can be evaluated numerically using the expressions described in the previous sections.

For example, to evaluate $\frac{\partial\phi}{\partial u}$, we note that

$$\frac{\partial\phi}{\partial u} = \frac{1}{|\Psi|}\left(1 - \frac{\Psi\Psi^t}{|\Psi|^2}\right)\frac{\partial\Psi}{\partial u} = \frac{1}{|\Psi|}(1-P)\frac{\partial\Psi}{\partial u},$$

where P is the projector onto the one dimensional sub-space spanned by Ψ. Then $\frac{\partial\Psi}{\partial u}$ can be evaluated using the appropriate finite difference operator $D_u(\Psi)$ and exploiting the fact that at each lattice point ϕ and Ψ take identical values we have

$$\frac{\partial\phi}{\partial u} = (1-\phi\phi^t)D_u(\Psi) = (1-\phi\phi^t)D_u(\phi). \tag{6.1}$$

Similarly, for the second order derivatives operators, we can derive the following expression

$$\begin{aligned}\frac{\partial^2\phi}{\partial u\partial v} = {} & D_{uv}\phi - D_u\phi(\phi\cdot D_v\phi) - D_v\phi(\phi\cdot D_u\phi) - \phi[(D_u\phi\cdot D_v\phi)\\ & + (\phi\cdot D_{uv}\phi) - 3(\phi\cdot D_u\phi)(\phi\cdot D_v\phi)].\end{aligned} \tag{6.2}$$

This result can also be obtained by putting

$$\frac{\partial^2\phi}{\partial u\partial v} = D_{uv}\phi - \alpha D_u\phi - \beta D_v\phi - \gamma\phi$$

and using

$$\phi\cdot\frac{\partial^2\phi}{\partial u\partial v} + \frac{\partial\phi}{\partial u}\cdot\frac{\partial\phi}{\partial v} = 0,$$

which follows from the normalisation of ϕ, to derive the expressions for α, β, and γ.

Using the Polar Angles

When using the polar angles the main difficulty stems from having to use two different maps to describe the field. To do this we have to store, at every lattice point, the values of the fields θ and φ (and their time derivatives) as well as the information as to which map is used at this point (we can call them maps 1 and 2 and store the index of the map).

The finite difference operators that we use involve only the 8 nearest neighbours of a given lattice point. To evaluate the finite difference operators at a given point on the lattice we must make sure that the fields at these neighbouring points are expressed using the same map as the map of the point in question. As the grid is scanned one lattice point after another we make a copy of the fields for the 8 neighbouring points and convert them, if required, to the map being used for the current lattice point. Then together with the field at the “central” point, the 8 converted fields form a small 3 × 3 subgrid which can be used to compute the finite difference operators at this point.

Notice that the two maps will be used in two separate regions of the lattice, and that the “conversion” between the two maps will have to be performed only in the neighbourhood of the curve which separates the two regions. On the other hand, this border curve will move with time and so the program has to check at each time step where on the grid the “conversion” is required.

There is a further problem that arises as φ is defined modulo 2π. To compute the difference between 2 values of φ we must take the value modulo 2π. Thus, for example, the difference between 6.1 and 0.1 is $6.1 - 2\pi - 0.1 \approx -0.28318$ and not 6.

When the values of both θ and φ have been updated, during the time integration, one must check that their new values are still inside the correct range and, if they are not, perform the appropriate transformation. The most common cases of problems arise when φ becomes larger than 2π or smaller than 0. We must then subtract or add, respectively, 2π to the original value.

After each time integration step we must also check, at each lattice point, that $\pi/2 \leq \theta \leq 3\pi/2$. When this condition is not satisfied we must use (3.1) or (3.2) to transform the field from one map to the other.

The 4th order Runge–Kutta method involves 4 successive sub-steps and we must make sure that the same map is used in all substeps for a given lattice point. It is compulsory to use the same map in all 4 sub-steps as the last part of the Runge–Kutta method involves adding a linear combination of the 4 terms computed in each substep. This only makes sense if they have all been evaluated in the same map. We can thus only change the map after each time step.

For the leapfrog method there is a further problem. The integration procedure can be summarised as

$$f(t + dt/2) = f(t - dt/2) + dt * F(t), \tag{6.3}$$

where $f(t)$ represents the fields at time t and $F(t)$, the right hand side of the equation, is evaluated with the fields at time t. As, at some points of the grid, $f(t)$ and $f(t-dt/2)$ may use different maps we must convert $f(t-dt/2)$ to the same map as $f(t)$ before we evaluate (6.3) (we could also convert $f(t)$ before computing $F(t)$ but this is less appropriate).

Using the Conformal Projection

For the description in terms of the complex field w we introduce two different maps, w and $1/w$, and at each point we use the map which satisfies the condition $|w|^2 \leq 1$. The implementation is similar to the one described for polar angles. Again, at each lattice point, we have to build a 3×3 local grid and convert the eight neighbouring points to the same map.

7. NUMERICAL COMPARISONS

We have performed many comparisons of different methods of integration applied to different models; in particular, we have looked at

—the S^2 model
—the holomorphic Skyrme model
—the baby Skyrme model.

For each of these models we have used 6 different methods of simulation. As we will refer to them inside the tables describing our results we have chosen a short name, given in boldface in the list below:

—Phi field (**Phi**).
—Phi field with curvature corrections (**Phi-Cor**).

—W field (**W**).
—W field with nonlinear finite difference operators (**W-NLFDO**).
—Polar angles field (**Polar**).
—Polar angles field with nonlinear finite difference operators (**Polar-NLFDO**).

We have also compared the 4th order Runge–Kutta and the leapfrog methods. For all the simulations presented here we have used fixed boundary conditions. This was to make the comparison between the different methods easier. Otherwise, we usually put some absorption at the edge of the grid to absorb radiation.

We present our results by looking at each model separately and comparing the different methods of integration of the fields of this model. The tests we have performed in each case are quite similar. First we looked at the field of one static soliton. Its analytical form is known in the case of the S^2 model and the holomorphic Skyrme model. For the baby Skyrme model we have computed it numerically, by solving the appropriate ordinary differential equation. The field of such a static soliton should be a solution of the equation of motion. However, in numerical simulations it does not solve it exactly; hence we can study its evolution during a certain length of time and then measure how much the configuration has changed during this time.

Another simple test consists in boosting the soliton. As the models are all Lorentz covariant, if $F(x, y)$ is a static solution of the equation, then so is the boosted solution $F(\gamma(x - vt), y)$ where $\gamma = 1/(1 - v^2)^{1/2}$.

Finally, we can perform a scattering of solitons. However, this time we do not have any analytical solutions with which to compare the fields obtained in our numerical simulations.

To quantify the quality of our integration we can look at several different parameters. First of all, we can monitor the conservation of energy and/or the topological charge. It is worth pointing out at this stage that we have discretised the equation of motion and the energy density separately. This implies that our discretised energy (*i.e.*, our approximation to the total energy) is not a conserved quantity for our set of discrete equations. Of course, as it approximates the conserved energy its variation with time is small. Thus, when we look at the time dependence of this energy we do not monitor just the quality of the time integration for our system of ODEs but, rather, evaluate the overall quality of integration for our PDE.

As mentioned before, when we use the vector ϕ, we have to normalise this vector, at least, after every few steps of time integration. As we scan the grid to normalise ϕ we find it useful to compute the maximum of $\delta|\phi|^2 = ||\phi|^2 - 1|$ (Max$_{grid}$ $(\delta|\phi|^2)$), over the grid and to monitor its evolution as a function of time. Our experience has taught us that when this quantity becomes too large this is usually a sign that something is wrong with the integration. We can only compute Max$_{grid}$ $(\delta|\phi|^2)$, for the ϕ field so this quantity is not very useful when comparing the different methods of integration. It is nevertheless a useful quantity to monitor in the ϕ formulation when the fields change a lot, so we will give an example of it when we analyse a scattering of solitons.

In some cases we know the exact expression for the field $\phi_e(x, y, t)$ at the time t. We can thus measure the quality of integration by calculating the difference between the exact configuration ϕ_e and the one obtained numerically ϕ_n. For this we can use the expression

$$\Delta\phi = \int dx\, dy |\phi_e - \phi_n|. \tag{7.1}$$

As the vector ϕ lies on a sphere, if ϕ_e and ϕ_n are completely uncorrelated, then $|\phi_e - \phi_n|^2$ equals 2, on average, and $\Delta\phi$ is given by the area over which the integration is performed (multiplied by $\sqrt{2}$).

As we are only interested in the field close to the Skyrmion we should compare $\Delta\phi$ to the area covered by the Skyrmion. In the example we have chosen and with our choice of parameters, the radius of our Skyrmion is about 5 units; thus $\Delta\phi$ should be compared to 115. We can also scan the grid and look at the largest difference between the two fields $\text{Max}\,\delta\,\phi = \text{Max}_{grid}\,|\phi_e - \phi_n|$. This will give us an upper bound on the error in the evaluation of the field.

When we do not know the exact solution, we can compare the fields by evaluating $\Delta\phi$ for any two fields obtained by two different methods. In practice, we will choose two methods which we believe to give the most accurate answers and then compare all the field configurations to the fields obtained with these two methods.

The S^2 Model

As we know the analytic form of the static solution of (2.3), the first test we have performed involved taking such a solution and checking to what extent it satisfied also the numerical equation. We have already mentioned that the soliton solutions of this model are unstable and that the smallest perturbation can make them shrink or expand. This implies that small perturbations, caused by numerical errors, are likely to make the soliton expand or blow up unless the method of integration is extremely accurate. Integrating the time evolution of this model is thus very difficult because of the genuine instability of the solitons.

What we have found is that the most accurate method of integrating the S^2 model is to use the complex field w together with the nonlinear finite difference operators. When we used the 9 point Laplacian (5.2) and the field ϕ without the curvature corrections the soliton blew up and with the other methods, the soliton oscillated in size. Table I summarises our observations. The first column presents the error in the conservation of energy. The second one shows the amplitude of oscillation, as a fraction of the total size, of the maximum of the energy density of the soliton. The last column gives the maximum kinetic energy of the soliton. This last quantity measures the inaccuracy of our scheme as this energy is entirely due to the fact that our initial condition does not satisfy our discretised equation and, as a result, the soliton oscillates and produces some waves. The energy of the soliton, in our units, is 1. We do not present the results of the comparison of the integrated field

TABLE I
Static S^2 Soliton on a 201 × 201 Grid, Using the 4th Order Runge–Kutta Method

Method	ΔE	Oscillation	Kin. En.
Phi-Cor	$6.5\ 10^{-4}$	0.19	$6\ 10^{-5}$
W	10^{-5}	$3.2\ 10^{-4}$	10^{-6}
W-NLFDO	$3\ 10^{-10}$	$6.5\ 10^{-6}$	$5\ 10^{-14}$
Polar	$6.5\ 10^{-5}$	$5\ 10^{-2}$	$2.5\ 10^{-6}$
Polar-NLFDO	$1.7\ 10^{-4}$	$6.5\ 10^{-2}$	$4.5\ 10^{-6}$

Note. $E = 1$.

with the exact solution as this depends too much on the value of the time at which the comparison is made. Let us just mention that for the W-NLFDO method, $\Delta\phi = 6.7\,10^{-5}$ and that the largest difference on the grid between the integrated field and the exact solution was Max $\delta\phi = 1.3\,10^{-6}$, thus showing the amazing accuracy of this method. In Table I we do not include any values for the Phi methods as in these cases the soliton blew up.

We see from Table I that the W-NLFDO is by far the most accurate method to integrate this model, followed by the W method and the two methods which use the polar angles. When we used the vector ϕ with corrections the results did improve and the soliton did not blow up, but this method was still substantialy less accurate than the other four.

Given the fact that the S^2 model is so intricate to integrate, we will not compare the different methods of integration for a boosted soliton and leave this comparison for the Skyrme models.

The Holomorphic Skyrme Model

To evaluate the accuracy of the integration of the holomorphic Skyrme model we have chosen the following parameter values, $K^2 = \theta = 0.2$ and we have performed the simulations on a 201×201 grid ranging from -10 to 10 in both directions and on a 401×401 grid ranging from -20 to 20 (hence $dx = dy = 0.1$). The static solution is then simply given by $w = x + iy$.

In our first test we have taken the static solution at rest at the centre of the grid as the initial condition and have integrated the equation up to $t = 100$; we have then compared the final configuration with the initial one. The results are given in Table II.

Once again, W-NLFDO is the most accurate while Phi the least accurate method. W and Polar-NLFDO are second best while Phi-Cor and Polar are very comparable. Note also that the error varies between 2% (Phi) and 0.001% (W-NLFDO).

For the second test, we have placed the soliton on the grid starting at $\mathbf{x} = (-1, 0)$ and boosted it across the grid with the speed $v = 0.2$. After integrating the equation up to $t = 10$ the final position of the soliton was found to be, as expected, quite close to $\mathbf{x} = (0, 1)$. It was not exactly at this value—but this was due to the tail of the soliton (which is cut off by the finite size of the grids used in the numerical procedures). This inacurracy has little to do with the method; it depends much more on the size of the lattice. To assess the quality of the integration, we have evaluated the difference between the analytical expression for the displaced Skyrmion and the fields obtained numerically. The results are presented in

TABLE II
Static Holomorphic Skyrmion on a 201 × 201 Grid Using the 4th Order Runge–Kutta Method

Method	$\Delta\phi$	Max $\delta\phi$	ΔE
Phi	1.2	0.017	$5\,10^{-5}$
Phi-Cor	0.14	0.0025	$2\,10^{-5}$
W	0.035	0.001	10^{-5}
W-NLFDO	0.00014	$2.8\,10^{-6}$	10^{-8}
Polar	0.22	0.0042	10^{-5}
Polar-NLFDO	0.05	0.0013	$7\,10^{-6}$

Note. $E = 1.125$.

TABLE III
Boosted Holomorphic Skyrmion on a 401 × 401 Grid Using the 4th Order Runge–Kutta Method

Method	$\Delta\phi$	Max $\delta\phi$	ΔE	$\Delta_W\phi$	Max $\delta_W\phi$	$\Delta_\phi\phi$	Max $\delta_\Phi\phi$
Phi	7.9	0.03	$1.8\ 10^{-4}$	2.3	0.031	2.05	0.03
Phi-Cor	5.8	0.01	$7\ 10^{-5}$	0.25	0.0036	—	—
W	5.6	0.01	$2.3\ 10^{-5}$	0.038	0.0011	0.27	0.0036
W-NLFDO	5.6	0.01	$8\ 10^{-7}$	—	—	0.25	0.0036
Polar	6.3	0.036	$7.5\ 10^{-4}$	0.72	0.036	0.55	0.035
Polar-NLFDO	5.9	0.022	$3.5\ 10^{-4}$	0.33	0.022	0.33	0.021

Note. $E = 1.149$.

Table III. All the obtained results appear to be very comparable. However, this is, again, mainly due to the fact that we have performed our integration on a finite grid and that the tail of the soliton, which extends beyond the edge of the grid, has been slowing down the soliton progressively and as a result, the soliton did not make it all the way to its final position. Thus we are comparing two field configurations which are slightly displaced with respect to each other.

To perform a more meaningful comparison of the different methods we have decided to compute the relative differences between the fields obtained in different numerical integrations. We have taken as the reference fields the fields obtained with the W-NLFDO and the Phi-Cor methods and labelled $\Delta\phi$ and Max $\delta\phi$ with indices W and ϕ, respectively, when the differences were evaluated with respect to these fields.

Looking at the table we see that Phi gives the least accurate results while, this time, W, W-NLFDO, and Phi-Cor produce the most accurate results and that the two methods involving the polar angles are comparable in terms of their accuracy. When we compare the different fields after integration, the maximum error varies between 1% and 4%.

The Baby Skyrme Model

To study the accuracy of the integration of the baby Skyrme model we have chosen the following values of the parameters: $K^2 = 1$, $\theta = 0.1$. Moreover, we have performed the simulations on a 201 × 201 grid ranging from -10 to 10 in both directions or on a 401 × 401 grid ranging from -20 to 20 (hence $dx = dy = 0.1$). The static solutions were obtained by solving an ordinary differential equation [6].

In our first test we have taken as our initial condition the static solution at rest at the centre of the grid and have integrated the equation up to $t = 100$. Then, for the second test, we have placed the soliton on the grid at $\mathbf{x} = (-1, 0)$ and boosted it across the grid with the speed $v = 0.2$. The comparison of the results obtained in different methods is summarised in Tables IV and V.

Looking at the tables we observe that, for this model, the difference between the different methods is far less striking and it is not clear how to identify the best one. Using one of the W methods looks like the best choice for a moving Skyrmion while these methods seem to be the least accurate ones for the static fields. In any case, the differences between all the 6 methods are relatively small as their accuracies differ by only a factor of 3 or 4 and the differences between the methods are always below 1%.

TABLE IV
Static Baby Skyrmion on a 201 × 201 Grid Using the 4th Order Runge–Kutta Method

Method	$\Delta\phi$	Max $\delta\phi$	ΔE
Phi	0.18	0.0035	$9\ 10^{-6}$
Phi-Cor	0.29	0.0017	10^{-5}
W	0.57	0.0054	$2\ 10^{-5}$
W-NLFDO	0.59	0.0044	10^{-5}
Polar	0.39	0.0039	$5\ 10^{-6}$
Polar-NLFDO	0.44	0.005	$8\ 10^{-6}$

Note. $E = 1.565$.

Finally we have compared the results of simulating the scattering of two baby Skyrmions. We have looked at a head-on collision between two Skyrmions located initially at $x = -10$ and $x = 10$ each sent towards the centre of the grid with the speed $v = 0.5$. We have used a 401 × 401 grid extending from -20 in 20 in both directions.

In all simulations the two Skyrmions moved towards each other, collided just before $t = 20$, then overlapped and came out of their interaction region at 90 degrees along the y axis and with a speed at about half of its initial value (such a collision is highly non-elastic).

As there are 2 Skyrmions, the “reference” value of $\Delta\phi$ is 230. We see from Table VI that in the worst case we have an overall accuracy of about 10^{-4} and that the largest difference between 2 vectors on the grid is 0.002. The least accurate method seems to be Phi without the curvature corrections, while both the polar angle methods are quite accurate. Phi-Cor and W-NLFDO have a large overall difference between them despite the fact that the largest difference between the two vectors is only 0.0065. This suggests that the difference is, most probably, due to an overall displacement between the 2 fields.

We have mentioned before that, when using ϕ, it is useful to look at how much the field ϕ must be normalised after each time integration step. In Fig. 2 we show $\max_{grid} \delta|\phi|$ obtained with the Polar-NLFDO method for a 2 Skyrmion scattering. The peak just before $t = 20$ corresponds to the time when the 2 solitons overlap, *i.e.*, “are on top of each other.” In some simulations, $\max_{grid} \delta|\phi|$ can suddenly change by a few orders of magnitudes. This is usually a sign that the grid is too coarse for the simulation and that a finer mesh should be used.

TABLE V
Boosted Baby Skyrmion on a 401 × 401 Grid Using the 4th Order Runge–Kutta Method

Method	$\Delta\phi$	Max $\delta\phi$	ΔE	$\Delta_W\phi$	Max $\delta_W\phi$	$\Delta_\phi\phi$	Max $\delta_\Phi\phi$
Phi	1.37	5.9 10−3	3 10−4	0.77	0.0092	0.4	0.005
Phi-Cor	1.6	2.2 10−3	$3\ 10^{-5}$	0.37	0.0042	—	—
W	1.92	4.1 10−3	$1.3\ 10^{-5}$	0.057	0.001	0.34	0.004
W-NLFDO	1.96	3.9 10−4	$1.5\ 10^{-5}$	—	—	0.37	0.0042
Polar	1.7	5 10−3	$3\ 10^{-4}$	0.32	0.0047	0.23	0.0062
Polar-NLFDO	1.73	3 10−3	$3\ 10^{-5}$	0.24	0.0025	0.18	0.0039

Note. $E = 1.598$.

TABLE VI
Two Baby Skyrmions on a 401 × 401 Grid Using the 4th Order Runge–Kutta Method

Method	ΔE	$\Delta_W \phi$	Max $\delta_W \phi$	$\Delta_\phi \phi$	Max $\delta_\Phi \phi$
Phi	$6\ 10^{-3}$	8.6	0.044	5.7	0.04
Phi-Cor	$9\ 10^{-3}$	3.3	0.0065	—	—
W	$9\ 10^{-3}$	1.9	0.023	3.6	0.025
W-NLFDO	$1.2\ 10^{-2}$	—	—	3.3	0.0065
Polar	$6\ 10^{-3}$	1.4	0.0038	2.3	0.0046
Polar-NLFDO	$6\ 10^{-3}$	1.5	0.006	2.1	0.0067

Note. $E = 3.616$.

Further Tests

In the last sections we have given results of simulations obtained using the 4th order Runge–Kutta method for the time integration. We have also repeated some of these simulations using the leapfrog method and have seen essentially no difference in terms of the final results and accuracy except in the case of the polar angle methods. In this last case the leapfrog method appeared to be unstable and the simulations have always blown up.

We have also repeated some of the simulations for the two Skyrme models using grids with 4 times as many points—keeping the "physical" dimension of the lattice unchanged and so decreasing dx and dt by a factor of 2. Overall, the effects have been the same for most methods of integration: the errors decreased by a factor of about 20 and this was observed for both the 4th order Runge–Kutta method and the leapfrog method (with a few exceptions for the leapfrog method).

When boosting solitons on a lattice, it is natural to make them travel along one of the lattice's main axis. This may bias the results of the simulations. To check that we do not

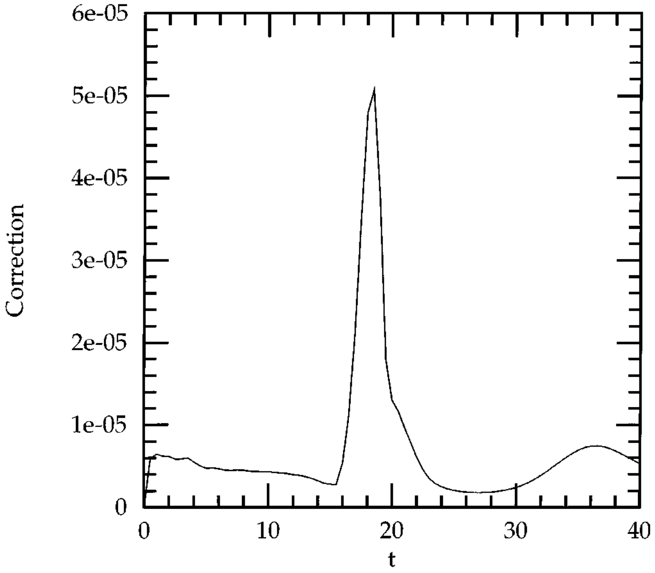

FIG. 2. Baby Skyrmion scattering: $\max_{grid} \delta|\phi|$ for the Phi-Cor method.

TABLE VII
Integration Time (s) for 100 Time Steps on a 201 × 201 Grid

Method	RK4	Leapfrog
Phi	133	70
Phi-Cor	139	73
W	145	72.5
W-NLFDO	159	80
Polar	280	—
Polar-NLFDO	302	—

have such a bias we have boosted our Skyrmions along different directions and have found no difference for either of the Skyrme models. We also verified that the 90 degree scattering had taken place in all these cases.

We should also say a few words about the resources needed to implement the various methods of our tests. When using the real vector ϕ we need 50% more storage space than with the complex field w or the polar angles. When using multiple maps, we also need an extra array to store a number telling us which map is used, but this can be done using an array of characters (or even a single bit per lattice site) and thus it requires a negligible amount of extra computer memory. To implement the Runge–Kutta method we need 3 copies of the fields while the leapfrog method requires only 2 copies. Thus the most economical method, in terms of storage requirement, is the leapfrog method for the w fields. This combination requires only just over 50% of the memory needed to use the Runge–Kutta with the ϕ field.

In terms of speed, the leapfrog method is, as expected, about 2 times faster than the Runge–Kutta method. Using multi-map fields also is slower than using the ϕ fields. This is because the extra time required for checking and converting the fields between the different maps is larger than the time gained by having to deal with only 2 fields instead of 3.

In Table VII we report the time required to perform 100 integration steps on a 201 × 201 grid for the Skyrme models. To increase the reliability of our work, we had implemented in a single program the various methods of integration for the 3 models we analysed. Our program thus had to perform many context switches and unecessarily large loops which could affect the speed of the code. Using a single program for any specific method would most probably result in a better performance than the ones given below. Our tests were performed on a sparc Ultra 2 300 Mhz fitted with 1 GB of RAM.

8. FURTHER METHODS

There are other methods of integrating the time evolution of models valued on the sphere S^2 or some other manifolds. One of the first numerical integrations [10] of the $(2+1)$ dimensional $S^2\sigma$ model was performed by first discretising the action of the model and then deriving the equation of motion from it. This method has some advantages like giving an expression for the total energy that is automaticaly conserved. Unfortunately it is more difficult to apply this method to models containing higher order terms like the Skyrme term.

We have also tested other methods of integration. First of all, we have used a 25 point Laplacian for the S^2 model using the real vector ϕ. In this case the Laplacian is proportional to the expression

$$\begin{aligned}\Delta\phi(i,j) \sim \{&a[\phi(i+1,j+1)+\phi(i-1,j-1)+\phi(i+1,j-1)+\phi(i-1,j+1)]\\ &+b[\phi(i+1,j)+\phi(i-1,j)+\phi(i,j+1)+\phi(i,j-1)]\\ &+c[\phi(i+2,j)+\phi(i-2,j)+\phi(i,j-2)+\phi(i,j+2)]\\ &+d[\phi(i+1,j+2)+\phi(i+1,j-2)+\phi(i-1,j+2)+\phi(i-1,j-2)\\ &+\phi(i+2,j+1)+\phi(i+2,j-1)+\phi(i-2,j+1)+\phi(i-2,j-1)]\\ &+e[\phi(i+2,j+2)+\phi(i-2,j+2)+\phi(i+2,j-2)+\phi(i-2,j-2)]\\ &-4(a+b+c+2d+e)\phi(i,j)\}\frac{1}{dx^2}. \qquad (8.1)\end{aligned}$$

Different choices of the parameters $a, \ldots, e$ give different Laplacians. Note that our 9 point Laplacian corresponds to $c = d = e = 0$. A convenient choice is provided by taking $a = 68/27$, $b = -58/81$, $c = -8/27$, $d = 10/81$, and $e = -11/648$ for, as shown by Rutenberg [11], such a choice of "lattice" Laplacian has a vanishing k^4 term in its Fourier transform and, in addition, its k^6 and k^8 average to zero. We have performed some tests with this choice of Laplacian and have found that, indeed, it introduced very little perturbation of a one Skyrmion field; hence a soliton of the pure S^2 model changed its size very little and so did not blow up in any of our simulations. However, the price to be paid for the use of this Laplacian is the time needed for the simulation to be completed (when compared with the simulations which used a nine point Laplacian, the 25 point ones required about 2.5 times more computer time to perform the same integration).

All simulations reported in this paper involved square, equally spaced, grids with $dx = dy$. We have also looked at irregular, in general rectangular, grids and a hexagonal grid. In the first case, to eliminate the effects of the boundaries we mapped R^2 into a regular grid in (z_1, z_2) by $z_1 = \frac{\alpha x}{1+\alpha|x|}$ (for some choice of α and with $(-1 < z_1 < 1)$ and similarily with y and z_2. This has eliminated most of the effects of the boundaries but compensated for it by significantly decreasing the accuracy of the numerical method everywhere in the grid. The gain was overcompensated by the loss. Our view has become that to reduce the effects of the waves reflecting from the boundaries it is better to introduce some absorption at the boundaries rather than modify the grid itself. The results of the use of hexagonal grids were very much as expected; the results were more accurate but not sufficiently so to justify the larger memory requirements (assuming the same dx more points are needed to cover the same "physical" area).

To increase the accuracy, we have also implemented fixed multi-grid methods. Here the idea involves embedding grids of different mesh sizes located, in an onion like fashion, at the centre of the grid. The embedded grids all have the same number of points but they have mesh sizes dx half the size of the grid they are embedded into. So if the outer lattice extends from $-L$ to L (in both the x and y directions) with mesh size $dx_0 = L/(N-1)$, where N is the number of points along each direction of the grid, then the second grid extends from $-L/2$ to $L/2$ with a mesh size $dx_1 = dx_0/2$. In general the n's grid extends from $-L/(2^n)$ to $L/(2^n)$ and has a mesh size $dx_n = dx_1/(2^n)$.

Integrating the different meshes separately and merging them after each integration step we have been able to use the same array for the two extra temporary fields needed for the Runge–Kutta method. This means that by using 2 embedded grids we could reduce the mesh size dx by a factor of two (at the centre of the grid) and use only $4/3$ of the memory needed for the original grid, while using a single grid with similarly reduced dx everywhere on the grid this time, would require 4 times as much memory. For 3 embedded grids, the mesh size at the centre is reduced by a factor of 4 with only $5/3$ increase in the memory requirement, while a single grid with the same dx would raise the memory needs by a factor of 16.

We have used the multigrid methods to study the scattering of solitons as well as the blowing up of a single S^2 soliton [3]. The motivation for reducing the mesh size at the centre of the grid comes from the fact that in most cases we are interested in a scattering which takes place at the centre of the grid. Thus the fields change the most at the centre of the grid and it is there that we need higher accuracy. The multigrid method can also be used to increase the size of the lattice so that the waves, usually generated during the scattering, take longer to reach the edge of the grid and so have a smaller impact on the scattering itself. Waves can usually be absorbed on the edges of the grid, but the absorption is never perfect and there is always a small amount of reflection taking place. Using a larger grid reduces such effects.

We have not implemented our multigrid methods with the leapfrog method. In this case the gain in memory requirement is less important. The main reason for this is that in the leapfrog method we must keep 2 copies of the fields everywhere on the grid and so we are unable to use the same extra temporary fields for different subgrids. To embed n subgrids we would thus need n times the memory required for a single grid.

The spectral method has also been applied, with success, by P. Sutcliffe [12] for the baby Skyrme model using the ϕ field formulation. His results are promising but because of the global nature of the spectral methods, it is not clear if they can be used for multi-map fields.

9. CONCLUSIONS

It is clear that solving partial differential equations on a non-flat manifold requires the development of special methods to take into account the curvature of the manifold. We have looked at three different methods of performing a numerical integration of sigma models valued in S^2; they involved describing the fields of the model by, respectively, a unit length real vector, the polar angles on the sphere, or a complex field corresponding to the stereographic projection of S^2 onto the complex plane.

When we use the complex field or the polar angle formulations it is advisable to use two different maps to avoid coordinates singularities. Each point on the lattice is described by one map and when the derivatives of the field, at this point, are calculated the adjacent points may have to be converted to this map. Moreover, the map may vary as the fields evolve with time. Thus to implement such methods the code must perform the necessary bookkeeping to ensure that any computation involving fields at different points is performed in a specific map.

Our tests have shown that all our simulations, which involved 3 different descriptions of the sphere and which used finite difference operators for spatial derivatives, give reliable results. However, when more accurate results are required each of our methods can be improved; when using the unit length vector ϕ, the numerical errors can be decreased by modifying the various differential operators appearing in the equations to take into account the fact that the 3 components of the vector are not independent. For the polar angle and the complex fields we have derived some nonlinear finite difference differential operators which, when used, also improve the quality of the integration.

We have also compared the leapfrog and the 4th order Runge–Kutta methods and found no significant difference between the two, except that the leapfrog method seemed unstable when used with the polar angles.

ACKNOWLEDGMENTS

We thank our collaborators, and in particular M. Peyrard, P. Sutcliffe, and A. W. Craig, for sharing with us their numerical experience. One of us (W.J.Z.) thanks A. D. Rutenberg for the discussions of the accuracy of multipoint Laplacians.

REFERENCES

1. E. Cartan, La théorie de groupes finis et continus et l’ analysis situs, *Mém. Sci. Math. Fasc.* **42** (1930).
2. R. A. Leese, M. Peyrard, and W. J. Zakrzewski, Soliton scatterings in some relativistic models in $(2+1)$ dimensions, *Nonlinearity* **3**, 773 (1990).
3. B. Piette and W. J. Zakrzewski, Shrinking of solitons in the $2+1$ dimensional S^2 sigma model, *Nonlinearity* **9**, 897 (1996).
4. R. A. Battye and P. M. Sutcliffe, Multi-soliton dynamics in the Skyrme model, *Phys. Lett. B* **391**, 150 (1997).
5. M. Peyrard, B. Piette, and W. J. Zakrzewski, Solitons scattering in the Skyrme model in $(2+1)$ dimensions. 2. More general systems, *Nonlinearity* **5**, 585 (1992).
6. B. Piette, B. J. Schroers, and W. J. Zakrzewski, Multisolitons in a two dimensional Skyrme model, *Z. Phys. C* **65**, 165 (1995).
7. B. Piette, B. J. Schroers, and W. J. Zakrzewski, Dynamics of baby Skyrmions, *Nucl. Phys. B* **439**, 205 (1995).
8. H. J. Borchers and W. D. Garber, Local theory of solutions for the $O(2k+1)\,\sigma$ model, *Comm. Math. Phys.* **72**, 77 (1980).
9. W. J. Zakrzewski, Classical solutions to CP^{n-1} models and their generalizations, in Lecture Notes in Physics (Springer-Verlag, New York/Berlin), Vol. 151, p. 160.
10. R. A. Leese, M. Peyrard, and W. J. Zakrzewski, Soliton stability in the $O(3)$ sigma model in $(2+1)$ dimensions, *Nonlinearity* **3**, 387 (1990).
11. A. D. Rutenberg, private communication.
12. P. M. Sutcliffe, private communication.