ESAIM: M2AN 46 (2012) 515–534
DOI: 10.1051/m2an/2011054

ESAIM: Mathematical Modelling and Numerical Analysis
www.esaim-m2an.org

INTEGRATION OF THE EPDIFF EQUATION BY PARTICLE METHODS *, **, ***

Alina Chertock[1], Philip Du Toit[2] and Jerrold Eldon Marsden[3]

Abstract. The purpose of this paper is to apply particle methods to the numerical solution of the EPDiff equation. The weak solutions of EPDiff are contact discontinuities that carry momentum so that wavefront interactions represent collisions in which momentum is exchanged. This behavior allows for the description of many rich physical applications, but also introduces difficult numerical challenges. We present a particle method for the EPDiff equation that is well-suited for this class of solutions and for simulating collisions between wavefronts. Discretization by means of the particle method is shown to preserve the basic Hamiltonian, the weak and variational structure of the original problem, and to respect the conservation laws associated with symmetry under the Euclidean group. Numerical results illustrate that the particle method has superior features in both one and two dimensions, and can also be effectively implemented when the initial data of interest lies on a submanifold.

Mathematics Subject Classification. 35708, 37K10, 65M25, 74J35, 76B15.

Received September 23, 2009.
Published online January 11, 2012.

1. Introduction

This paper is concerned with numerical solutions of the EPDiff-equation; that is, the Euler-Poincaré (EPDiff) equation associated with the diffeomorphism group. In the d-dimensional case on $\mathbb{R}^d$, this equation reads as follows:

$$\frac{\partial \mathbf{m}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{m} + \nabla \mathbf{u}^T \cdot \mathbf{m} + \mathbf{m}(\operatorname{div} \mathbf{u}) = 0. \tag{1.1}$$

Here the momentum $\mathbf{m}$ and velocity $\mathbf{u}$ are vector functions of the time variable t and the d spatial variables $\mathbf{x} = (x_1, \ldots, x_d)$, and are related by the second-order Helmholtz operator,

$$\mathbf{m} = \mathbf{u} - \alpha^2 \Delta \mathbf{u}, \tag{1.2}$$

with length scale α.

Keywords and phrases. Solitons, peakons, integrable Hamiltonian systems, particle methods, weak solutions, variational principle, momentum maps, shallow water and internal waves.

* *The research of A. Chertock is partially supported by NSF grant DMS-0712898.*

** *The research of P. Du Toit is partially supported by AFOSR contract FA9550-08-1-0173.*

*** *The research of J. E. Marsden is partially supported by AFOSR contract FA9550-08-1-0173.*

[1] Department of Mathematics, North Carolina State University, Raleigh, 27695 NC, USA. chertock@math.ncsu.edu

[2] Control and Dynamical Systems, California Institute of Technology, Pasadena, 91125 CA, USA. pdutoit@cds.caltech.edu

[3] Control and Dynamical Systems, California Institute of Technology, Pasadena, 91125 CA, USA. jmarsden@caltech.edu

Article published by EDP Sciences

© EDP Sciences, SMAI 2012

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

The system (1.1)–(1.2) arises in diverse scientific applications. In particular, it applies in a variety of fluid situations ranging from solitons to turbulence. The EPDiff equation coincides with the dispersionless case of the Camassa-Holm (CH) equation for shallow water in one-dimension (1-D) and two-dimensions (2-D), discussed in [4, 30]. Applying viscosity to the incompressible, three-dimensional (3-D) analog of this equation produces the Navier-Stokes-α model for the averaged fluid equations (see, *e.g.*, [8]). The nonlocal relation (1.2) between the momentum density $\mathbf{m}$ and the velocity $\mathbf{u}$ also appears in the theory of fully nonlinear shallow water waves (see [4, 27, 28]). The EPDiff equation has many further interpretations beyond fluid applications. For instance, in 2-D, it also coincides with the averaged template matching equation (ATME) for computer vision (see, *e.g.*, [23,26] and references therein). In [29], it was shown that geometrical structures in computational anatomy, such as landmarks and image outlines, can also be described by singular solutions of the EPDiff equation.

The EPDiff equation has some very interesting properties. The analysis in [26, 27], for instance, shows that a class of weak solutions of (1.1)–(1.2) can be expressed as

$$\mathbf{m}(\mathbf{x},t)=\sum_{i=1}^{N}\int \mathbf{P}_i(s,t)\delta(\mathbf{x}-\mathbf{Q}_i(s,t))\,\mathrm{d}s \tag{1.3}$$

with the corresponding velocity field

$$\mathbf{u}(\mathbf{x},t)=\sum_{i=1}^{N}\int \mathbf{P}_i(s,t)G(\mathbf{x}-\mathbf{Q}_i(s,t))\,\mathrm{d}s, \tag{1.4}$$

where $\mathbf{P}_i,\mathbf{Q}_i\in\mathbb{R}^d$ for $i=1,\dots,N$ and G is the Green's function for the Helmholtz operator (1.2). Solutions (1.3) are vector-valued functions supported in $\mathbb{R}^d$ on a set of N surfaces (or curves) of codimension $d-k$, where $1\le k<d$ is a positive integer and $s\in\mathbb{R}^k$. For instance, these solutions may be supported on a set of points (vector peakons) if $k=0$, 1-D filaments (strings) if $k=1$, or 2-D surfaces (sheets) if $k=2$ in three dimensions. As was shown in [26], the evolution of these support sets is governed by canonical Hamiltonian equations,

$$\frac{\partial\mathbf{Q}_i(s,t)}{\partial t}=\frac{\delta H^N}{\delta\mathbf{P}_i},\qquad \frac{\partial\mathbf{P}_i(s,t)}{\partial t}=-\frac{\delta H^N}{\delta\mathbf{Q}_i}, \tag{1.5}$$

where the Hamiltonian function H^N is

$$H^N=\frac{1}{2}\iint\sum_{i,j=1}^{N}\left(\mathbf{P}_i(s,t)\cdot\mathbf{P}_j(s',t)\right)G(\mathbf{Q}_i(s,t)-\mathbf{Q}_j(s',t))\,\mathrm{d}s\,\mathrm{d}s' \tag{1.6}$$

and $\mathbf{P}_i\cdot\mathbf{P}_j$ denotes the inner product of two vectors in R^d.

The main source of difficulty one comes across while numerically solving the EPdiff equation is the loss of smoothness, that is, the solution may develop discontinuities even for infinitely smooth initial data, see, *e.g.*, [26, 27]. For instance, in 2-D, the nonlinear wave structure of the solution (1.4) for the velocity includes contact discontinuities that carry momentum so that the wave front interactions represent collisions in which momentum is exchanged. This behavior is reminiscent of the soliton paradigm in 1-D. Indeed, the 1-D EPDiff equation (that is, the dispersionless CH equation), like the KdV equation, is completely integrable and the corresponding solitary waves that emerge from the initial data, exhibit a remarkable stability – their identity is preserved through nonlinear interactions, as was shown by [4, 5]. See also [1] and references therein. There are however some important differences between the KdV and CH solitons. A particularly unusual feature of the CH equation is that it admits solutions that are nonlinear superpositions of traveling waves and troughs that have a discontinuity in the first derivative at their peaks and therefore are called peakons. Capturing these solutions numerically is a challenging task especially when a peakon-antipeakon interaction needs to be resolved.

There are only a few numerical works in the literature that consider the EPDiff equation. Several finite-difference, finite-element, i finite-volume, and spectral methods have been developed for simulation of peakon

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

interactions in 1-D; see, for example, [2, 13, 24, 25, 36, 38]. Even in the 1-D case, the implementation of these schemes is computationally demanding and requires a large number of grid points along with adaptivity techniques to be used. The level of numerical complexity increases even further in 2-D and 3-D, since the nonlinear interaction between the waves may lead to an extremely complicated structure of the solutions, as was shown in [28], where a numerical method for the EPDiff equation, which is referred to as the compatible differencing algorithm (CDA), has been developed.

In this paper, we use particle methods for the numerical simulation and investigation of solitary wave structures of the EPDiff equation in 1-D and 2-D. In these methods, the solution is sought as a linear combination of Dirac distributions, whose positions and coefficients represent locations and weights of the particles, respectively. The solution is then found by following the time evolution of the locations and the weights of these particles according to a system of ODEs obtained by considering a weak formulation of the problem. The main advantage of particle methods is their (extremely) low numerical diffusion that allows one to capture a variety of nonlinear waves with high resolution. Even though the most "natural" application of particle methods is to the case of "pure" transport equations, (see, *e.g.*, [15, 35] and references therein), over the years, the range of these methods has been extended to the case of convection-diffusion (see, *e.g.*, [12, 16–20]) and dispersive equations (see, *e.g.*, [10, 11]).

Particle methods have also recently been applied to numerical solutions of a 1-D nonlinear shallow water equation in [6, 7], and to the study of the dynamics of N point particles ("blobs") governed by the Euler equations, [34]. The methods presented in [6, 7, 34] have been derived using a discretization of a variational principle, while here the equivalent representation of the particle system is obtained by considering a weak formulation of the problem.

The paper is organized as follows. In the next section, we give an overview of particle methods and some of their main features for the convenience of the reader. Then in Section 3 we consider the 1-D case, in which the EPDiff equation coincides with the dispersionless limit of the CH equation. Even though this case has been extensively studied, we show that, even here, particle methods do have some advantages. Unlike previous works, here we derive a particle method based on the weak formulation of the problem and show that the resulting equations of motion coincide with those obtained from the variational formulation. Section 4 is devoted to the analytical and numerical study of the 2-D case. Symmetries are discussed in Section 4.1, and it is shown that particle methods are compatible with Euclidean symmetries and that they preserve the Hamiltonian structure of the problem. Numerical experiments reported in Section 4.2 show that particle methods can effectively handle the 2-D case as well and present huge computational savings when the initial data of interest are distributed along straight line or curve segments.

2. Particle methods – an overview

In this section, we briefly describe particle methods in the context of linear transport equations. For a complete description of the method and its derivation, we refer the reader to [35] and the references therein.

We consider a linear transport equation with variable coefficients written in conservation form:

$$\frac{\partial m}{\partial t} + \nabla \cdot (\mathbf{a}\, m) + a_0 m = f, \qquad \mathbf{x} \in \mathbb{R}^d, \qquad t > 0, \tag{2.1}$$

subject to initial data

$$m_0(\mathbf{x}) := m(\mathbf{x}, 0), \qquad \mathbf{x} \in \mathbb{R}^d, \tag{2.2}$$

which is to be solved for an unknown function $m(\mathbf{x}, t)$. In the general multidimensional case, m, a_0 and f are functions of a time variable t and d spatial variables $\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{a}(\mathbf{x}, t) = (a_1(\mathbf{x}, t), \ldots a_d(\mathbf{x}, t))$ is a vector field in $\mathbb{R}^d$.

In order to define a particle method for problem (2.1)–(2.2), we need to consider weak solutions of (2.1). Following [35], we denote by $\mathfrak{M}(S)$ the space of measures defined on a subspace S of $\mathbb{R}^d$, that is, the space dual

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

to the space $C_0^0(S)$ of continuous functions $S \to \mathbb{R}$ with compact support. In this context, one introduces the following:

Definition 2.1. *A function $m \in \mathfrak{M}(\mathbb{R}^d \times [0,T))$ is called a* **weak solution** *of equation (2.1) if*

$$-\int_{\mathbb{R}^d} m_0(\mathbf{x})\phi(\mathbf{x},0)\,\mathrm{d}\mathbf{x} - \int_0^T\int_{\mathbb{R}^d} m\left[\phi_t + \mathbf{a}\cdot\nabla\phi\right]\,\mathrm{d}\mathbf{x}\,\mathrm{d}t + \int_0^T\int_{\mathbb{R}^d} a_0 m\phi\,\mathrm{d}\mathbf{x}\,\mathrm{d}t = \int_0^T\int_{\mathbb{R}^d} f\phi\,\mathrm{d}\mathbf{x}\,\mathrm{d}t \tag{2.3}$$

holds for any function $\phi \in C_0^1(\mathbb{R}\times[0,T))$.

Note that this definition makes sense if $m_0 \in \mathfrak{M}(\mathbb{R}^d)$ and $f \in \mathfrak{M}(\mathbb{R}^d \times [0,T))$.

A first step in the particle method consists of approximating the initial data (2.2) by a linear combination of Dirac distributions,

$$m_0^N(\mathbf{x}) = \sum_{i=1}^N p_i^0 \delta(\mathbf{x} - \mathbf{x}_i^0), \tag{2.4}$$

for some finite set $(\mathbf{x}_i^0, p_i^0)$ of points (particles) $\mathbf{x}_i^0 \in \mathbb{R}^d$ and their weights $p_i^0 \in \mathbb{R}$, where N is the total number of particles. Then, an approximate particle solution m^N of (2.1)–(2.2) is found by letting

$$m^N(\mathbf{x},t) = \sum_{i=1}^N p_i(t)\delta(\mathbf{x} - \mathbf{x}_i(t)), \tag{2.5}$$

where

$$\begin{cases} \dfrac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t} = \mathbf{a}(\mathbf{x}_i(t),t), \qquad \mathbf{x}_i(0) = \mathbf{x}_i^0, \\[2ex] \dfrac{\mathrm{d}p_i}{\mathrm{d}t} + a_0(\mathbf{x}_i(t),t)p_i(t) = \beta_i(t), \qquad p_i(0) = p_i^0. \end{cases} \tag{2.6}$$

Here $\beta_i(t)$ reflects the contribution of the source term f (see, *e.g.*, [14, 35]), which can be approximated by

$$f(\mathbf{x},t) = \sum_{i=1}^N \beta_i(t)\delta(\mathbf{x} - \mathbf{x}_i(t)), \quad \beta_i(t) = \int_{D_i(t)} f(\mathbf{x},t)\,\mathrm{d}\mathbf{x} \approx f(\mathbf{x}_i(t),t)\cdot|D_i(t)|, \tag{2.7}$$

where, $D_i(t)$ is the domain that includes the ith particle and is such that $p_i(t) = \int_{D_i(t)} m(\mathbf{x},t)\,\mathrm{d}\mathbf{x}$. The size of $D_i(t)$ is typically obtained by solving the following ODE:

$$\frac{\mathrm{d}}{\mathrm{d}t}|D_i(t)| = |D_i(t)|\cdot \operatorname{div}\mathbf{a}(\mathbf{x}_i,t). \tag{2.8}$$

We recall now why it is that the function m^N given by (2.5)–(2.6) is indeed a weak solution of (2.1)–(2.2).

Proposition 2.2. *Assume that $a_i \in C^0(\mathbb{R}^d \times [0,T]), 1 \le i \le d$. If $m_0 \in \mathfrak{M}(\mathbb{R}^d)$ and $f \in \mathfrak{M}(\mathbb{R}^d \times [0,T))$, then the function m^N given by (2.5)–(2.6) is a weak solution of the problem (2.1)–(2.2).*

Proof. Let $\phi \in C_0^1(\mathbb{R}\times[0,T))$. Substituting (2.5) and (2.7) into the weak formulation (2.3) and changing the order of summation and integration, yields:

$$\begin{aligned} &-\sum_{i=1}^N p_i(0)\phi(\mathbf{x}_i(0),0) - \sum_{i=1}^N \int_0^T p_i(t)\left[\phi_t(\mathbf{x}_i(t),t) + \mathbf{a}(\mathbf{x}_i(t),t)\cdot\nabla\phi(\mathbf{x}_i(t),t)\right]\,\mathrm{d}t \\ &\qquad + \sum_{i=1}^N \int_0^T p_i(t)a_0(\mathbf{x}_i(t),t)\phi(\mathbf{x}_i(t),t)\,\mathrm{d}t = \sum_{i=1}^N \int_0^T \beta_i(t)\phi(\mathbf{x}_i(t),t)\,\mathrm{d}t. \end{aligned} \tag{2.9}$$

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

We now add and subtract $\sum_{i=1}^{N}\int_0^T p_i(t)\,\frac{\mathrm{d}\mathbf{x}_i}{\mathrm{d}t}\cdot\nabla\phi(\mathbf{x}_i(t),t)\,\mathrm{d}t$ in the last equation, use the fact that

$$\phi_t+\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\cdot\nabla\phi=\frac{\mathrm{d}\phi}{\mathrm{d}t},$$

and rewrite (2.9) as follows:

$$\begin{aligned}&-\sum_{i=1}^{N}p_i(0)\phi(\mathbf{x}_i(0),0)-\sum_{i=1}^{N}\int_0^T p_i(t)\frac{\mathrm{d}\phi(\mathbf{x}_i(t),t)}{\mathrm{d}t}\,\mathrm{d}t\\&\quad+\sum_{i=1}^{N}\int_0^T p_i(t)\left[\frac{\mathrm{d}\mathbf{x}_i(t)}{\mathrm{d}t}-\mathbf{a}(\mathbf{x}_i(t),t)\right]\cdot\nabla\phi(x_i(t),t)\,\mathrm{d}t\\&\quad+\sum_{i=1}^{N}\int_0^T p_i(t)a_0(\mathbf{x}_i(t),t)\phi(\mathbf{x}_i(t),t)\,\mathrm{d}t=\sum_{i=1}^{N}\int_0^T\beta_i(t)\phi(\mathbf{x}_i(t),t)\,\mathrm{d}t.\end{aligned}\tag{2.10}$$

Integrating by parts the second term in the first row in (2.10), and rearranging other terms, we finally obtain:

$$\begin{aligned}&\int_0^T\sum_{i=1}^{N}p_i(t)\left[\frac{\mathrm{d}\mathbf{x}_i(t)}{\mathrm{d}t}-\mathbf{a}(\mathbf{x}_i(t),t)\right]\cdot\nabla\phi(\mathbf{x}_i(t),t)\,\mathrm{d}t\\&\quad+\int_0^T\sum_{i=1}^{N}\left[\frac{\mathrm{d}p_i(t)}{\mathrm{d}t}+p_i(t)a_0(\mathbf{x}_i(t),t)-\beta_i(t)\right]\phi(\mathbf{x}_i(t),t)\,\mathrm{d}t=0.\end{aligned}\tag{2.11}$$

Since the functions $\mathbf{x}_i(t)$ and $p_i(t)$ satisfy the system (2.6), the last equation holds for any ϕ implying that m^N defined by (2.5)–(2.6) is a weak solution of (2.1)–(2.2). This completes the proof. □

In practice, except for very special cases, the functions $\mathbf{x}_i(t)$ and $p_i(t)$, $i=1,\dots,N$ have to be determined numerically and the system (2.6) must be integrated by an appropriate ODE solver. In order to initialize the time integration, one should choose the initial positions of particles, $\mathbf{x}_i^0$, and the weights, p_j^0, so that (2.4) represents a high-order approximation to the initial data. The latter can be done in the sense of measures on $\mathbb{R}^d$. Given a test function $\phi\in C_0^1(\mathbb{R}^d)$, the inner product, $(m_0(\cdot),\phi(\cdot))$, should be approximated by

$$(m^N(\cdot),\phi(\cdot))=\int_{\mathbb{R}^d}m_0(\mathbf{x})\phi(\mathbf{x})\,\mathrm{d}\mathbf{x}\approx\sum_{i=1}^{N}p_i^0\phi(\mathbf{x}_i).$$

In other words, the constants $\{p_i^0\}$, can be determined by solving the standard numerical quadrature problem. One way of solving the last equation can be, *e.g.*, to cover the computational domain $\mathbb{R}^d$ with a uniform mesh of spacing $\Delta x>0$ and denote by C_i the cell

$$C_i=\left\{\mathbf{x}\in\mathbb{R}^d\;\middle|\;(i_k-1/2)\Delta x\le x_k\le(i_k+1/2)\Delta x,\ 1\le k\le d\right\},$$

and by $x_i^0=(i_k\Delta x)_{1\le k\le d}$ the center of C_i, $i=(i_1,\dots,i_d)\in\mathbb{Z}^d$. For example, a midpoint quadrature is then given by setting $p_i^0=(\Delta x)^d m_0(x_i^0)$.

Finally, in order to recover point values of the solution $m(\mathbf{x},t)$ at some time $t>0$, one needs to regularize the particle solution $m^N(\mathbf{x},t)$. Such a regularization is usually performed by a convolution product with a "cutoff function" $\zeta(\mathbf{x})$ that after a proper scaling takes into account the initial tightness of the particle discretization, namely

$$m_\varepsilon^N(\mathbf{x},t)=(m^N*\zeta_\varepsilon)(\mathbf{x},t)=\sum_{i=1}^{N}p_i(t)\zeta_\varepsilon(\mathbf{x}-\mathbf{x}_i(t)).\tag{2.12}$$

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

The function $\zeta_\varepsilon(\mathbf{x})$ is taken as a smooth approximation of the δ-function which satisfies

$$\zeta_\varepsilon(\mathbf{x}) = \frac{1}{\varepsilon^d}\zeta\left(\frac{\mathbf{x}}{\varepsilon}\right), \qquad \text{and} \qquad \int_{\mathbb{R}^d} \zeta(\mathbf{x})\,\mathrm{d}\mathbf{x} = 1. \tag{2.13}$$

There is an extensive discussion in the literature on the selection of the cutoff function and its relation to the accuracy of particle methods, see, *e.g.* [12, 15, 22, 35] and references therein. We also refer the reader to [9], where several different strategies for recovering point values of particle solutions were suggested.

3. One-dimensional case

In this section, we consider the 1-D version of the EPDiff equation (1.1)–(1.2):

$$m_t + (um)_x + u_x m = 0 \qquad \text{with} \qquad m = u - \alpha^2 u_{xx}. \tag{3.1}$$

As mentioned above, (3.1) coincides with the dispersionless CH equation, [4], and was first derived by using asymptotic expansions directly in the Hamiltonian for Euler's equations in the shallow water regime. The equation is completely integrable and possesses soliton solutions, called peakons, that have a discontinuity in the first derivative at their peaks (see, *e.g.*, [4, 5] and references therein). As was noted in [1], for some cases in which one wants to capture jumps in the derivatives in a variational context, it is necessary to make use of weak solutions in a *space-time* sense. This is compatible with our particle method approach, which also uses a space-time formulation of weak solutions.

Here, we numerically solve equation (3.1) by a particle method, described in the last section. To this end, we seek a solution of (3.1) as a linear combination of Dirac distributions:

$$m^N(x,t) = \sum_{i=1}^{N} p_i(t)\delta(x - x_i(t)), \tag{3.2}$$

whose positions, $x_i(t)$, and coefficients, $p_i(t)$, represent locations and weights of the particles, respectively, and N is a total number of particles. Recall that we consider the weak formulation (2.3) and substitute the Ansatz (3.2) to obtain a set of corresponding ODEs, namely those given by the system (2.6). In our case, these equations for $x_i(t)$ and $p_i(t)$ become:

$$\begin{cases} \dfrac{\mathrm{d}x_i}{\mathrm{d}t} = u(x_i,t), \\[2ex] \dfrac{\mathrm{d}p_i}{\mathrm{d}t} + u_x(x_i,t)p_i = 0. \end{cases} \tag{3.3}$$

The velocity u in (3.3) is obtained by convolution $u = G * m$ with the Green's function,

$$G(|x-y|) = \frac{1}{2}\mathrm{e}^{-|x-y|/\alpha}, \tag{3.4}$$

for the 1-D Helmholtz operator that relates m and u in (3.1). Consequently, the velocity and its derivative corresponding to the solution (3.2) are given by

$$u(x,t) = \frac{1}{2}\sum_{i=1}^{N} p_i(t)\mathrm{e}^{-|x-x_i(t)|/\alpha} \tag{3.5}$$

and

$$u_x(x,t) = -\frac{1}{2\alpha}\sum_{i=1}^{N} p_i(t)\mathrm{sgn}(x - x_i(t))\mathrm{e}^{-|x-x_i(t)|/\alpha}, \tag{3.6}$$

respectively.

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

As described in Section 2, the system (3.3) should be integrated numerically by an appropriate ODE solver. In our numerical experiments below (both 1-D and 2-D), we have tested a variety of methods: a standard fourth order Runge-Kutta method, a standard fourth (fifth) order adaptive Runge-Kutta-Fehlberg method, a strong-stability preserving (SSP) Runge-Kutta method [21], as well as a variational integrators derived from a discrete Hamilton's phase space variational principle (see, *e.g.*, [33]). Except for a few degenerate cases to be discussed later, the integration of the ODEs is straightforward and all the methods provide consistent solutions.

Notice that when solving the 1-D EPDiff equation by the particle method, one does not need to use a cutoff function (2.12), (2.13) to recover point values of the computed solution: once the positions, x_i, of the particles and their weights, p_i, are obtained from (3.3), the solution at any point can be easily computed using the formula (3.5).

3.1. Properties of the particle system

We would like to point out that the system (3.3), obtained using the weak formulation of the problem, is equivalent to the one derived in [6,7] by considering the Hamiltonian structure of equation (3.1). One can verify that the functions $x_i(t)$ and $p_i(t)$ in (3.3) satisfy the canonical Hamiltonian equations:

$$\frac{\mathrm{d}x_i}{\mathrm{d}t} = \frac{\partial H^N}{\partial p_i}, \qquad \frac{\mathrm{d}p_i}{\mathrm{d}t} = -\frac{\partial H^N}{\partial x_i}, \qquad i = 1, \ldots, N, \tag{3.7}$$

where H^N is the Hamiltonian function defined as

$$H^N(t; x, p) = \frac{1}{4\alpha} \sum_{i=1}^{N} \sum_{j=1}^{N} p_i(t) p_j(t) \mathrm{e}^{-|x_i(t) - x_i(t)|/\alpha}, \tag{3.8}$$

and that the total linear momentum of the particle system is conserved, that is,

$$\frac{\mathrm{d}}{\mathrm{d}t} \left[\sum_{i=1}^{N} p_i(t) \right] = 0. \tag{3.9}$$

Since solutions of Hamilton's canonical equations satisfy a variational principle, this shows that the weak solutions (3.2) obtained by the particle method also satisfy Hamilton's phase space variational principle as long as x and p are smooth functions in time; that is,

$$\delta \int_0^T \sum_{i=1}^{N} \left[p_i \dot{x}_i - H^N(t; x_i, p_i) \right] \mathrm{d}t = \int_0^T \sum_{i=1}^{N} \left[\left(\dot{x}_i - \frac{\partial H^N}{\partial p_i} \right) \delta p_i - \left(\dot{p}_i + \frac{\partial H^N}{\partial x_i} \right) \delta x_i \right] \mathrm{d}t = 0 \tag{3.10}$$

for all δx_i and δp_i.

In fact, discretizing Hamilton's phase space variational principle allows for the derivation of a *variational integrator* whose order of accuracy is determined by the order of accuracy of the quadrature rule used to approximate the integral, see, *e.g.* [33]. Using a variational integrator ensures that the momenta are conserved exactly throughout the numerical integration of the ODEs, as we will demonstrate in the numerical experiments below.

3.2. Numerical examples

We now use the particle method to numerically solve equation (3.1) with $\alpha = 1$.

Single peakon. The particle method for simulating the evolution of a single peakon assumes a particularly simple form. Suppose that we choose a suitably refined initial grid of particles to represent the peakon in which the initial weights, p_i, of the particles are all zero except for the n-th particle that initially has weight $p_n(0) = 1$.

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

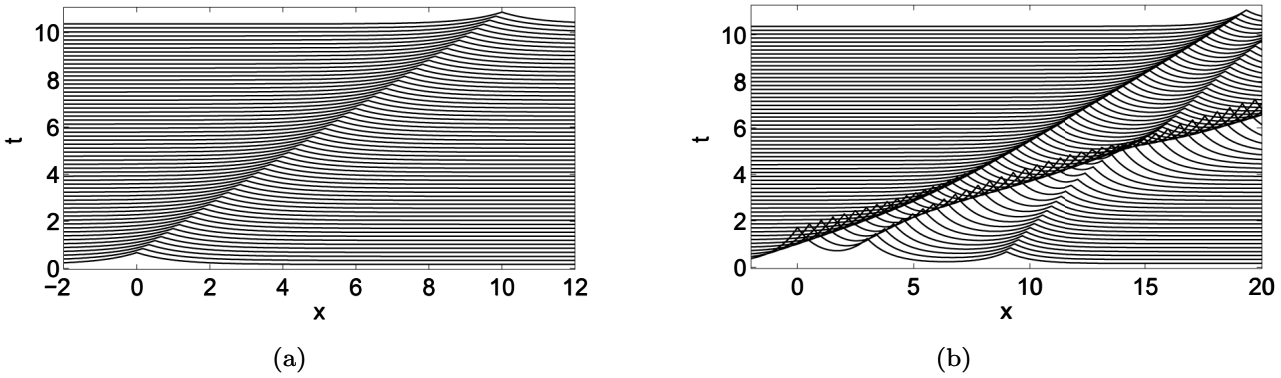


FIGURE 1. Time evolution of (a) one peakon and (b) three peakons computed by the particle method.

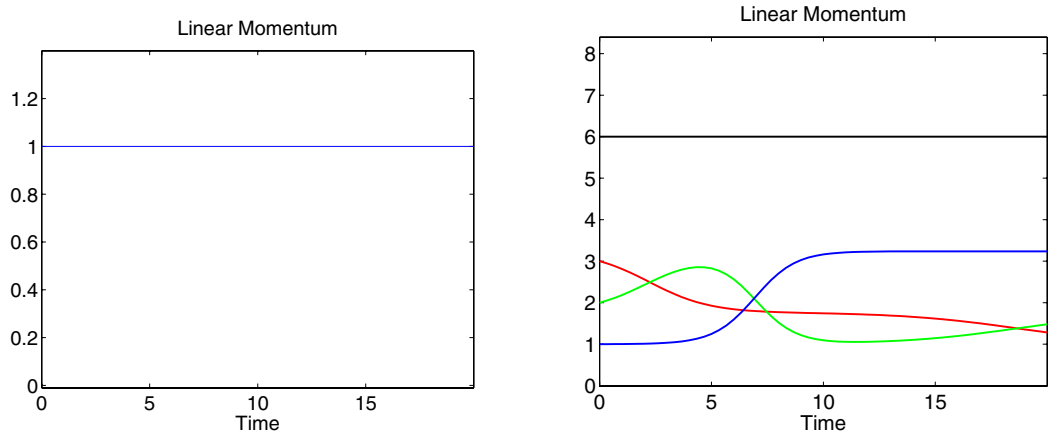
A brief study of (3.3) reveals that $\mathrm{d}p_i/\mathrm{d}t = 0$ for each i, and hence the weights of all the particles remain constant in time. Notwithstanding this fact, the weightless particles ($p_i = 0$) may still move from their initial positions since the solution values, $u(x_i, t)$, that dictate their velocity, need not be zero. The motion of the n-th particle (the only particle with non-zero weight), and the evolution of the solution, $u(x, t)$, depend only on $x_n(t)$ and $p_n(0)$. Hence, the evolution of a solitary peakon can be recovered by integrating the motion of a single particle as described by a single second order ODE. A sample solution is shown in Figure 1a. In general, given any initial data for the particle weights, only the evolution of the particles with non-zero weights needs to be integrated. This observation holds true for numerical integration in 2-D, and represents a huge computational savings when the initial data of interest lies on a submanifold, as we demonstrate in Section 4.2.

Multiple peakon interactions. By increasing the number of particles with non-zero weights, we can study the interaction of multiple peakons. The solutions for the 1-D EPDiff equation are obtained very quickly by the particle method, using just a single particle to represent each peakon. The interaction of three peakons is shown in Figure 1b and illustrates the particle nature of the peakons as they undergo collisions and momentum. The three peakons were initially placed at $x_{n_1}(0) = 0$, $x_{n_2}(0) = 4$ and $x_{n_3}(0) = 10$ with the weights $p_{n_1}(0) = 3$, $p_{n_2}(0) = 2$ and $p_{n_3}(0) = 1$, respectively. In Figure 2, we also illustrate that the total linear momentum of the particle system is conserved in this and in the previous example. We would like to note that similar computations performed by finite-difference, finite-element and finite-volume methods require the use of very fine grids along with some (mesh) adaptive techniques, see, *e.g.*, [2, 13, 24, 25, 38].

A peakon-antipeakon interaction computed by the particle method is shown in Figure 3. The peakon and antipeakon are initially located at $x_{n_1}(0) = 0$ and $x_{n_2}(0) = 18$ and have momenta of equal magnitude but opposite sign ($p_{n_1}(0) = 2$, $p_{n_2}(0) = -2$) so that the total momentum is zero. During the simulation, the total momentum remains zero, however the magnitudes grow very large as the peakon traveling from the left to the right collides with the antipeakon traveling in the opposite direction. In the antisymmetric case, considered here, the solution at the collision time is identically zero. One can prolong the solution after collision in infinitely many ways, but at least two scenarios are possible. The first consists of letting u remain identically zero after collision (Fig. 3a), while according to the second scenario, shown in Figure 3b, at later times the peakons redevelop and depart again according to the symmetry $u(x, t) \to -u(-x, -t)$, see, *e.g.* [3, 4]. This is a particularly difficult situation to treat numerically, as it was shown, for example, in [13], where a finite-difference scheme with a very small mesh size ($\Delta x \approx 10^{-9}$) was used to simulate the peakon-antipeakon interaction, but, due to the numerical viscosity, the scheme was only able to generate the dissipative solution so that for t larger than the collision time, the approximate solution vanished.

Evidently, the particle method is capable of capturing both the dissipative and conservative solutions. These numerical results can be well-understood simply from the viewpoint of ODEs and numerical integration, and by

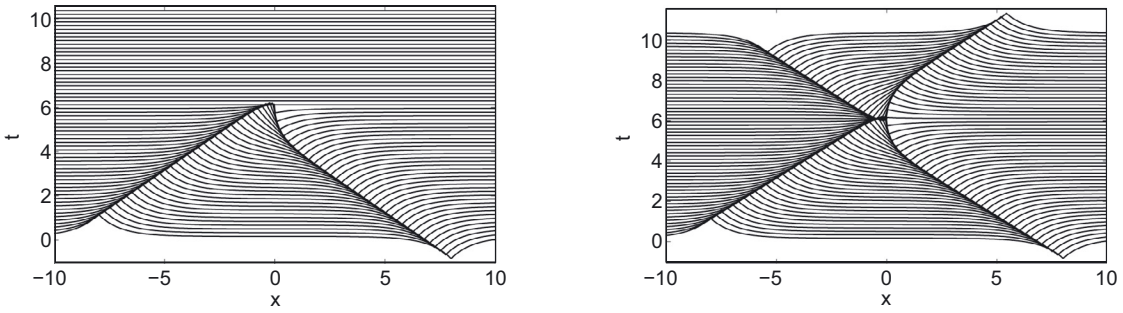
https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press



(a) Momentum of a single peakon is conserved

(b) Total momentum of three interacting peakons is conserved

FIGURE 2. Conservation of the total linear momentum during simulation of peakon solutions presented in Figures 1a and 1b, respectively.



(a) Time evolution of the dissipative solution

(b) Time evolution of the conservative solution

FIGURE 3. Peakon-antipeakon collision computed by the particle method.

looking at the vector field for the system (3.3). Recall that the evolution of two peakons is completely described by two particles, since all the zero weight particles influence neither the motion of the peakons nor the final solution. Therefore, the range of possible interactions can be thoroughly documented and reveals a very simple strategy: if the numerical integration of the system (3.3) is performed so that the particles are allowed to cross each other, then one can recover the symmetric (conservative) solution, while a numerical integration, in which the trajectory that reaches the fixed point "dies", will give the appearance of the dissipative antipeakon collision leading to zero. In practice, this can be done by introducing a critical distance, d_{cr}, between particles. Once the distance between two particles becomes smaller than the critical one, that is, $|x_{n_1} - x_{n_2}| < d_{\mathrm{cr}}$, the two particles either cross each other by switching their locations or coalesce into one particle. In the latter case, the weight of the new particle is simply the sum of the weights of the original particles and therefore is equal to zero.

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

4. Two-dimensional case

In this section, we consider the 2-D EPDiff equation (1.1), (1.2) and rewrite it in the coordinate form, convenient for the particle approximation:

$$\begin{aligned}&\frac{\partial m_1}{\partial t}+(um_1)_x+(vm_1)_y+m_1u_x+m_2v_x=0,\\&\frac{\partial m_2}{\partial t}+(um_2)_x+(vm_2)_y+m_1u_y+m_2v_y=0,\end{aligned}\tag{4.1}$$

where $\mathbf{x}=(x,y)^T,\mathbf{m}=(m_1,m_2)^T$ and $\mathbf{u}=(u,v)^T$. Following the discussion in Section 2, we seek a particle solution of (4.1) as a linear combination of Dirac distributions,

$$m_k^N(x,y,t)=\sum_{i=1}^N p_{k,i}(t)\delta(x-x_i(t))\delta(y-y_i(t)),\quad k=1,2.\tag{4.2}$$

The position of particles, $\mathbf{x}_i=(x_i,\ y_i)$, and the corresponding weights, $\mathbf{p}_i=(p_{1,i},\ p_{2,i})$, in (4.2) are computed by numerically integrating the system of ODEs,

$$\begin{cases}\dfrac{\mathrm{d}\mathbf{x}_i(t)}{\mathrm{d}t}=u(\mathbf{x}_i(t),t),\\[2mm]\dfrac{\mathrm{d}p_{1,i}(t)}{\mathrm{d}t}=-u_x(\mathbf{x}_i(t),t)\,p_{1,i}(t)-v_x(\mathbf{x}_i(t),t)\,p_{2,i}(t),\\[2mm]\dfrac{\mathrm{d}p_{2,i}(t)}{\mathrm{d}t}=-u_y(\mathbf{x}_i(t),t)\,p_{1,i}(t)-v_y(\mathbf{x}_i(t),t)\,p_{2,i}(t).\end{cases}\tag{4.3}$$

The velocities u and v in (4.3) are obtained by solving the 2-D Helmholtz equation (1.2) and are given by:

$$\mathbf{u}(\mathbf{x}_i,t)=\frac{1}{2\pi\alpha^2}\sum_{j=1}^N\mathbf{p}_j(t)K_0\left(\frac{|\mathbf{x}_i-\mathbf{x}_j|}{\alpha}\right),\tag{4.4}$$

and

$$\mathbf{u}_x(\mathbf{x}_i,t)=\frac{1}{2\pi\alpha^3}\sum_{j=1}^N\mathbf{p}_j(t)K_0'\left(\frac{|\mathbf{x}_i-\mathbf{x}_j|}{\alpha}\right)\frac{x_i-x_j}{|\mathbf{x}_i-\mathbf{x}_j|},\tag{4.5}$$

$$\mathbf{u}_y(\mathbf{x}_i,t)=\frac{1}{2\pi\alpha^3}\sum_{j=1}^N\mathbf{p}_j(t)K_0'\left(\frac{|\mathbf{x}_i-\mathbf{x}_j|}{\alpha}\right)\frac{y_i-y_j}{|\mathbf{x}_i-\mathbf{x}_j|},\tag{4.6}$$

where K_0 is the modified Bessel function and K_0' is its derivative.

Note that the function $K_0(\mathbf{x})$ has a singularity at $\mathbf{x}=0$ and therefore implementing directly equations (4.3)–(4.6) may lead to very large values when two particles approach each other. Hence, in practice we replace K_0 with a mollification $K_{0,\varepsilon}$ obtained by setting

$$K_{0,\varepsilon}(\mathbf{x})=(K_0*\zeta_\varepsilon)(\mathbf{x})=\int_{\mathbb{R}^2}K_0\left(|\mathbf{x}-\mathbf{y}|\right)\zeta_\varepsilon(\mathbf{y})\,\mathrm{d}\mathbf{y},\tag{4.7}$$

where ζ is a smooth cutoff function that satisfies (2.13). One of the simplest examples of a cutoff function, which we used in the numerical experiments below, is the Gaussian function defined by

$$\zeta_\varepsilon(\mathbf{x})=\frac{1}{2\pi\varepsilon^2}\mathrm{e}^{-\frac{|\mathbf{x}|^2}{2\varepsilon^2}}.\tag{4.8}$$

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

In this case, the convolution integral (4.7) can be computed using the fact that the Fourier transform of $K_{0,\varepsilon}$ is related to the Fourier transforms of K_0 and ζ_ε by

$$\mathcal{F}[K_{0,\varepsilon}](r) = 2\pi\mathcal{F}[K_0](r)\mathcal{F}[\zeta_\varepsilon](r),$$

where

$$\mathcal{F}[K_0](r) = \frac{\alpha^2}{1+\alpha^2 r^2} \quad \text{and} \quad \mathcal{F}[\zeta_\varepsilon](r) = \frac{1}{2\pi}\mathrm{e}^{-\frac{r^2\varepsilon^2}{2}}.$$

$K_{0,\varepsilon}$ is then found by the inverse Fourier transform, which due to the radial symmetry, reduces to the inverse Hankel transform given by

$$K_{0,\varepsilon}(|\mathbf{x}|) = 2\pi\int_0^\infty rJ_0(r|\mathbf{x}|)\mathcal{F}[K_0](m)\mathcal{F}[\zeta_\varepsilon](r)\,\mathrm{d}r = \int_0^\infty rJ_0(r|\mathbf{x}|)\,\frac{\alpha^2\mathrm{e}^{-\frac{r^2\varepsilon^2}{2}}}{1+\alpha^2r^2}\,\mathrm{d}r, \tag{4.9}$$

where $J_0(x)$ is the Bessel function of the first kind.

We would like to point out that the system (4.3) can be viewed as a particular case of the system of integro-partial-differential equations (1.3)–(1.4) (see, *e.g.*, [26]). We would also like to note that the system (4.3), derived from the weak formulation of particle methods as a starting point, is equivalent to a Hamiltonian form of the particle method that (without using α) was obtained in [34] – so their derivation using a discretization of the variational principle gives an equivalent result.

4.1. Properties of the particle system

In this section, we discuss the properties of the derived particle method and show that, even with a mollified kernel, the particle system (4.3) is Hamiltonian. The Hamiltonian will be shown to be translation and rotationally invariant yielding conservation of linear and angular momentum. As we shall show, these conservation properties can also be seen directly.

Conservation of linear momentum. First, we show that the total linear momentum of the particle system is conserved, that is,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left[\sum_{i=1}^N p_{1,i}(t) + \sum_{i=1}^N p_{2,i}(t)\right] = 0. \tag{4.10}$$

Moreover, we show that, in fact, each component of the linear momentum in (4.10) is conserved separately. Indeed, using (4.3) together with (4.5) with K_0 replaced by the mollified kernel $K_{0,\varepsilon}$, yields

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=1}^N p_{1,i} &= -\sum_{i=1}^N [p_{1,i}u_x(\mathbf{x}_i,t) + p_{2,i}v_x(\mathbf{x}_i,t)] \\ &= -\frac{1}{2\pi\alpha^3}\sum_{i=1}^N\sum_{j=1}^N (p_{1,i}p_{1,j}+p_{2,i}p_{2,j})K'_{0,\varepsilon}\left(\frac{|\mathbf{x}_i-\mathbf{x}_j|}{\alpha}\right)\frac{x_i-x_j}{|\mathbf{x}_i-\mathbf{x}_j|} = 0,\end{aligned} \tag{4.11}$$

since the summation in (4.11) is performed over all $i,j = 1,\ldots,N$. In a similar fashion, using (4.3) and (4.6), one can show that

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i=1}^N p_{2,i} &= -\sum_{i=1}^N [p_{1,i}u_y(\mathbf{x}_i,t) + p_{2,i}v_y(\mathbf{x}_i,t)] \\ &= -\frac{1}{2\pi\alpha^3}\sum_{i=1}^N\sum_{j=1}^N (p_{1,i}p_{1,j}+p_{2,i}p_{2,j})K'_{0,\varepsilon}\left(\frac{|\mathbf{x}_i-\mathbf{x}_j|}{\alpha}\right)\frac{y_i-y_j}{|\mathbf{x}_i-\mathbf{x}_j|} = 0.\end{aligned} \tag{4.12}$$

Substituting (4.11) and (4.12) into (4.10), we conclude that the linear momentum is conserved.

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

Conservation of angular momentum. Next, we show that the angular momentum, $\mathbf{x}\times\mathbf{p}$, $\mathbf{x}=(x,y)^T$, $\mathbf{p}=(p_1,p_2)^T$, is also conserved, that is,

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\mathbf{x}\times\mathbf{p}\right)=\frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t}\times\mathbf{p}+\frac{\mathrm{d}\mathbf{p}}{\mathrm{d}t}\times\mathbf{x}=\mathbf{0}. \tag{4.13}$$

Note that (4.13) is a vector equation, whose first two components are trivially zero, and therefore it can be reduced to a scalar equation for its third component, namely,

$$\sum_{i=1}^{N}\left[\frac{\mathrm{d}x_i}{\mathrm{d}t}\,p_{2,i}-\frac{\mathrm{d}y_i}{\mathrm{d}t}\,p_{1,i}+x_i\,\frac{\mathrm{d}p_{2,i}}{\mathrm{d}t}-y_i\,\frac{\mathrm{d}p_{1,i}}{\mathrm{d}t}\right]=0. \tag{4.14}$$

Using (4.3), the right-hand side of (4.14) becomes

$$\begin{aligned}&\sum_{i=1}^{N}\left[u_ip_{2,i}-v_ip_{1,i}-x_i\left(u_{y,i}p_{1,i}+v_{y,i}p_{2,i}\right)+y_i\left(u_{x,i}p_{1,i}+v_{x,i}p_{2,i}\right)\right]\\&=\sum_{i=1}^{N}\left[u_ip_{2,i}-v_ip_{1,i}\right]+\sum_{i=1}^{N}p_{1,i}\left(y_iu_{x,i}-x_iu_{y,i}\right)+\sum_{i=1}^{N}p_{2,i}\left(y_iv_{x,i}-x_iv_{y,i}\right),\end{aligned} \tag{4.15}$$

where u_i,v_i and $u_{x,i},u_{y,i},v_{x,i},v_{y,i}$ are the values of u,v and their derivatives with respect to x and y at the point $(x_i(t),y_i(t),t)$, respectively.

We shall now show that each sum in the second row of (4.15) is equal to zero. We denote each of these sums by

$$S_1=\sum_{i=1}^{N}\left[u_ip_{2,i}-v_ip_{1,i}\right],\quad S_2=\sum_{i=1}^{N}p_{1,i}\left(y_iu_{x,i}-x_iu_{y,i}\right),\quad S_3=\sum_{i=1}^{N}p_{2,i}\left(y_iv_{x,i}-x_iv_{y,i}\right).$$

Substituting the expressions (4.4)–(4.6), with K_0 replaced by $K_{0,\varepsilon}$, into the above sums, yields

$$S_1=\frac{1}{2\pi\alpha^2}\sum_{i=1}^{N}\left[p_{2,i}\sum_{j=1}^{N}p_{1,j}K_{0,\varepsilon}\left(\frac{|\mathbf{x}_i-\mathbf{x}_j|}{\alpha}\right)-p_{1,i}\sum_{j=1}^{N}p_{2,j}K_{0,\varepsilon}\left(\frac{|\mathbf{x}_i-\mathbf{x}_j|}{\alpha}\right)\right]=0, \tag{4.16}$$

$$S_2=\frac{1}{2\pi\alpha^3}\sum_{i=1}^{N}\sum_{j=1}^{N}p_{1,i}p_{1,j}K'_{0,\varepsilon}\left(\frac{|\mathbf{x}_i-\mathbf{x}_j|}{\alpha}\right)\frac{(x_iy_j-x_jy_i)}{|\mathbf{x}_i-\mathbf{x}_j|}=0, \tag{4.17}$$

and

$$S_3=\frac{1}{2\pi\alpha^3}\sum_{i=1}^{N}\sum_{j=1}^{N}p_{2,i}p_{2,j}K'_{0,\varepsilon}\left(\frac{|\mathbf{x}_i-\mathbf{x}_j|}{\alpha}\right)\frac{(x_iy_j-x_jy_i)}{|\mathbf{x}_i-\mathbf{x}_j|}=0. \tag{4.18}$$

Here, once again, we used the fact that the term corresponding to $i=j$ in (4.17) vanishes and that the summation is performed over all $i,j=1,\ldots,N$. Finally, combining the results (4.16), (4.17) and (4.18), we conclude that the formula (4.15) holds and the total angular momentum of the described system of particles is conserved.

Translational and rotational symmetries of the particle system. We now show that the system of ODEs (4.3) is invariant under translation and rotation and has momentum maps associated with these actions. These geometrical features are very important properties of the particle system as, for instance, the translational symmetry allows one to perform sympletic reduction and obtain the associated reduced Hamiltonian system and thus may be very useful in analyzing the motion of two interacting peakons in 2-D.

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

As was already mentioned, the system (4.3) can be viewed as a particular case of the system of integro-partial-differential equations (1.3)–(1.4) (see also [26]). The functions $\mathbf{x}_j(t)$ and $\mathbf{p}_j(t)$ satisfy the canonical Hamiltonian equations:

$$\frac{\mathrm{d}\mathbf{x}_j}{\mathrm{d}t} = \frac{\partial H^N}{\partial \mathbf{p}_j}, \qquad \frac{\mathrm{d}\mathbf{p}_j}{\mathrm{d}t} = -\frac{\partial H^N}{\partial \mathbf{x}_j}, \qquad j = 1, \ldots, N, \tag{4.19}$$

where H^N is the Hamiltonian function defined as

$$\begin{aligned} H^N(t; \mathbf{x}, \mathbf{p}) &= \frac{1}{4\pi\alpha^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\mathbf{p}_i \cdot \mathbf{p}_j) K_{0,\varepsilon}\left(\frac{|\mathbf{x}_i - \mathbf{x}_j|}{\alpha}\right) \\ &= \frac{1}{4\pi\alpha^2} \sum_{i=1}^{N} \sum_{j=1}^{N} \left[p_{1,i}p_{1,j} + p_{2,i}p_{2,j}\right] K_{0,\varepsilon}\left(\frac{|\mathbf{x}_i - \mathbf{x}_j|}{\alpha}\right). \end{aligned} \tag{4.20}$$

A symmetry of a Hamiltonian system (M, ω, H) is a function $S : M \to M$ that preserves both the symplectic form ω and the Hamiltonian function H (see, *e.g.*, [37], Def. 22). Here we have

$$M = \mathbb{R}^{4N} = \{(\mathbf{x}, \mathbf{p})\}, \tag{4.21}$$

$$\omega = \sum_{j=1}^{N} \left[\mathrm{d}p_{1,j} \wedge \mathrm{d}x_j + \mathrm{d}p_{2,j} \wedge \mathrm{d}y_j\right], \tag{4.22}$$

and H^N given by (4.20) above.

We first consider the translation action of $(\mathbb{R}^{2N}, +)$ on M: for each $\mathbf{g} = (g_x, g_y)^T$ in $\mathbb{R}^{2N}$, we define the function $S_\mathbf{g} : M \to M$ by

$$S_\mathbf{g}(\mathbf{x}, \mathbf{p}) = S_\mathbf{g}(\mathbf{x} + \mathbf{g}, \mathbf{p}).$$

It can be shown that $S_\mathbf{g}$ preserves the manifold M and its symplectic form ω. Obviously, the Hamiltonian H^N remains unchanged under the translation action, *i.e.* $S_\mathbf{g}^* H^N = H^N$, since $S_\mathbf{g}$ changes only the values of $\mathbf{x}$ and $|\mathbf{x}_i + \mathbf{g} - (\mathbf{x}_j + \mathbf{g})| = |\mathbf{x}_i - \mathbf{x}_j|$. Thus, translation of the coordinate axes is a symmetry of our Hamiltonian system.

Next, we consider the rotational symmetry of the Hamiltonian system (4.21), (4.22) and (4.20). We denote the set of real orthogonal 2×2 matrices of determinant 1 by $SO(2)$, that is,

$$SO(2) = \left\{ \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} : \theta \in \mathbb{R} \right\},$$

and define an action of $SO(2)$ on M as follows. For each $\mathbf{g} \in SO(2)$, we have

$$S_\mathbf{g} : M \to M, (\mathbf{x}, \mathbf{p}) \to (\mathbf{g}\mathbf{x}, \mathbf{p}\mathbf{g}^T).$$

Once again, following [37], one can easily show that $S_\mathbf{g}^* \omega = \omega$. Now let us check the invariance of H^N.

$$\begin{aligned} H^N(t; \mathbf{g}\mathbf{x}, \mathbf{p}\mathbf{g}^T) &= \frac{1}{4\pi\alpha^2} \sum_{i=1}^{N} \sum_{j=1}^{N} ((\mathbf{p}_i\mathbf{g}^T) \cdot (\mathbf{p}_j\mathbf{g}^T)) K_{0,\varepsilon}\left(\frac{\mathbf{g}(\mathbf{x}_i - \mathbf{x}_j)}{\alpha}\right) \\ &= \frac{1}{4\pi\alpha^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (\mathbf{p}_i \cdot \mathbf{p}_j) K_{0,\varepsilon}\left(\frac{|\mathbf{x}_i - \mathbf{x}_j|}{\alpha}\right) = H^N(t; \mathbf{x}, \mathbf{p}) \end{aligned}$$

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

since for any orthogonal matrix $\mathbf{g}$, one has $(\mathbf{u}\cdot\mathbf{w}) = (\mathbf{gu}\cdot\mathbf{gw}) = (\mathbf{ug}^T\cdot\mathbf{wg}^T)$, and therefore

$$((\mathbf{p}_i\mathbf{g}^T)\cdot(\mathbf{p}_j\mathbf{g}^T)) = (\mathbf{p}_i\cdot\mathbf{p}_j),$$

and

$$|\mathbf{g}(\mathbf{x}_i-\mathbf{x}_j)| = \sqrt{(\mathbf{g}(\mathbf{x}_i-\mathbf{x}_j)\cdot\mathbf{g}(\mathbf{x}_i-\mathbf{x}_j))} = \sqrt{((\mathbf{x}_i-\mathbf{x}_j)\cdot(\mathbf{x}_i-\mathbf{x}_j))} = |\mathbf{x}_i-\mathbf{x}_j|.$$

Thus, rotation of the coordinate axes is a symmetry of our Hamiltonian system.

The translational and rotational symmetries are related to conservation of linear momentum and angular momentum, respectively (see [32,37]). Namely, the value of the momentum map associated with the translation action, is equal to the total linear momentum of the system (4.3), that is, an equivariant momentum map $\mathbf{J} : T^*\mathbb{R}^{2N} \to \mathbb{R}^2$ is given by

$$\mathbf{J}(\mathbf{x},\mathbf{p}) = \sum_{j=1}^{N}[p_{1,j}+p_{2,j}]. \tag{4.23}$$

The latter follows from [32], Theorem 12.1.15.

Similarly, the value of the momentum map associated with the rotation action, is equal to the total angular momentum of the system (4.3). Indeed, according to [32], Theorem 12.1.15, an equivariant momentum map $\mathbf{J} : M \to \mathfrak{so}(2)^*$, is given by

$$\mathbf{J}(\mathbf{x},\mathbf{p}) = \mathbf{x}\times\mathbf{p}, \tag{4.24}$$

where $\mathfrak{so}(2)$ is the Lie algebra of $SO(2)$ and * denotes the dual space.

4.2. Numerical examples

In this section, we apply the particle method to the 2-D EPDiff equation (1.1), (1.2) (with $\alpha = 0.5$) and demonstrate the ability of the method to capture different collision scenarios for singular solutions of the EPDiff equation effectively and with a high resolution. First, in Figure 4, we illustrate a peakon-antipeakon collision, as well as an interaction of two peakons, Figure 5, simulated by the particle method. Once again, a single particle with a non-zero weight is used to represent each peakon.

In the next set of numerical experiments, presented in Figures 6–9, we illustrate the emergent and collision properties of peakon wave fronts in 2-D – the phenomena that are observed in nature, for instance, as trains of internal waves in the south China Sea, [31]. The initial data for the examples below (shown in Figs. 6a, 7a, 8a, and 9a) consist of contact discontinuities that are either straight or curved line segments in the plane and were chosen to provide qualitatively similar solutions to those obtained by [28], in which a far more computationally intensive finite differencing approach was used.

In each example, 400 particles were initially distributed uniformly along a codimension one submanifold (see Figs. 6d, 7d, 8d, and 9d), and various geometries and initial momenta are chosen to highlight the ensuing propagation and interaction of the wavefronts. In Figures 6–9, we plot the magnitude of the solution vector, $|\mathbf{u}|$, of (1.1), (1.2) (subfigures (b) and (c)) and the locations of particles (subfigures (e) and (f)) for different sets of initial data and at different times. The mollified kernel (4.9) with $\varepsilon = 0.1$ was used in all simulations. The small number of particles required to recover the solutions dramatically reduces the computation time needed to produce the solutions; each computation required less than 10 seconds on a single 2.5 GHz processor (for comparison, the compatible differencing algorithm (CDA), implemented in [28], requires a resolution of $\Delta x = \Delta y = 1/1024$ to simulate the collision among the curves). During the particle computations, we also

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

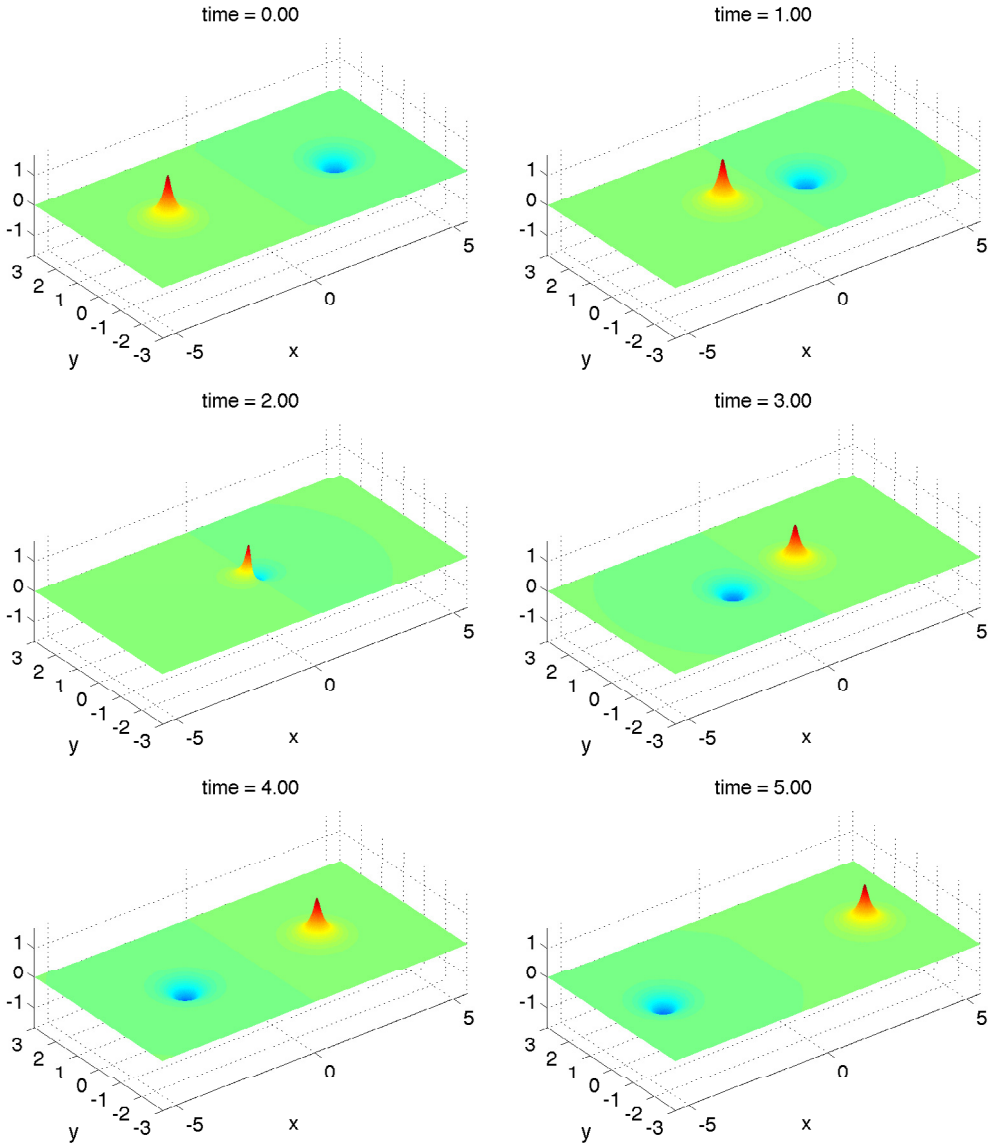


FIGURE 4. 2-D simulation of peakon-antipeakon interactions by the particle method.

monitored the momenta and Hamiltonian to ensure that they are very nearly conserved, see, *e.g.* Figure 10, in which we show the linear and angular momenta for the "star" initial data (presented in Fig. 9) are exactly conserved by the particle method.

5. CONCLUSION

In this paper, we have provided a new way of solving the one- and two-dimensional EPDiff equation. The particle method has been derived using a weak formulation of the problem and successfully applied to a number

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

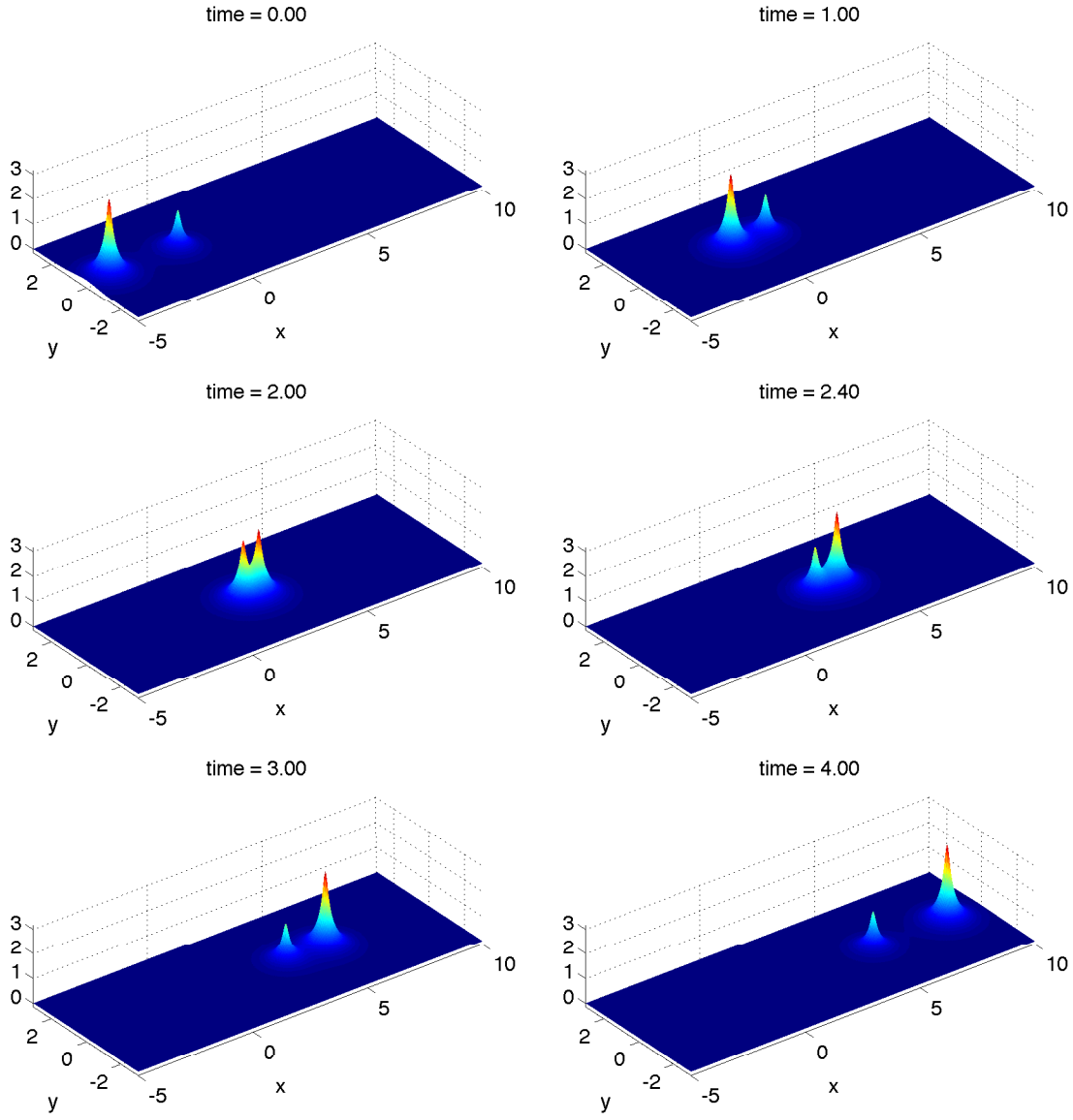


FIGURE 5. 2-D simulation of two-peakon interactions by the particle method.

of problems, in which the initial data of interest lies on a submanifold. An extended version of this paper will be published in the future and will provide additional details and some verification of results that were omitted above. Furthermore, numerical experiments will be performed with arbitrary initial data, which can be approximated by uniformly distributing particles throughout the plane. This will require many more particles to be used in computations as well as their reinitialization as time evolves. The particle method can be nevertheless be efficiently implemented in straightforward fashion in a parallel computing environment and the results will be compared with those obtained by classical numerical methods.

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

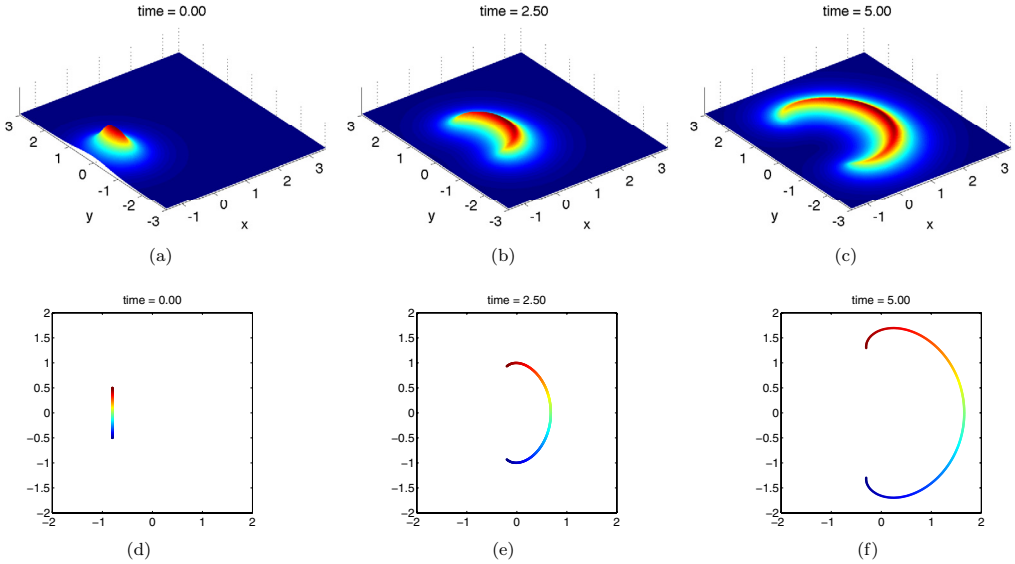


FIGURE 6. A simulation of the "plate" initial condition: 400 particles are initially placed along a straight line of unit length parallel to the y-axis each with an identical momentum component of 1/399 (the inter-particle spacing) in the positive x-direction. The simulation length of 10 time units required 3.9 s of computation on a single 2.5 GHz CPU.

17

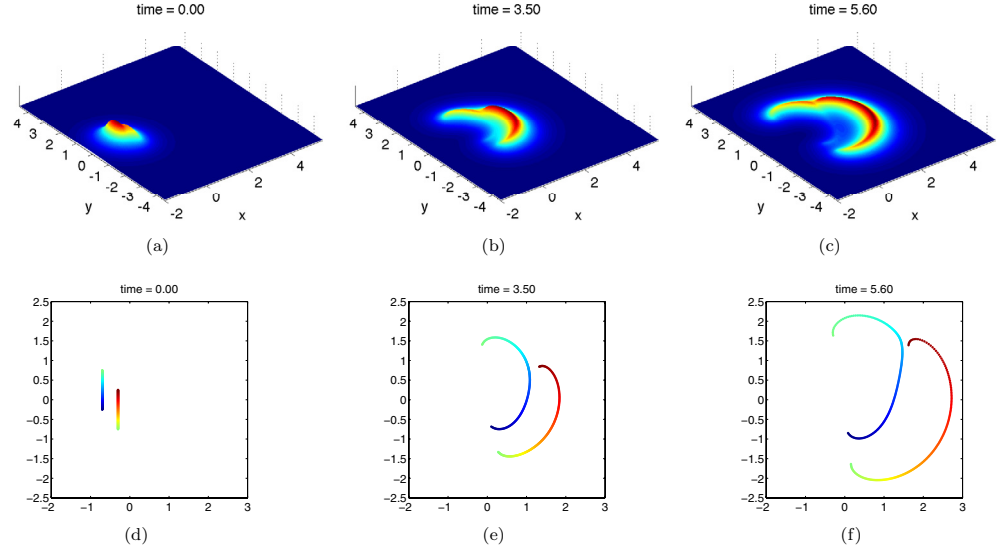


FIGURE 7. A simulation of the "parallel" initial condition: 200 particles are initially placed along each of two parallel lines of unit length parallel to the y-axis and offset slightly from each other in the y-direction. The particles in the left line segment initially each have an initial momentum component of 1/199 in the positive x-direction, while particles in the right line segment have lesser momentum of 0.5/199, also in the x-direction. The simulation length of 12 time units required 4.6 s of computation on a single 2.5 GHz CPU.

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

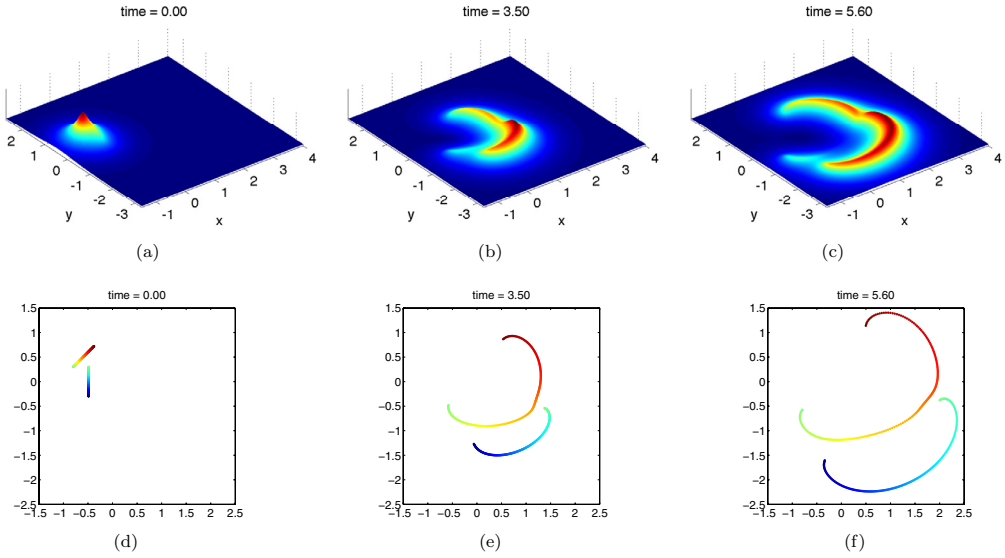


FIGURE 8. A simulation of the "skew" initial condition: 200 particles are initially placed along each of two line segments of length 0.6 set at an angle of 45 degrees to one another. The momenta of the particles is chosen orthogonal to the line segment in which they lie in the positive x-direction. The magnitude of the momenta in the leftmost line segment are chosen greater than the momenta in the rightmost segment by a factor of $2/\sqrt{2}$. The simulation length of 15 time units required 5.9 s of computation on a single 2.5 GHz CPU.

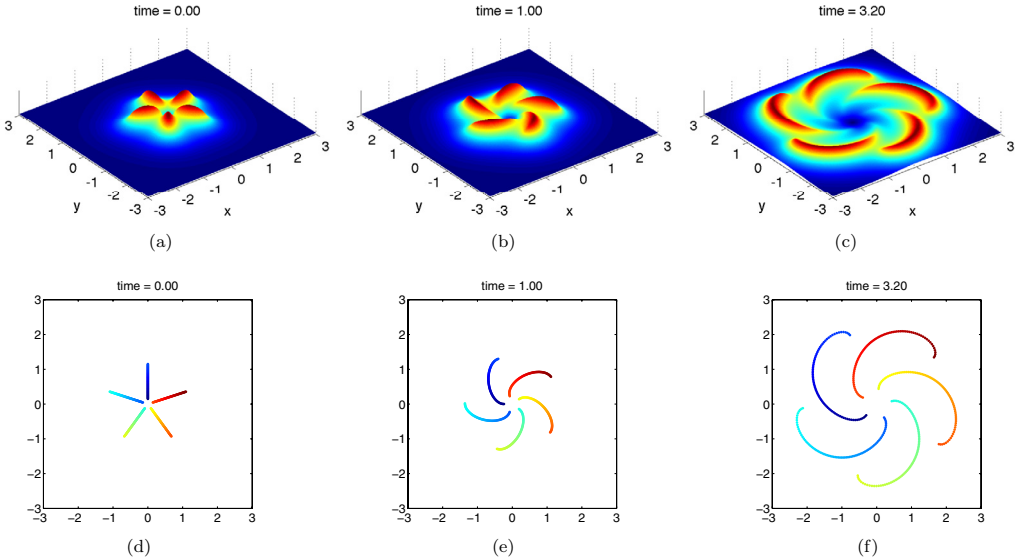
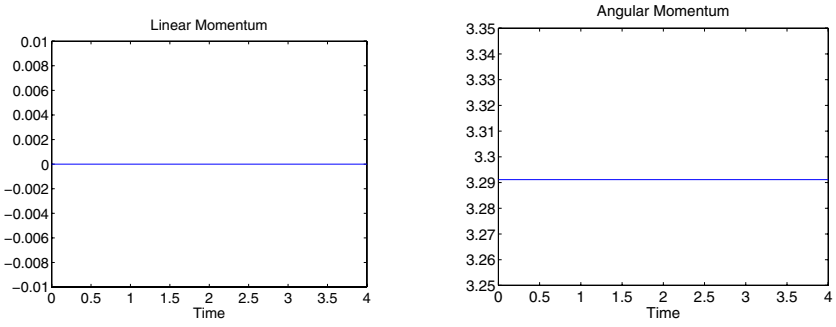


FIGURE 9. A simulation of the "star" initial condition: 80 particles are initially placed along each of the 5 arms forming the star. The initial momentum of each particle is $\sqrt{2}/80$ in the direction orthogonal to the line segment in which it lies, creating a counterclockwise sense of rotation of the star. The simulation length of 4 time units required 6.32 s of computation on a single 2.5 GHz CPU.

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press



(a) (b)

FIGURE 10. Conservation of the total linear (a) and angular (b) momenta during the evolution of the "star" initial data presented in Figure 9.

REFERENCES

[1] M.S. Alber, R. Camassa, Y.N. Fedorov, D.D. Holm and J.E. Marsden, The complex geometry of weak piecewise smooth solutions of integrable nonlinear PDE's of shallow water and Dym type. *Commun. Math. Phys.* **221** (2001) 197–227.

[2] R. Artebrant and H.J. Schroll, Numerical simulation of Camassa-Holm peakons by adaptive upwinding. *Appl. Numer. Math.* **56** (2006) 695–711.

[3] R. Beals, D.H. Sattinger and J. Szmigielski, Peakon-antipeakon interaction. *J. Nonlin. Math. Phys.* **8** (2001) 23–27; Nonlinear evolution equations and dynamical systems, Kolimbary (1999).

[4] R. Camassa and D.D. Holm, An integrable shallow water equation with peaked solitons. *Phys. Rev. Lett.* **71** (1993) 1661–1664.

[5] R. Camassa, D.D. Holm and J.M. Hyman, A new integrable shallow water equation. *Adv. Appl. Mech.* **31** (1994) 1–33.

[6] R. Camassa, J. Huang and L. Lee, On a completely integrable numerical scheme for a nonlinear shallow-water wave equation. *J. Nonlin. Math. Phys.* **12** (2005) 146–162.

[7] R. Camassa, J. Huang and L. Lee, Integral and integrable algorithms for a nonlinear shallow-water wave equation. *J. Comput. Phys.* **216** (2006) 547–572.

[8] S. Chen, C. Foias, D.D. Holm, E. Olson, E.S. Titi and S. Wynne, Camassa-Holm equations as a closure model for turbulent channel and pipe flow. *Phys. Rev. Lett.* **81** (1998) 5338–5341.

[9] A. Chertock and A. Kurganov, On a practical implementation of particle methods. *Appl. Numer. Math.* **56** (2006) 1418–1431.

[10] A. Chertock and D. Levy, Particle methods for dispersive equations. *J. Comput. Phys.* **171** (2001) 708–730.

[11] A. Chertock and D. Levy, A particle method for the KdV equation. *J. Sci. Comput.* **17** (2002) 491–499.

[12] A.J. Chorin, Numerical study of slightly viscous flow. *J. Fluid Mech.* **57** (1973) 785–796.

[13] G.M. Coclite, K.H. Karlsen and N.H. Risebro, A convergent finite difference scheme for the Camassa-Holm equation with general H^1 initial data. *SIAM J. Numer. Anal.* **46** (2008) 1554–1579.

[14] A. Cohen and B. Perthame, Optimal approximations of transport equations by particle and pseudoparticle methods. *SIAM J. Math. Anal.* **32** (2000) 616–636 (electronic).

[15] G.-H. Cottet and P.D. Koumoutsakos, *Vortex methods*. Cambridge University Press, Cambridge (2000).

[16] G.-H. Cottet and S. Mas-Gallic, *A particle method to solve transport-diffusion equations, Part 1: the linear case*. Tech. Report 115, Ecole Polytechnique, Palaiseau, France (1983).

[17] G.-H. Cottet and S. Mas-Gallic, A particle method to solve the Navier-Stokes system. *Numer. Math.* **57** (1990) 805–827.

[18] P. Degond and S. Mas-Gallic, The weighted particle method for convection-diffusion equations. I. The case of an isotropic viscosity. *Math. Comput.* **53** (1989) 485–507.

[19] P. Degond and S. Mas-Gallic, The weighted particle method for convection-diffusion equations. II. The anisotropic case. *Math. Comput.* **53** (1989) 509–525.

[20] P. Degond and F.-J. Mustieles, A deterministic approximation of diffusion equations using particles. *SIAM J. Sci. Statist. Comput.* **11** (1990) 293–310.

[21] S. Gottlieb, C.-W. Shu and E. Tadmor, High order time discretization methods with the strong stability property. *SIAM Rev.* **43** (2001) 89–112.

[22] O.-H. Hald, *Convergence of vortex methods*, *Vortex methods and vortex motion*. SIAM, Philadelphia, PA (1991) 33–58.

[23] A.N. Hirani, J.E. Marsden and J. Arvo, *Averaged Template Matching Equations*, *EMMCVPR, Lecture Notes in Computer Science* **2134**. Springer (2001) 528–543.

[24] H. Holden and X. Raynaud, Convergence of a finite difference scheme for the Camassa-Holm equation. *SIAM J. Numer. Anal.* **44** (2006) 1655–1680 (electronic).

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press

[25] H. Holden and X. Raynaud, A convergent numerical scheme for the Camassa-Holm equation based on multipeakons. *Discrete Contin. Dyn. Syst.* **14** (2006) 505–523.

[26] D.D. Holm and J.E. Marsden, *Momentum maps and measure-valued solutions (peakons, filaments, and sheets) for the EPDiff equation*, *The breadth of symplectic and Poisson geometry, Progr. Math.* **232**. Birkhäuser Boston, Boston, MA (2005) 203–235.

[27] D.D. Holm and M.F. Staley, Wave structure and nonlinear balances in a family of evolutionary PDEs. *SIAM J. Appl. Dyn. Syst.* **2** (2003) 323–380 (electronic).

[28] D.D. Holm and M.F. Staley, *Interaction dynamics of singular wave fronts*, under "Recent Papers" at `http://cnls.lanl.gov/~staley/.`

[29] D.D Holm, J.T. Ratnanather, A. Trouvé and L. Younes, Soliton dynamics in computational anatomy. *NeuroImage* **23** (2004) S170–S178.

[30] H.-P. Kruse, J. Scheurle and W. Du, *A two-dimensional version of the Camassa-Holm equation, Symmetry and perturbation theory*. World Sci. Publ., Cala Gonone, River Edge, NJ (2001) 120–127.

[31] A.K. Liu, Y.S. Chang, M.-K. Hsu and N.K. Liang, Evolution of nonlinear internal waves in the east and south China Sea. *J. Geophys. Res.* **103** (1998) 7995–8008.

[32] J.E. Marsden and T.S. Ratiu, *Introduction to mechanics and symmetry*, *Texts in Applied Mathematics* **17**, 2nd edition. Springer-Verlag, New York (1999).

[33] R.I. McLachlan and P. Atela, The accuracy of symplectic integrators. *Nonlinearity* **5** (1992) 541–562.

[34] R. McLachlan and S. Marsland, N-particle dynamics of the Euler equations for planar diffeomorfism. *Dyn. Syst.* **22** (2007) 269–290.

[35] P.-A. Raviart, An analysis of particle methods. *Numerical methods in fluid dynamics (Como, 1983), Lecture Notes in Math.* **1127**. Springer, Berlin (1985) 243–324.

[36] G.D. Rocca, M.C. Lombardo, M. Sammartino and V. Sciacca, Singularity tracking for Camassa-Holm and Prandtl's equations. *Appl. Numer. Math.* **56** (2006) 1108–1122.

[37] S.F. Singer, *Symmetry in mechanics*: *A gentle, modern introduction*. Birkhäuser Boston Inc., Boston, MA (2001).

[38] Y. Xu and C.-W. Shu, A local discontinuous Galerkin method for the Camassa-Holm equation. *SIAM J. Numer. Anal.* **46** (2008) 1998–2021.

https://doi.org/10.1051/m2an/2011054 Published online by Cambridge University Press