Elastic overtaking collisions of large-amplitude ion-acoustic solitons

Carel P. Olivier [1],†

Pure and Applied Analytics, School of Mathematical and Statistical Sciences, North-West University,
Mmabatho 2735, South Africa

Overtaking collisions of large-amplitude solitons are investigated via fluid simulations for a plasma consisting of cold ions and Boltzmann-distributed electrons. To achieve this, a new fluid simulation code is presented. In addition, a novel approach to construct soliton initial conditions is developed. Using these ideas, initial conditions are combined that allows the simulation of overtaking collisions. It is shown that, in the small-amplitude regime, simulation results agree well with the two-soliton solution obtained from reductive perturbation theory. Interestingly, in the large amplitude regime, both the slow and fast solitons re-emerge after the collision with no significant change, showing that the collisions remain elastic. A comparison between reductive perturbation analysis and the simulations show that the only significant effect of higher order nonlinearities on overtaking collisions is a reduction in the magnitude of the phase shifts of both solitons.

Keywords: plasma nonlinear phenomena

---

1. Introduction

Bipolar electric field structures propagating parallel to the magnetic field have been observed in many regions of Earth's magnetosphere (see for example Hansel *et al.* (2021) and references therein). An interesting feature of these observations is that the structures often appear in large clusters, as is clearly demonstrated by Bale *et al.* (1998) and Bounds *et al.* (1998). These structures are often interpreted as Bernstein–Greene–Kruskal modes obtained from kinetic models (Muschietti *et al.* 1999; Main, Newman & Ergun 2006) or ion-acoustic solitons from fluid models, as argued in the review paper by Lakhina *et al.* (2018). This paper concerns itself with the latter, namely fluid theory. The fact that these pulses have different widths and amplitudes implies that they propagate with different velocities. As such, one may expect frequent overtaking collisions to occur when the faster solitons overtake the slower ones.

In fluid theory, there are two theoretical approaches to study soliton solutions. The first approach is reductive perturbation theory (RPT) that was introduced by Washimi & Taniuti (1966), where Korteweg–deVries (KdV)-type equations are derived. These equations govern small-amplitude solitons. The study allows for the construction of

† Email address for correspondence: carel.olivier@nwu.ac.za

so-called *N*-soliton solutions, allowing one to study collisions in the small-amplitude regime. These solutions show that overtaking collisions are elastic in the small-amplitude regime.

The second theoretical approach to study solitons in fluid models is the so-called Sagdeev pseudopotential analysis that was first applied by Sagdeev (1966). The advantage of this methodology is its ability to construct single-soliton solutions without small-amplitude restrictions. However, the method is formulated in a way that excludes the possibility to consider collisions Therefore, to study overtaking collisions in the large-amplitude regime, one must resort to experiments or simulations.

In recent years, fluid simulation studies of solitons have gained much traction. Most of these studies focus on two aspects of soliton dynamics, namely wave-breaking (Kakad, Omura & Kakad 2013; Kakad & Kakad 2016; Lotekar, Kakad & Kakad 2017) and generation mechanisms of solitons from initial ion number density disturbances (Kakad, Kakad & Omura 2014; Kakad, Lotekar & Kakad 2016; Lotekar, Kakad & Kakad 2016; Kakad *et al.* 2019; Lotekar, Kakad & Kakad 2019*b*; Singh *et al.* 2021, 2022; Guo *et al.* 2023). For the latter aspect, results show that such disturbances are sufficient to generate solitons in a wide range of plasma models. Moreover, Lotekar *et al.* (2016) showed that supersolitons can also be generated from a generalized initial disturbance in appropriate models.

As was alluded to in the introductory paragraph, another important aspect of soliton dynamics is their collisions. This topic has received less attention to date in terms of fluid simulation studies. Previous studies of this topic are limited to head-on collisions, that is, collisions where solitons move in opposite directions (Lotekar, Kakad & Kakad 2019*a*). However, to the best of my knowledge, fluid simulation studies of overtaking soliton collision have not been undertaken to date. Nevertheless, the topic of overtaking collisions has been studied via kinetic simulations (Jenab & Spanier 2016) and particle-in-cell (PIC) simulations (Sharma, Sengupta & Sen 2015). In the latter case, solitons were generated through solutions obtained from KdV solutions. An unfortunate consequence of this approach is that in the large-amplitude regime, the initial disturbances undergo a steepening process before the final form of the soliton emerges while producing a disturbance in the wake of the soliton. This complicates the analysis of the effect of the collision, as it is difficult to distinguish between the effects of steepening, tail-formation and collisional effects. As a result, the topic of the elasticity of overtaking collisions of large-amplitude solitons received little attention from Sharma *et al.* (2015).

To address the issue of initial steepening and tail-formation, this paper introduces a novel approach based on the works of Olivier, Verheest & Maharaj (2017) and Olivier, Verheest & Hereman (2018) to construct soliton solutions directly via Sagdeev pseudopotential analysis (Sagdeev 1966). The advantage of this approach is that its solutions retain the full nonlinearity of the fluid equations. This eliminates the problem of initial steepening. In addition, it allows one to simulate solitons with larger amplitudes than those obtained from KdV approximations or other initial disturbances.

The purpose of this paper is to use this new approach to study the elasticity of overtaking collisions of large-amplitude solitons. To perform the simulation, we constructed a fluid simulation code that is conceptually similar to those used in earlier studies (Lotekar *et al.* 2016, 2019*a*), with three modifications, namely that the second-order bootstrap time integration method is replaced by a fourth-order accurate Runge–Kutta method, a fourth-order accurate spatial derivative approximation is used to solve Poisson's equation instead of a second-order approximation and Newton's method is used to solve the nonlinear system of equations resulting from the discretized Poisson's equation, rather than the successive-over-relaxation (SOR) method (Young 2014). It should be noted that

the use of Newton's method to solve the Poisson equation has been applied successfully in PIC simulation studies (Sharma *et al.* 2015).

The paper is structured as follows. In § 2, we present a standard fluid model consisting of cold ions and Boltzmann electrons. We also obtain soliton solutions by means of RPT and Sagdeev pseudopotential analysis. Section 3 provides a brief overview of the numerical scheme, supplemented by more details in Appendices A–C. In § 4, the fluid simulation code is validated through simulations of single-soliton solutions. In § 5, the results of the simulation of overtaking soliton collisions are presented. Conclusions are discussed in § 6.

2. Fluid model and analytical solutions

We now proceed to introduce the fluid model that is studied in this paper. In addition, we also provide an overview of soliton solutions obtained by means of RPT and Sagdeev pseudopotential analysis.

2.1. *Fluid model*

The fluid equations are governed by the normalized continuity equation,

$$\frac{\partial n}{\partial t} + \frac{\partial}{\partial x}(nu) = 0, \tag{2.1}$$

the normalized momentum equation,

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{\partial \phi}{\partial x} = 0, \tag{2.2}$$

and the normalized Poisson equation,

$$\frac{\partial^2 \phi}{\partial x^2} + n - e^\phi = 0. \tag{2.3}$$

In the normalized fluid equations, n denotes the ion number density normalized with respect to the equilibrium ion density n_{i0}. The ion fluid velocity is represented by u and is normalized with respect to the ion-acoustic speed for Boltzmann electrons $C_{IA} = (k_B T_e/m_i)$, where m_i denotes the ion mass, and the electrostatic potential ϕ is normalized with respect to $k_B T_e/e$, where k_B is the Boltzmann constant, T_e is the electron temperature and e is the electron charge. In addition, the time variable t is normalized with respect to the inverse ion plasma frequency $\omega_{pi}^{-1} = (m_i \varepsilon_0/n_{i0}e^2)^{1/2}$, where ε_0 is the permittivity of free space. The spatial variable x is normalized with respect to the Debye length $\lambda_D = (\varepsilon_0 k_B T_e/n_{i0}e^2)^{1/2}$.

2.2. *Reductive perturbation analysis*

In the following, we use the results obtained by Washimi & Taniuti (1966). Rather than producing a full derivation, we report the important definitions and solutions used in this study.

For reductive perturbation analysis, we introduce the following perturbation expansions:

$$n = 1 + \varepsilon n_1 + \varepsilon^2 n_2 + \cdots, \tag{2.4}$$

$$u = \varepsilon u_1 + \varepsilon^2 u_2 + \cdots, \tag{2.5}$$

$$\phi = \varepsilon \phi_1 + \varepsilon^2 \phi_2 + \cdots, \tag{2.6}$$

along with the moving frame coordinates

$$\xi = \varepsilon^{1/2}(x - t), \quad \tau = \varepsilon^{3/2}t. \qquad (2.7a,b)$$

By substituting these expansions, as well as a Taylor series expansion of the function e^ϕ in Poisson's equation, it follows that the electrostatic potential ϕ_1 satisfies the following KdV equation:

$$\frac{\partial \phi_1}{\partial \tau} + \frac{1}{2}\frac{\partial^3 \phi_1}{\partial \xi^3} + \phi_1 \frac{\partial \phi_1}{\partial \xi} = 0, \qquad (2.8)$$

while $n_1 = u_1 = \phi_1$.

For the purpose of this paper, we are interested in two solutions. The first of these is the single soliton solution, given by

$$\phi_1(\xi, \tau) = 3c\, \mathrm{sech}^2 \left[\sqrt{\frac{c}{2}}(\xi - c\tau) \right]. \qquad (2.9)$$

To express the solution in the original coordinates, we note that $\phi \approx \varepsilon\phi_1$. By setting $c = 1$ and using the original coordinates as expressed in (2.7a,b), one obtains the following solution for the electrostatic potential:

$$\phi(x, t) = 3\varepsilon\, \mathrm{sech}^2 \left[\sqrt{\frac{\varepsilon}{2}}(x - (1 + \varepsilon)t) \right]. \qquad (2.10)$$

It follows that the amplitude of the soliton is given by $A = 3\varepsilon$, while the speed is given by $M = 1 + \varepsilon$. Therefore, for any choice of $M > 1$, one can determine the value of ε to determine the solution. However, it should be noted that the approximation works best for $\varepsilon \ll 1$. In addition, since $n_1 = \phi_1$ and $u_1 = \phi_1$, one obtains the following solutions for the number density and ion fluid velocity:

$$n = 1 + 3\varepsilon\, \mathrm{sech}^2 \left[\sqrt{\frac{\varepsilon}{2}}(x - (1 + \varepsilon)t) \right], \qquad (2.11)$$

$$u = 3\varepsilon\, \mathrm{sech}^2 \left[\sqrt{\frac{\varepsilon}{2}}(x - (1 + \varepsilon)t) \right]. \qquad (2.12)$$

The second solution we consider is the two-soliton solution. In particular, we consider the two-soliton solution where the excess velocity of the fast soliton is twice as large as the excess velocity of the slow soliton. Here, the excess velocity is the speed in excess of the ion acoustic speed. More specifically, if the speed of the soliton is given by M, the excess speed is given by $M - 1$. The solution is given by

$$\phi_1(\xi, \tau) = 12 \frac{2\cosh^2 \dfrac{\eta_1}{\sqrt{2}} + \sinh^2 \eta_2}{\left[a_- \cosh\left(\eta_2 + \dfrac{\eta_1}{\sqrt{2}}\right) + a_+ \cosh\left(\eta_2 - \dfrac{\eta_1}{\sqrt{2}}\right) \right]^2}, \qquad (2.13)$$

where $a_\pm = \sqrt{2} \pm 1$ and $\eta_j = \xi - jt$ for $j = 1, 2$. We can express the solution in terms of the original coordinates, so that

$$\phi(x, t) = 12\varepsilon \frac{2\cosh^2 \zeta_1 + \sinh^2 \zeta_2}{\left[a_- \cosh(\zeta_2 + \zeta_1) + a_+ \cosh(\zeta_2 - \zeta_1) \right]^2}, \qquad (2.14)$$

where $\zeta_1 = \sqrt{\varepsilon/2}(x - (1 + \varepsilon t))$ and $\zeta_2 = \sqrt{\varepsilon}(x - (1 + 2\varepsilon)t)$. During periods of the solution where $|t| \gg 1$, the fast soliton propagates with a speed of $M_f = 1 + 2\varepsilon$, while the slow soliton propagates with speed $M_s = 1 + \varepsilon$. In addition, the corresponding solutions for the ion density and ion fluid velocity are given by

$$n(x, t) = 1 + 12\varepsilon \frac{2 \cosh^2 \zeta_1 + \sinh^2 \zeta_2}{\left[a_- \cosh(\zeta_2 + \zeta_1) + a_+ \cosh(\zeta_2 - \zeta_1)\right]^2} \tag{2.15}$$

and

$$u(x, t) = 12\varepsilon \frac{2 \cosh^2 \zeta_1 + \sinh^2 \zeta_2}{\left[a_- \cosh(\zeta_2 + \zeta_1) + a_+ \cosh(\zeta_2 - \zeta_1)\right]^2}, \tag{2.16}$$

respectively.

As an example, consider the two-soliton solution (2.14) with $\varepsilon = 0.1$, corresponding to the collision of two solitons, where the slow and fast solitons propagate with speeds $M_s = 1.1$ and $M_f = 1.2$, respectively. In figure 1, we show different representations of the solution. In figure 1(*a*), the solution is shown in terms of the original coordinates by using the perturbation expansions (2.5) and transformations (2.7*a,b*). Unfortunately, the width of the solitons is small relative to the total spatial domain so that the resulting solutions only produce thin lines that provide little detail. To gain a better perspective, we plot the solutions in terms of moving frames coordinate defined in terms of the slow and fast soliton speeds, defined as

$$\xi_s = x - M_s t, \quad \xi_f = x - M_f t. \tag{2.17a,b}$$

Figure 1(*b*) shows the solution plotted with respect to the slow soliton time frame ξ_s. Notice that prior to the collision, the slow soliton (light green line) is vertical, indicating that its speed coincides with the moving frame. After the collision, we see that this line shifts to the left, indicative of the phase shift that results from the collision. Following the collision, the shifted curve remains vertical, indicating that the speed after the collision is unaffected. Similarly, figure 1(*c*) shows the propagation of the fast soliton (yellow curve) to remain unchanged after the collision, except for a phase shift to the right. Due to the full recovery of both solitons after the collisions, these collisions are referred to as elastic collisions.

2.3. *Sagdeev pseudopotential analysis*

An important aspect of reductive perturbation analysis is the fact that it is limited to the small-amplitude regime. For larger amplitudes, higher order nonlinear effects become significant and even dominant for large amplitudes. For such solutions, Sagdeev pseudopotential analysis provides an alternative approach that retains the full nonlinearity of the system. However, the resulting analysis is only relevant to the construction of single-soliton solutions.

Sagdeev pseudopotential analysis introduces a moving frame of the form

$$\xi = x - Mt, \tag{2.18}$$

where M is the propagation speed. The idea is then to look for solutions that remain constant in this frame of reference. A necessary condition for obtaining soliton solutions is that one must have a propagation speed that exceeds the acoustic speed, that is, $M > 1$. In addition, we impose asymptotic boundary conditions that ensure that the plasma returns to the equilibrium far away from the soliton, given by the following set of boundary
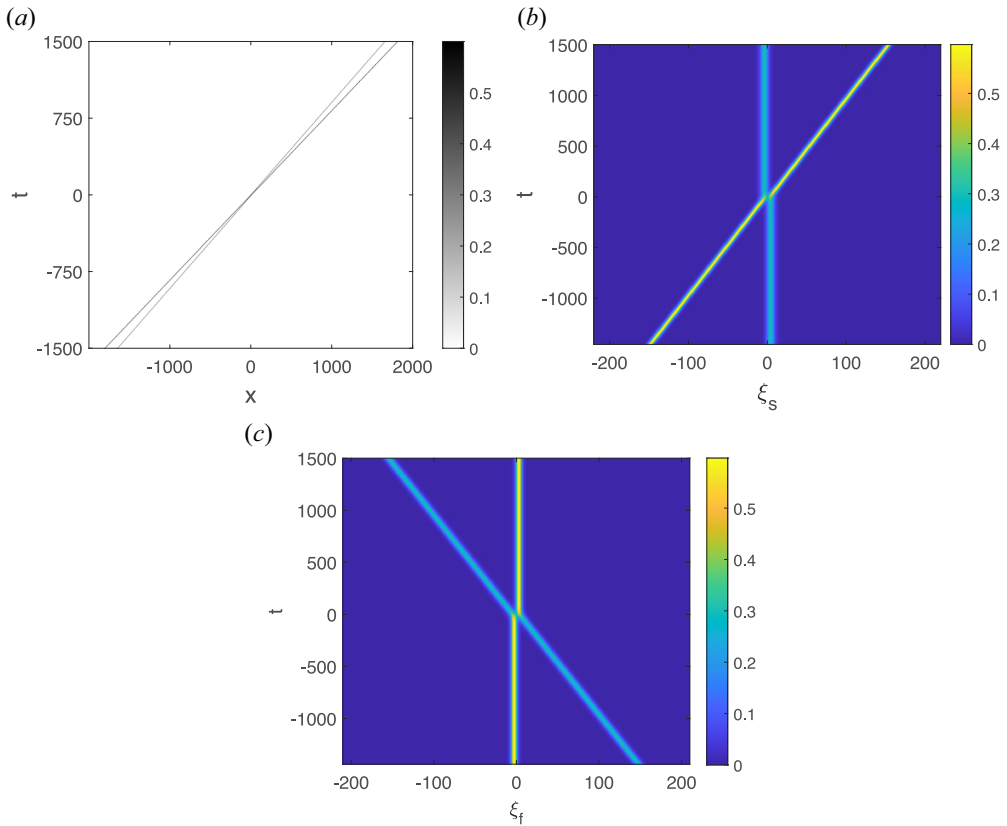
FIGURE 1. Different graphical representations of the time evolution of the electrostatic potential ϕ corresponding to the two-soliton solution with $\varepsilon = 0.1$. (a) Solution plotted in terms of the original coordinates of x and t. (b) and (c) Solutions plotted in the moving frames ξ_s and ξ_f, respectively.

conditions: $n \to 1$, $u \to 0$ and $\phi \to 0$ when $|\xi| \to \infty$. By substituting the moving frame variable into the continuity equation and performing a straightforward integration yields

$$u = M - \frac{M}{n}. \tag{2.19}$$

Similarly, the momentum equation can be integrated in a straightforward manner. By using (2.19) to eliminate u, one obtains

$$n = \frac{1}{\sqrt{1 - \dfrac{2\phi}{M^2}}}. \tag{2.20}$$

Substitution of (2.20) into (2.19) allows us to also express the fluid velocity u in terms of ϕ, given by

$$u = M - \sqrt{M^2 - 2\phi}. \tag{2.21}$$

The final step is to use Poisson's equation to obtain the electrostatic potential ϕ. By substituting the moving frame variable into (2.3), Poisson's equation becomes

$$\frac{\mathrm{d}^2\phi}{\mathrm{d}\xi^2} = \mathrm{e}^\phi - \frac{1}{\sqrt{1 - \dfrac{2\phi}{M^2}}}. \tag{2.22}$$

By multiplying this equation by $\mathrm{d}\phi/\mathrm{d}\xi$, integrating and applying the boundary conditions, one obtains the following first-order ordinary differential equation (ODE) in the form of an energy integral:

$$\frac{1}{2}\left(\frac{\mathrm{d}\phi}{\mathrm{d}\xi}\right)^2 + V(\phi) = 0, \tag{2.23}$$

where the Sagdeev potential $V(\phi)$ is given by

$$V(\phi) = M^2\left[1 - \sqrt{1 - \frac{2\phi}{M^2}}\right] + 1 - \mathrm{e}^\phi. \tag{2.24}$$

Unfortunately, (2.23) and (2.24) cannot be integrated analytically. However, one can integrate the equation numerically to construct the soliton solutions. Once the electrostatic potential is constructed numerically, one can substitute the solutions into (2.20) and (2.21) to obtain the ion number density and ion fluid velocity, respectively The unstable nature of the solutions of (2.23) leads to some inaccuracies for the region where $|\phi| \ll 1$. To deal with this, one can apply an asymptotic analysis, as discussed in § 4. Throughout this paper, this novel approach is used to construct initial conditions for solitons.

2.4. *Higher-order effects on single-soliton solutions*

For solitons with small amplitudes and speeds only marginally faster than the acoustic speed $M_a = 1$, the difference between the single-soliton solutions obtained from the KdV equation and the Sagdeev pseudopotential are very small. Gradually, as the soliton amplitude increases, the differences become increasingly more apparent, thus indicating that higher order nonlinear effects become significant. To illustrate this, let us consider the number densities obtained for solutions with small amplitude that arises when $M = 1.01$ and a relatively large amplitude that arises when $M = 1.3$. The results are shown in figure 2. In panel (*a*), we see that the solution obtained from the KdV equation agrees well with the Sagdeev potential. The two sets of solutions are almost identical except at the peak, where we see that the KdV solution slightly underestimates the maximum number density. However, in panel (*b*), we see a dramatic difference between the KdV approximation and the exact solution obtained from the Sagdeev approach. We see that the solution obtained from the RPT overestimates the width, but significantly underestimates the peak density.

This leads to the question: how does the increase in peak density affect overtaking collisions of solitons? In particular, two-soliton solutions of the KdV equations show that overtaking soliton collisions are elastic, with collisions only resulting in a phase shift. For large-amplitude soliton collisions, are these collision properties still valid or do higher order nonlinear effects, such as larger peak ion densities, lead to inelastic collisions? Since there is no analytical approach available to study this question, we investigate this problem by means of numerical simulation.
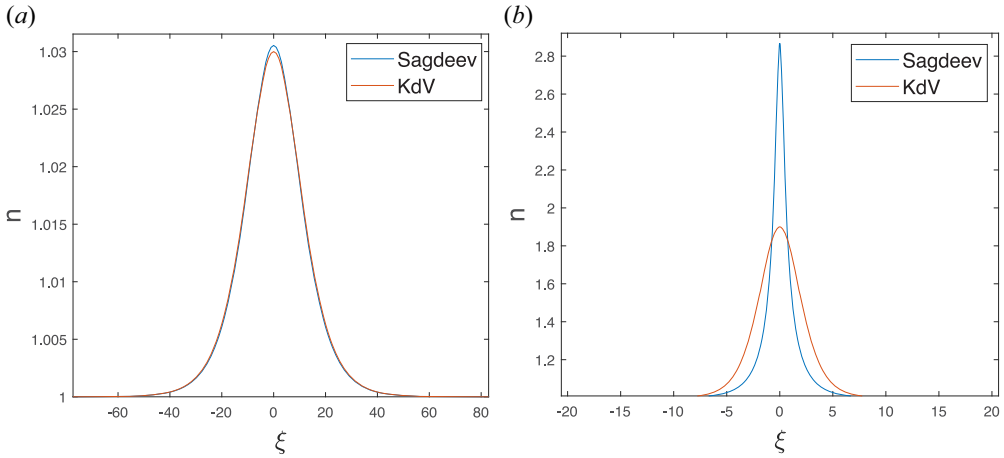
FIGURE 2. Comparison of solutions for the ion number density obtained from Sagdeev pseudopotential analysis (blue curves) and RPT (red curves): (*a*) solutions for $M = 1.01$; (*b*) solutions for $M = 1.3$.

3. Numerical scheme

The main purpose of this fluid code is to simulate solitons that are typically defined on the interval $x \in \mathbb{R}$. As such, the first step is to truncate the interval to a large but finite interval $x \in [-L/2, L/2]$. In addition, we introduce periodic boundary conditions. This approach is typical in soliton simulations.

The next step is to introduce a discretization. To discretize the spatial domain, we introduce a grid of $N + 1$ equidistant points on the interval, denoted by

$$x_i = -\frac{L}{2} + i\Delta x, \quad i = 0, 1, \ldots, N - 1. \tag{3.1}$$

Here $\Delta x = L/N$ denotes the spatial step size. Since periodic boundary conditions are used, all function values at x_0 are equal to function values at x_N, so that the function values at this grid point do not need to be approximated.

In a similar way, we discretize the time domain by introducing a time step size of Δt and denote the jth time step as

$$t_j = j\Delta t. \tag{3.2}$$

To ensure that the Courant–Friedrichs–Lewy condition (Courant, Friedrichs & Lewy 1928) is satisfied, we choose $\Delta t = 0.1\Delta x$ throughout the paper.

Having fully discretized the domain, we introduce vectors to denote the different function value approximations at different time steps. To this end, let $n_{i,j} = n(x_i, t_j)$ and let

$$\boldsymbol{n}^{(j)} = \left[n_{0,j}, n_{1,j}, \ldots, n_{N-1,j} \right]^{\mathrm{T}} \tag{3.3}$$

denote the approximate solution at all the grid points when $t = t_j$. Using similar notation, we introduce vectors for the fluid velocity and electrostatic potential, given by

$$\boldsymbol{u}^{(j)} = \left[u_{0,j}, u_{1,j}, \ldots, u_{N-1,j} \right]^{\mathrm{T}} \tag{3.4}$$

and

$$\boldsymbol{\phi}^{(j)} = \left[\phi_{0,j}, \phi_{1,j}, \ldots, \phi_{N-1,j} \right]^{\mathrm{T}}, \tag{3.5}$$

respectively.

To ensure that all spatial approximations are fourth-order accurate, we use the finite difference formula

$$\frac{\partial f}{\partial x}(\bar{x}, t) = \frac{f(\bar{x} - 2\Delta x, t) - 8f(\bar{x} - \Delta x, t) + 8f(\bar{x} + \Delta x, t) - f(\bar{x} + 2\Delta x, t)}{12\Delta x} + \mathcal{O}\left(\Delta x^4\right)$$

(3.6)

to approximate the spatial derivatives in the continuity and momentum equations (2.1) and (2.2), respectively. For the spatial derivative in Poisson's equation, we use the approximation

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2}(\bar{x}, t) = \frac{1}{12\Delta x^2}\big[&-f(\bar{x} - 2\Delta x, t) + 16f(\bar{x} - \Delta x, t) - 30f(\bar{x}, t) \\ &+ 16f(\bar{x} + \Delta x, t) - f(\bar{x} + 2\Delta x, t)\big] + \mathcal{O}\left(\Delta x^4\right).\end{aligned}$$

(3.7)

A crucial element of the numerical scheme is to solve the boundary value problem associated with the Poisson equation. We follow a similar approach to that used by previous schemes, namely by using finite difference approximations to obtain a nonlinear system of equations. There are two notable changes however. First, we use a fourth-order finite difference approximation (3.7) to approximate the second-order derivative, an improvement on the second-order approximation used previously. Second, we use Newton's method to solve the nonlinear system of equations, as opposed to the SOR method used previously. The details of this approach are discussed in Appendix A.

To solve the equations of continuity and momentum, we use a method of lines approach to yield a system of ODEs. The resulting system of ODEs is solved through a fourth-order Runge–Kutta method, once again improving (in terms of accuracy) on the second-order bootstrap method of earlier codes. The details of this approach are discussed in Appendix B. In addition, some important steps were taken to improve the efficiency of the code, as detailed in Appendix C.

4. Single-soliton simulation

Before we investigate soliton collisions, it is important to demonstrate the ability to simulate single-soliton solutions. This not only validates the accuracy of the numerical simulation code, but also tests the construction of the soliton initial conditions. A crucial component of such simulations is the ability to construct the soliton solution on an interval of arbitrary length by means of an asymptotic analysis. Once we have established this construction, we show the results obtained for a single soliton with speed $M = 1.1$.

4.1. *Constructing soliton initial conditions on arbitrary interval lengths*

Despite the analytical form for the Sagdeev potential, (2.23) has no closed form solutions. As such, one must integrate (2.23) numerically to obtain the solution of ϕ. To do so, we consider the initial value problem given by (2.22) with initial conditions given by $\phi(0) = \phi_r$ and $d\phi/d\xi = 0$, where ϕ_r is the positive root of the Sagdeev potential, that is, $V(\phi_r) = 0$ with $\phi_r > 0$. Due to the symmetry of soliton solutions, it then follows that $\phi(\xi) = \phi(-\xi)$, so that the solution only has to be constructed for $\xi > 0$ and can then be reflected about the ϕ axis to produce the remainder of the solution.

One issue that arises from this approach is that the solution of the ODE initial value problem is unstable. As a result, numerical and round-off errors lead to inaccuracies in the limit when $\phi \to 0$. To understand this problem, it is useful to refer to the potential well analogy of (2.23), where the Sagdeev potential acts as a frictionless potential well for a fictitious particle. For the particle to finish on the top of the potential well, the

solution must be solved exactly, otherwise the particle will either fall back into the well (underestimation) or fall off on the other side of the well (overestimation).

To deal with the tails, we briefly review the results of Olivier *et al.* (2017), where an asymptotic analysis was used to compare the tails of regular solitons with those of acoustic speed solitons. The idea is to use a Taylor series analysis to fit a parabola for $|\phi| \ll 1$. Since $V(0) = V'(0) = 0$, it follows that the Taylor series expansion of V about $\phi = 0$ is given by

$$V(\phi) = \tfrac{1}{2} V''(0) \phi^2 + \mathcal{O}(\phi^3), \tag{4.1}$$

so that we can approximate the Sagdeev potential as

$$V(\phi) \approx -\frac{M^2 - 1}{2M^2} \phi^2, \tag{4.2}$$

provided that $|\phi| \ll 1$. By substituting this approximation into (2.23), one can easily integrate the equation analytically to obtain

$$\phi(\xi) = C \exp\left(\pm\sqrt{\frac{M^2 - 1}{M^2}}\xi\right), \tag{4.3}$$

where C is an integration constant. To construct the initial condition, we solve the initial value problem (2.22) numerically until one observes a sufficiently small solution of ϕ. One then substitutes the given ξ value, say ξ_0 into (4.3), to determine the constant of integration C. Clearly for $\xi \to +\infty$, one must use the lower minus sign in the exponent, so that

$$C = \phi_0 \exp\left(\sqrt{\frac{M^2 - 1}{M^2}}\xi_0\right). \tag{4.4}$$

As an example, we show the Sagdeev potential for $M = 1.1$ in figure 3(*a*). Here the red dot represents a pseudoparticle for illustrative purposes. Now, if the particle is released from this position, the absence of friction will mean that it will approach the local maximum of the potential well at the origin, and the position ϕ will approach zero as ξ approaches infinity, as ξ plays the role of time in the analogy. However, the solution obtained from numerically integrating (2.22) is shown in figure 3(*b*) by the blue curve. Here we see that the solution behaves appropriately for $0 \leqslant \xi \leqslant 40$. However, at some point, the value of ϕ starts to increase. This corresponds to the particle returning to its original position as if it does not have enough energy to reach the top. However, as stated earlier, this is due to numerical errors and the unstable nature of the solution. Indeed, numerically, one always observes either that the particle returns to $\phi \approx \phi_r$ or that the particle overshoots the origin, resulting in $\phi \to -\infty$. The red curve shows the solution obtained by fitting the asymptotic tail at $\phi \approx 10^{-4}$. Here, the curve decreases exponentially as one would expect. The resulting solution can be constructed for the necessary interval length without any stability issues.

4.2. *Results of single-soliton simulation*

To simulate the soliton solutions, we use the solution obtained through integrating Poisson's equation numerically, along with the tail fitted by means of the asymptotic analysis, as the initial condition electrostatic potential $\phi(x, 0)$. This solution is then substituted into (2.20) and (2.21) to obtain the initial number density, $n(x, 0)$, and the initial fluid velocity, $u(x, 0)$, respectively. Throughout all simulations, we choose
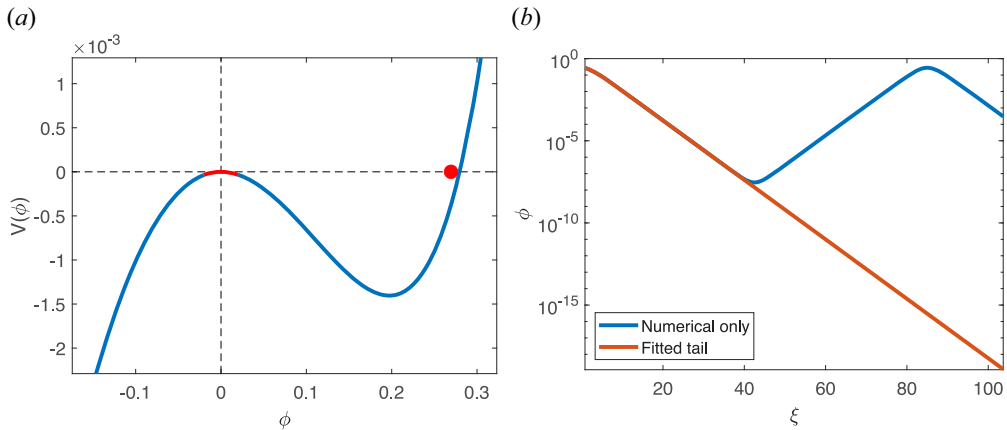
(a)

(b)

FIGURE 3. Construction of the tail of the soliton initial condition. In panel (*a*), the Sagdeev pseudopotential well is shown with the blue curve, while the red dot indicates the fictitious particle. The red part of the potential near the origin shows the part of the curve where the asymptotic expansion is applied. In panel (*b*), the blue curve shows the solution obtained from numerical integration only, while the red curve shows the soliton solution obtained by fitting the asymptotic tail.

$\Delta t = 0.1 \Delta x$ to ensure numerical stability while also satisfying the Courant–Friedrichs–Lewy condition.

In figure 4(*a*), the solution is shown for the initial condition associated with $M = 1.1$ for $0 \leqslant t \leqslant 100$. Here, an interval length of $L = 100$ is used, with $\Delta x = 0.1$. It is clear that the soliton propagates with a fixed speed and without changing form, as expected from a soliton solution. Notice that the solution re-emerges on the left of the domain at $t \approx 45$. This is due to the incorporation of periodic boundary conditions. In figure 4(*b*), the same solution is shown relative to the moving frame $\xi = x - Mt$, where $M = 1.1$. The advantage of this representation is that the analytical solution remains stationary in this frame. The vertical nature of the yellow curve indicates that the constant speed of the soliton is recreated numerically. In figure 4(*c*), we plot the absolute error of the solution at $t = 100$. Here, the absolute error is given by the absolute difference between the solution at $t = 100$ plotted in the moving frame and the initial condition $\phi(x, 0)$, that is,

$$\text{Absolute error} = |\phi(x - 110, 100) - \phi(x, 0)|. \tag{4.5}$$

The maximum of the absolute error is five orders of magnitude smaller than the soliton amplitude. This validates both the numerical scheme and the construction of the soliton initial conditions.

5. Overtaking soliton collision simulations

We have now established all the elements necessary to simulate overtaking collisions of solitons. To do this, we construct the solution of the slow soliton and a fast soliton using the Sagdeev pseudopotential method. Once both solutions are available, we shift the initial condition of the fast solution sufficiently far to the left of the origin to ensure that there is almost no overlap with the slow soliton. The two solutions are then added together to form the initial condition for the collision. To start off, we consider an overtaking collision in the small-amplitude regime. We then consider an overtaking collision in the large-amplitude
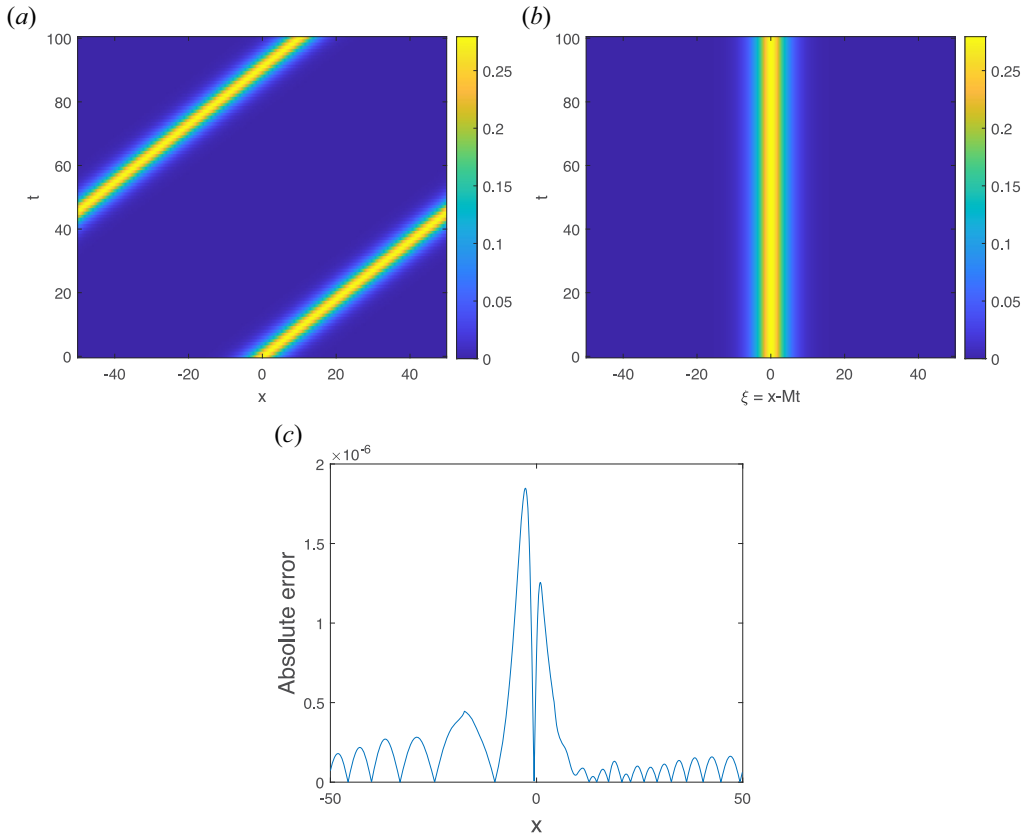
(*a*)

(*b*)

(*c*)

FIGURE 4. Simulation results for a soliton simulation with speed $M = 1.1$. (*a*) Full solution in the laboratory coordinates. (*b*) Solution in the moving frame $\xi = x - Mt$. (*c*) Absolute error for the solution at $t = 100$.

regime. Finally, we will discuss the effect of amplitude on the phase shifts associated with soliton collisions.

5.1. *Small-amplitude soliton collision $M_s = 1.01$*

To start off, we consider the simulation results from a collision between two solitons, where the slow soliton propagates with a speed of $M_s = 1.01$ and the fast soliton has a speed of $M_f = 1.02$. The amplitudes of the two solitons are given by $\phi_s \approx 0.029777$ and $\phi_f \approx 0.059117$, respectively. These values agree well with the corresponding values associated with reductive perturbation analysis, given by $\phi_s = 0.03$ and $\phi_f = 0.06$, respectively. This is to be expected, as both solitons are in the small-amplitude regime.

The large widths of the solitons require a sufficiently large choice of truncated interval length L to avoid soliton overlap at the initial condition. Here we used a truncated interval length $L = 1000$. Conversely, the small gradients associated with the large widths allow us to use a relatively sparse spatial grid. For $M_s = 1.01$, the simulations provided accurate results for a spatial step size of $\Delta x = 1$ and a temporal step size of $\Delta t = 0.1$. Since the difference between the speeds is small, the time integration must be performed over a substantial time period. Here, we solved the solution for $0 \leqslant t \leqslant 60\,000$.

The result of the simulation is shown in figure 5. In figure 5(*a*), the solution is shown in terms of the slow moving frame $\xi_s = x - M_s t$. After the collision at $t \approx 30\,000$, we see
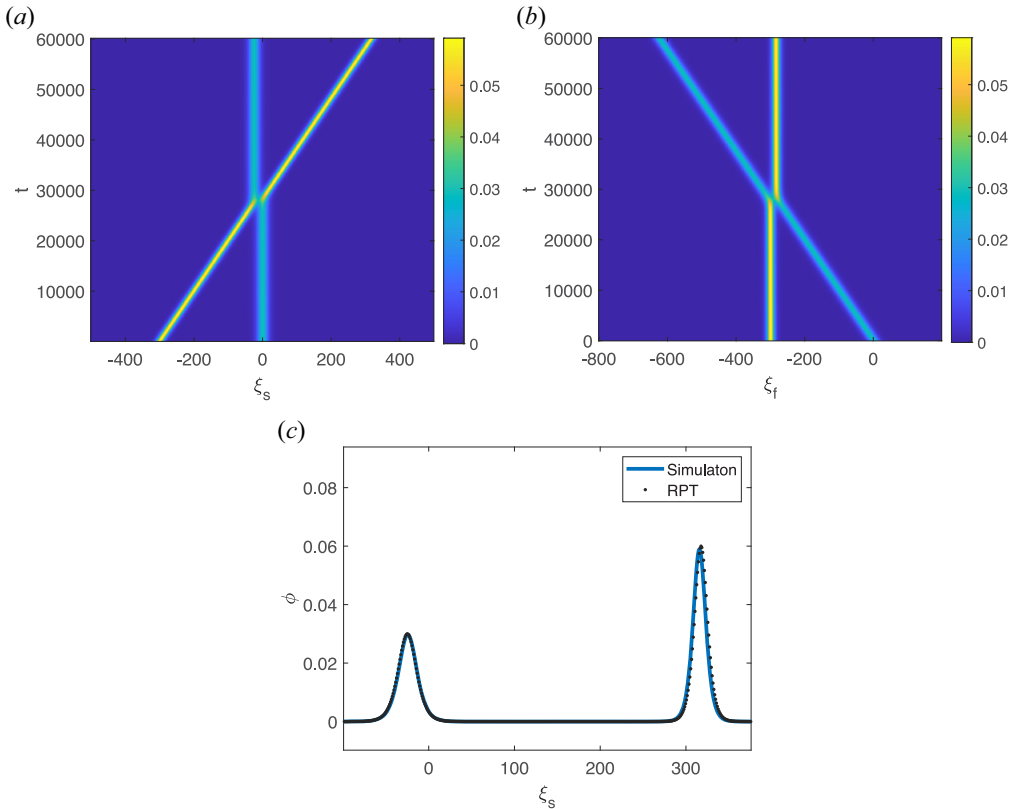
FIGURE 5. Simulation results for a soliton collision with slow and fast soliton speeds of $M_s = 1.01$ and $M_f = 1.02$, respectively. The moving frame references $\xi_s = x - M_s t$ and $\xi_f = x - M_f t$ are used in panels (*a*) and (*b*), respectively. In panel (*c*), the solid blue line shows the simulation results at $t = 60\,000$, while the associated two-soliton solution from RPT (2.14) is shown with the black dots.

that the slow soliton (green line) remains vertical, indicative that the propagation of the slow soliton after the collision is unchanged. In addition, we see that the soliton re-emerges slightly to the left of its original position, indicating a phase shift to the left.

To consider the effect of the collision on the fast soliton, we plot the solution in terms of the fast moving frame $\xi_f = x - M_f t$ in figure 5(*b*). The fact that the fast soliton (yellow curve) remains vertical after the collision shows that the speed of the fast soliton is also unaffected by the collision. In addition, we see that the fast soliton re-emerges to the right of its original position, indicating a phase shift towards the right.

Figure 5(*c*) shows a comparison between the two-soliton solution from RPT (2.14) and the simulation result at the termination time $t = 60\,000$. The solid blue line shows the result obtained from the simulation, while the black dots show the two-soliton solution obtained from RPT. This panel shows good agreement between the simulation result and RPT. This is to be expected, as the solitons fall well within the small-amplitude regime.

This leads to the main question of the paper: how do higher order nonlinear effects affect soliton collisions in terms of elasticity and phase shift? To investigate this question, we next consider a collision of two solitons in the large-amplitude regime.

5.2. *Large-amplitude soliton collision $M_s = 1.2$*

We now consider the simulation results for a collision between two large-amplitude solitons, where the slow soliton propagates with a speed of $M_s = 1.2$ and the fast soliton has a speed of $M_f = 1.4$. The amplitudes of the two solitons are given by $\phi_s \approx 0.52439$ and $\phi_f \approx 0.93827$, respectively. For solitons with such large amplitudes, the effects of higher-order nonlinearities can no longer be ignored, so that one would expect deviations from the small-amplitude regime.

For these simulations, the widths of the solitons are fairly small, so that a significantly smaller interval length of $L = 200$ can be used. However, the large gradients associated with these small widths require us to choose a much finer spatial grid than for the previous example, namely a spatial step size of $\Delta x = 0.01$ and a temporal step size of $\Delta t = 0.001$. Fortunately, the large difference between the speeds means that the time integration can be performed over a relatively short time span. Here, we solved the solution for $0 \leqslant t \leqslant 500$.

The results of the simulation are shown in figure 6. In figure 6(a), the solution is shown in terms of the slow moving frame $\xi_s = x - M_s t$. After the collision at $t \approx 250$, we see that the slow soliton (green line) remains vertical. Remarkably, this shows that the propagation of the slow soliton after the collision is unchanged. In addition, the leftward phase shift is clearly visible. Similarly, figure 6(b) shows that the fast soliton also recovers its original speed after the collision. In this panel, the rightward phase shift of the fast soliton is clearly visible.

To compare the simulation results with the associated result from RPT, we plot the result of the simulation along with the two-soliton solution (2.14) in figure 6(c) at the termination time $t = 500$. The solid blue line shows the result of the solution obtained from the simulation, while the black dashed line show the two-soliton solution of RPT. As mentioned in § 2.4, the difference in shapes is due to higher order nonlinear effects. The most important aspect that is shown here is that there is a significant difference in phase shifts. In particular, the leftward phase shift of the slow soliton is less than that predicted by the two-soliton solution (2.14) from RPT. Similarly, the rightward phase shift of the fast soliton is smaller than that predicted by RPT.

To summarize, the simulation shows that the elastic nature of overtaking collisions is conserved in the large-amplitude regime. The only effect of higher order nonlinearities is a reduction in the magnitude of the phase shifts of both slow and fast solitons. To investigate this further, we now take a closer look at the effect of higher order nonlinearities on phase shifts.

5.3. *Higher order nonlinear effects on phase shift*

The results from the large-amplitude simulation revealed that the only effect of higher order nonlinear effects is to reduce the phase shift of the solitons after the collision. In the following, we make a comparison between phase shifts obtained from simulations with those obtained from the two-soliton solution (2.14) via RPT.

In table 1, we show the two sets of phase shifts corresponding to different speeds. For both the simulation and the two-soliton solution, the phase shifts were determined by comparing the solutions before and after the collision. For the slow solitons, we see good agreement for speeds $M_s \leqslant 1.1$. Beyond this, we see that the differences between the KdV and simulation results grow increasingly fast. For large speeds, we see that the phase shift of the slow soliton is closer to the simulation of the fast soliton (in magnitude) than to the phase shift predicted by RPT. Similarly, the difference of the phase shifts between the simulated and RPT for the fast soliton becomes larger as the speed increases. Note that
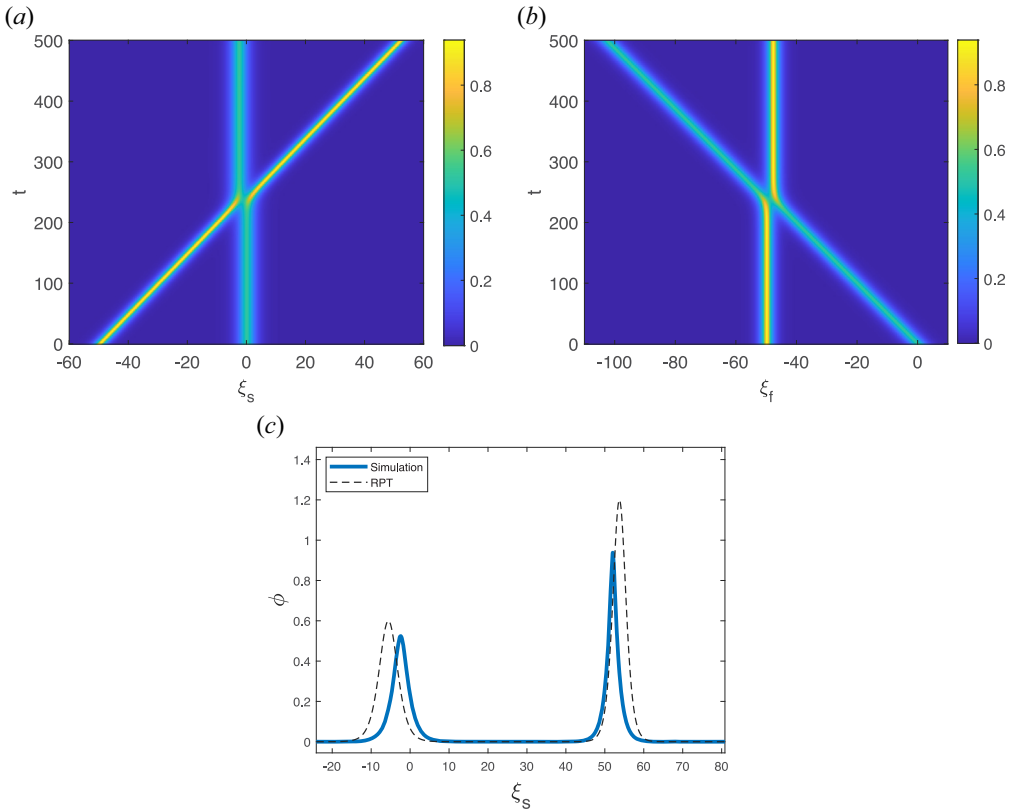
FIGURE 6. Simulation results for a soliton collision with slow and fast soliton speeds of $M_s = 1.2$ and $M_f = 1.4$, respectively. The moving frame references $\xi_s = x - M_s t$ and $\xi_f = x - M_f t$ are used in panels (*a*) and (*b*), respectively. In panel (*c*), the solid blue line shows the simulation results at $t = 500$, while the associated two-soliton solution from RPT is shown with the black dashed line.

increasing speed also leads to an increase in amplitude, thereby amplifying the effects of higher order nonlinearities.

6. Conclusions

In this paper, a new algorithm is designed and implemented to study ion-acoustic solitons for a plasma consisting of cold ions and Boltzmann electrons. The numerical scheme uses a fourth-order Runge–Kutta method to integrate over time, while also using the fourth-order accurate finite difference approximation to approximate all spatial derivatives, thereby resulting in a more accurate scheme than previously implemented. In addition, a novel approach to construct soliton initial conditions directly is derived by means of an asymptotic analysis.

Before proceeding with the simulation of overtaking soliton collisions, we use single-soliton solutions to validate the simulation code. This result shows accuracy up to a five orders of magnitude. Following this, the collisions are simulated. In the small-amplitude regime, collisions are shown to agree well with two-soliton solutions obtained in reductive perturbation analysis.

M_s	RPT slow	RPT fast	Simulation slow	Simulation fast
1.005	−35.25	24.93	−35.09	23.13
1.01	−24.93	17.63	−24.40	15.74
1.05	−11.15	7.884	−9.358	6.349
1.1	−7.884	5.574	−5.429	4.129
1.15	−6.436	4.552	−3.635	2.886
1.2	−5.574	3.942	−2.554	2.113
1.25	−4.986	3.526	−1.822	1.571

TABLE 1. Comparison between phase shifts predicted by RPT and obtained from simulations for different speeds.

For collision of solitons with large amplitudes, collisions are shown to maintain the elastic nature of the small-amplitude regime. This is a remarkable property that shows that solitons are robust against overtaking collisions. An analysis of the phase shift associated with collisions shows that large amplitude phase shifts are smaller (in magnitude) than predicted by RPT in the large-amplitude regime. This seems to be the only effect of higher order nonlinearities on overtaking soliton collisions.

It is important to emphasize that these results are only relevant to this specific fluid model. At present, these results cannot be generalized to more complicated fluid models. Nevertheless, it is the intention that this paper will provide a blueprint for future studies related to soliton collisions, a topic that has received little attention to date. The elastic nature of collisions can also be used as a benchmark for studies of collisions in more complex plasmas.

Acknowledgements

Funding

Declaration of interests

The author reports no conflict of interest.

Appendix A. Poisson's boundary value problem

An interesting challenge of the fluid code is the solution of Poisson's equation. Unlike the continuity and momentum equations, Poisson's equation has no time derivatives, reducing the problem to an ODE boundary value problem. Indeed, the time dependence is captured in the ion number density n, while the nonlinear dependence of the electron density n_e on the electrostatic potential ϕ means that the problem is nonlinear. The value of the electrostatic potential depends on the ion number density, that is to say, we can express the former as $\phi(n)$.

Poisson's equation is given by

$$\frac{\partial^2 \phi}{\partial x^2} = e^\phi - n. \tag{A1}$$

To discretize the problem, we approximate the second-order spatial derivative using the finite difference approximation (3.7). Substitution into Poisson's equation (A1) then gives

$$\frac{-\phi_{i-2} + 16\phi_{i-1} - 30\phi_i + 16\phi_{i+1} - \phi_{i+2}}{12\Delta x^2} = e^{\phi_i} - n_i \tag{A2}$$

for $i = 0, 1, \ldots, N - 1$. Here, $\phi_i = \phi(x_i, t)$ and $n_i = n(x_i, t)$ for some fixed time t. From the periodic boundary conditions, one has the following identities:

$$\phi_i = \phi_{i+N}, \quad \phi_i = \phi_{i-N}. \tag{A3a,b}$$

From these identities, it follows that $\phi_{-2} = \phi_{N-2}$, $\phi_{-1} = \phi_{N-1}$, $\phi_N = \phi_0$ and $\phi_{N+1} = \phi_1$. Substitution of $i = 0, 1, \ldots, N - 1$ into (A2) leads to a system of N nonlinear equations.

The use of iterative methods for linear systems of equations, namely the Jacobi iterative, Gauss–Seidel and successive over-relaxation (SOR) methods are often used to solve the resulting system of equations. In this scheme, we use the standard Newton method for nonlinear systems. To do this, we express the system of nonlinear equations in the form

$$\boldsymbol{f}(\boldsymbol{\phi}) = \mathbf{0}, \tag{A4}$$

where

$$\boldsymbol{\phi} = \left[\phi_0(t), \phi_1(t), \ldots, \phi_{N-1}(t)\right]^{\mathrm{T}}, \tag{A5}$$

and row $i + 1$ of the function $\boldsymbol{f}(\boldsymbol{\phi})$ is given by

$$\frac{-\phi_{i-2} + 16\phi_{i-1} - 30\phi_i + 16\phi_{i+1} - \phi_{i+2}}{12\Delta x^2} - e^{\phi_i} + n_i. \tag{A6}$$

Newton's method is an iterative method, given by

$$\boldsymbol{\phi}_{j+1} = \boldsymbol{\phi}_j - \boldsymbol{J}^{-1}(\boldsymbol{\phi}_j)\boldsymbol{f}(\boldsymbol{\phi}_j), \tag{A7}$$

where the bold subscript in $\boldsymbol{\phi}_j$ represents the jth iteration of the Poisson solver. Here, $\boldsymbol{J}(\boldsymbol{\phi}_j)$ is the nearly pentadiagonal Jacobian matrix, with entries at row a and column b given by

$$J_{ab}(\boldsymbol{\phi}_j) = \begin{cases} -\dfrac{5}{2\Delta x^2} - e^{\phi_{a-1}} & \text{for } a = b, \text{ where } a = 1, \ldots, N \\[2mm] \dfrac{4}{3\Delta x^2} & \begin{array}{c} \text{for } a = b + 1 \text{ where } a = 1, 2, \ldots, N - 1, \\ b = a + 1 \text{ where } b = 1, 2, \ldots, N - 1, \\ \text{and } (a, b) = (1, N), (N, 1) \end{array} \\[4mm] -\dfrac{1}{12\Delta x^2} & \begin{array}{c} \text{for } a = b + 2 \text{ where } a = 1, 2, \ldots, N - 2, \\ b = a + 2 \text{ where } b = 1, 2, \ldots, N - 2, \\ \text{and } (a, b) = (1, N-1), (2, N), (N-1, 1), (N, 2) \end{array} \\[4mm] 0 & \text{otherwise} \end{cases} \tag{A8}$$

The convergence criterion is given by $\|\boldsymbol{\phi}_{j+1} - \boldsymbol{\phi}_j\| < \varepsilon$, where $\varepsilon = 10^{-8}$ was typically used in our simulations. In the derivation of the numerical scheme, we use the notation

$$\boldsymbol{\phi} = \Pi(\boldsymbol{n}) \tag{A9}$$

to denote the application of the Poisson solver defined above, that is, the numerical method to solve the electrostatic potential $\boldsymbol{\phi}$ for a given ion number density vector $\boldsymbol{n}$.

Appendix B. Fourth-order Runge–Kutta time integration

Having established a way to solve Poisson's equation in Appendix A, we can now proceed to evaluate the temporal evolution by means of solving the continuity and momentum equations numerically. To this end, we use a fourth-order Runge–Kutta integrator. In this Appendix, we consider the advancement of one time step. To this end, we introduce a temporal step size $t_j = j\Delta t$, and use $\boldsymbol{n}^{(j)}$, $\boldsymbol{u}^{(j)}$ and $\boldsymbol{\phi}^{(j)}$ to denote the vectors of the number density, fluid velocity and electrostatic potential at $t = t_j$, respectively. These are then used to calculate $\boldsymbol{n}^{(j+1)}$, $\boldsymbol{u}^{(j+1)}$ and $\boldsymbol{\phi}^{(j+1)}$. This process can then merely be repeated until the solution at the desired termination time is calculated.

To start off, we consider the semi-discretized system of ODEs resulting from keeping the time derivative, but using the finite difference approximation for the spatial derivatives for the continuity equation (2.1). By using the fourth-order finite difference formula (3.6) for the spatial derivative, one obtains the general formula

$$\frac{\mathrm{d}n_i^{(j)}}{\mathrm{d}t} = -\frac{1}{12\Delta x} \left(n_{i-2}^{(j)} u_{i-2}^{(j)} - 8n_{i-1}^{(j)} u_{i-1}^{(j)} + 8n_{i+1}^{(j)} u_{i+1}^{(j)} - n_{i+2}^{(j)} u_{i+2}^{(j)} \right). \tag{B1}$$

A treatment of the boundary conditions similarly to that of the Poisson's equation in Appendix A leads to a system of N ODEs. In vector form, this can be expressed as

$$\dot{\boldsymbol{n}}^{(j)} = \boldsymbol{g}_1 \left(\boldsymbol{n}^{(j)}, \boldsymbol{u}^{(j)} \right). \tag{B2}$$

Similarly, the momentum equation can be semi-discretized to give the following set of equations:

$$\frac{\mathrm{d}u_i^{(j)}}{\mathrm{d}t} = -\frac{u_{i+1}^{(j)}}{12\Delta x} \left(u_{i-2}^{(j)} - 8u_{i-1}^{(j)} + 8u_{i+1}^{(j)} - u_{i+2}^{(j)} \right)$$
$$- \frac{1}{12\Delta x} \left(\phi_{i-2}^{(j)} - 8\phi_{i-1}^{(j)} + 8\phi_{i+1}^{(j)} - \phi_{i+2}^{(j)} \right). \tag{B3}$$

In vector form, this can be expressed as

$$\dot{\boldsymbol{u}}^{(j)} = \boldsymbol{g}_2 \left(\boldsymbol{u}^{(j)}, \boldsymbol{\phi}^{(j)} \right). \tag{B4}$$

By combining (B2) and (B4), one can express both continuity and momentum equations as one large system of ODEs, given by

$$\dot{\boldsymbol{w}}^{(j)} = \boldsymbol{g} \left(\boldsymbol{n}^{(j)}, \boldsymbol{u}^{(j)}, \boldsymbol{\phi}^{(j)} \right), \tag{B5}$$

where

$$\boldsymbol{w} = \begin{bmatrix} \boldsymbol{n}^{(j)} \\ \boldsymbol{u}^{(j)} \end{bmatrix}, \quad \boldsymbol{g} \left(\boldsymbol{n}^{(j)}, \boldsymbol{u}^{(j)}, \boldsymbol{\phi}^{(j)} \right) = \begin{bmatrix} \boldsymbol{g}_1 \left(\boldsymbol{n}^{(j)}, \boldsymbol{u}^{(j)} \right) \\ \boldsymbol{g}_2 \left(\boldsymbol{u}^{(j)}, \boldsymbol{\phi}^{(j)} \right) \end{bmatrix}. \tag{B6a,b}$$

The first step of the Runge–Kutta is straightforward, given by

$$\boldsymbol{k}_1 = \Delta t \boldsymbol{g} \left(\boldsymbol{n}^{(j)}, \boldsymbol{u}^{(j)}, \boldsymbol{\phi}^{(j)} \right). \tag{B7}$$

We use the notation $\boldsymbol{k}_1 = [\boldsymbol{n}^{(k_1)}, \boldsymbol{u}^{(k_1)}]^{\mathrm{T}}$ to denote the different components of the $\boldsymbol{k}_1$ vector. Note that we drop the time step j from this notation for all $\boldsymbol{k}_p$ vectors for $p = 1, 2, 3, 4$. Before proceeding, it is important to note that the approximation for the solution of the ion number density $\boldsymbol{n}$ and fluid velocity $\boldsymbol{u}$ was obtained. However, at this stage, no

approximation for the electrostatic potential ϕ was obtained. To do so, we need to solve the Poisson equation, so that

$$\phi^{(k_1)} = \Pi \left(n^{(j)} + \tfrac{1}{2} n^{(k_1)} \right). \tag{B8}$$

After performing this calculation, one can obtain the next approximation as

$$k_2 = \Delta t g \left(n^{(j)} + \tfrac{1}{2} n^{(k_1)}, u^{(j)} + \tfrac{1}{2} u^{(k_1)}, \phi^{(k_1)} \right). \tag{B9}$$

Once more, the approximation of the electrostatic potential must be calculated before proceeding, given by

$$\phi^{(k_2)} = \Pi \left(n^{(j)} + \tfrac{1}{2} n^{(k_2)} \right). \tag{B10}$$

By following the same pattern, one obtains

$$k_3 = \Delta t g \left(n^{(j)} + \tfrac{1}{2} n^{(k_2)}, u^{(j)} + \tfrac{1}{2} u^{(k_2)}, \phi^{(k_2)} \right), \tag{B11}$$

followed by the calculation of

$$\phi^{(k_3)} = \Pi \left(n^{(j)} + n^{(k_3)} \right). \tag{B12}$$

Finally, one obtains the final approximations, given by

$$k_4 = \Delta t g \left(n^{(j)} + n^{(k_3)}, u^{(j)} + u^{(k_3)}, \phi^{(k_3)} \right). \tag{B13}$$

From this, one obtains the following solutions at the next time step $j + 1$:

$$n^{(j+1)} = n^{(j)} + \tfrac{1}{6} \left(n^{(k_1)} + 2n^{(k_2)} + 2n^{(k_3)} + n^{(k_4)} \right) \tag{B14}$$

and

$$u^{(j+1)} = u^{(j)} + \tfrac{1}{6} \left(u^{(k_1)} + 2u^{(k_2)} + 2u^{(k_3)} + u^{(k_4)} \right). \tag{B15}$$

Once $n^{(j+1)}$ is obtained, one can calculate the updated electrostatic potential, given by

$$\phi^{(j+1)} = \Pi \left(n^{(j+1)} \right). \tag{B16}$$

The progression from the solutions at time step t_j to t_{j+1} is schematically represented in figure 7.

Appendix C. Optimizing the code

Of all the steps involved in the numerical scheme, the most time-consuming operation is the Poisson solver and particularly solving equation (A7), given by

$$\phi_{j+1} = \phi_j - J^{-1} \left(\phi_j \right) f \left(\phi_j \right). \tag{C1}$$

To make the code more efficient, it should be noted that inverting a matrix requires more calculations than solving a linear system of equations. As such, we introduce the vector

$$y_j = J^{-1} \left(\phi_j \right) f \left(\phi_j \right), \tag{C2}$$

leading to the linear system of equations

$$J \left(\phi_j \right) y_j = f \left(\phi_j \right). \tag{C3}$$

Once y_j is solved by means of *LU*-factorization, (A7) is merely given by

$$\phi_{j+1} = \phi_j - y_j. \tag{C4}$$

An important feature of the code is the effective use of Matlab's sparse matrix capabilities. In general, to solve (C3) for a system with N variables requires $\mathcal{O}(N^3)$
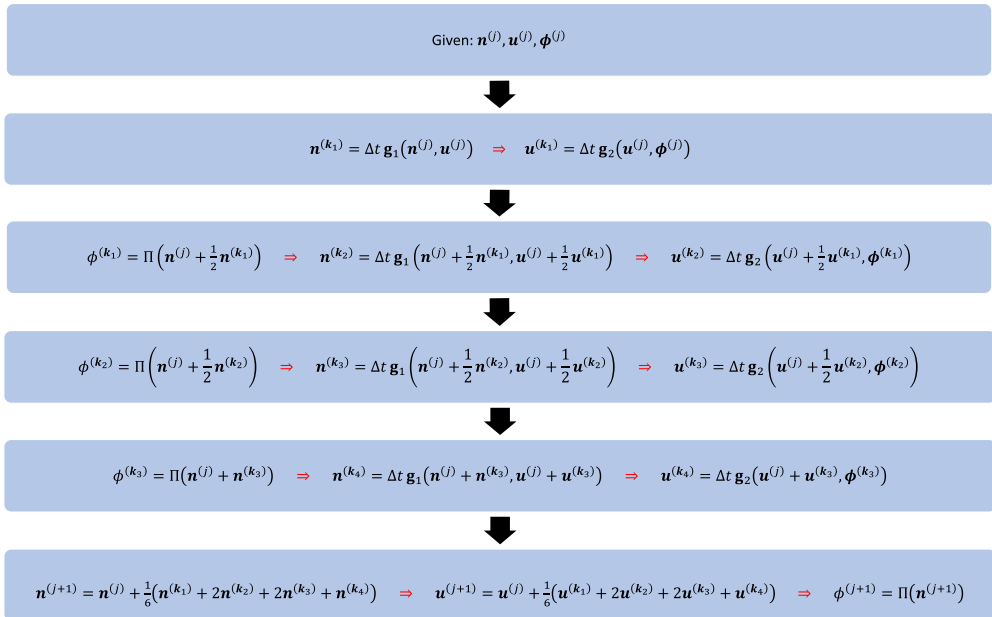
FIGURE 7. One iteration of the numerical scheme to progress from $n^{(j)}$, $u^{(j)}$, $\phi^{(j)}$ to $n^{(j+1)}$, $u^{(j+1)}$ and $\phi^{(j+1)}$.

calculations along with storage space for N^2 entries of the matrix. However, in our numerical scheme, the matrix $J(\phi)$ is a sparse matrix that is nearly pentadiagonal, with the exception of six non-zero entries, three in the north-east corner of the matrix and three in the south-west corner of the matrix.

Entering the Jacobian matrix as a sparse matrix in Matlab significantly reduces the calculation time. Indeed, the *LU*-factorization can be shown to consist of merely $\mathcal{O}(N)$ calculations, while the back-substitutions require only $\mathcal{O}(N)$ calculations. The total number of $\mathcal{O}(N)$ calculations required results in a significant reduction in calculation time, especially for large choices of N.

In a similar way, the sparse matrix requires a mere $5N$ number of storage spaces for the non-zero elements of the matrix. This significantly reduces the memory allocation required to store the elements of the Jacobi matrix. The combination of the reduction in calculation speed and storage requirements leads to vast improvements in calculation time. As a result, the simulations reported in this paper could be performed on a standard laptop. It is worth pointing out that this is in contrast to earlier fluid simulation studies where the use of supercomputers are frequently credited in the acknowledgements.

REFERENCES

BALE, S.D., KELLOGG, P.J., LARSON, D.E., LIN, R.P., GOETZ, K. & LEPPING, R.P. 1998 Bipolar electrostatic structures in the shock transition region: evidence of electron phase space hole. *Geophys. Res. Lett.* **25**, 2929–2932.

BOUNDS, S.R., PFAFF, R.F., KNOWLTON, S.F., MOZER, F.S., TEMERIN, M.A. & KLETZING, C.A. 1998 Solitary potential structures associated with ion and electron beams near $1R_E$ altitude. *J. Geophys. Res.* **104**, 28709–28717.

COURANT, R., FRIEDRICHS, K. & LEWY, H. 1928 Über die partiellen differenzengleichungen der mathematischen. *Physik. Math. Ann.* **100**, 32–74.

GUO, Z.J., ZHOU, H.B., FAN, H.L., LI, M.Q., WU, S.Z., HUANG, T.W., ZHANG, H. & ZHOU, C.T. 2023 Driven ion acoustic wave nonlinearities in superthermal electron plasmas. *Phys. Plasmas* **30**, 022114.

HANSEL, P.J., *et al.* 2021 Mapping MMS observations of solitary waves in Earth's magnetic field. *J. Geophys. Res.* **126**, e2021JA029389.

JENAB, S.M.H. & SPANIER, F. 2016 Study of trapping effect on ion-acoustic solitary waves based on a fully kinetic simulation approach. *Phys. Plasmas* **23**, 102306.

KAKAD, A. & KAKAD, B. 2016 Ponderomotive processes as proxies for breaking of ion acoustic solitary waves. *Phys. Plasmas* **23**, 122101.

KAKAD, A., KAKAD, B., LOTEKAR, A. & LAKHINA, G.S. 2019 Role of ion thermal velocity in the formation and dynamics of electrostatic solitary waves in plasmas. *Phys. Plasmas* **26**, 042112.

KAKAD, A., LOTEKAR, A. & KAKAD, B. 2016 First-ever model simulation of the new subclass of solitons supersolitons in plasma. *Phys. Plasmas* **23**, 110702.

KAKAD, A., OMURA, Y. & KAKAD, B. 2013 Experimental evidence of ion acoustic soliton chain formation and validation of nonlinear fluid theory. *Phys. Plasmas* **20**, 062103.

KAKAD, B., KAKAD, A. & OMURA, Y. 2014 Nonlinear evolution of ion acoustic solitary waves in space plasmas: fluid and particle-in-cell simulations. *J. Geophys. Res.* **119**, 5589–5599.

LAKHINA, G.S., SINGH, S.V., RUBIA, R. & SREERAJ, T. 2018 A review of nonlinear fluid models for ion- and electron-acoustic solitons and double layers: application to weak double layers and electrostatic solitary waves in the solar wind and lunar wake. *Phys. Plasmas* **25**, 080501.

LOTEKAR, A., KAKAD, A. & KAKAD, B. 2016 Fluid simulation of dispersive and nondispersive ion acoustic waves in the presence of superthermal electrons. *Phys. Plasmas* **23**, 102108.

LOTEKAR, A., KAKAD, A. & KAKAD, B. 2017 Generation of ion acoustic solitary waves through wave breaking in superthermal plasmas. *Phys. Plasmas* **24**, 102127.

LOTEKAR, A., KAKAD, A. & KAKAD, B. 2019*a* An effective approach to implement the Maxwellian and non-Maxwellian distributions in the fluid simulation of solitary waves in plasmas. *Commun. Nonlinear Sci. Numer. Simul.* **68**, 125–138.

LOTEKAR, A., KAKAD, A. & KAKAD, B. 2019*b* A fluid simulation-based evidence of the soliton-type behavior of supersolitary waves in plasma. *Phys. Plasmas* **26**, 100701.

MAIN, D.S., NEWMAN, D.L. & ERGUN, R.E. 2006 Double layers and ion phase-space holes in the auroral upward-current region. *Phys. Rev. Lett.* **97**, 185001.

MUSCHIETTI, L., ERGUN, R.E., ROTH, I. & CARLSON, C.W. 1999 Phase-space electron holes along the magnetic field lines. *Geophys. Res. Lett.* **26**, 1093–1096.

OLIVIER, C.P., VERHEEST, F. & HEREMAN, W.A. 2018 Collision properties of overtaking supersolitons with small amplitudes. *Phys. Plasmas* **25**, 032309.

OLIVIER, C.P., VERHEEST, F. & MAHARAJ, S.K. 2017 Asymptotic analysis of solitons and double layers at the acoustic speed. *J. Plasma Phys.* **83**, 905830605.

SAGDEEV, R.Z. 1966 Cooperative phenomena and shock waves in collisionless plasmas. *Rev. Plasma Phys.*, edited by M.A. Leontovich (Consultants Bureau, New York) **4**, 23–91.

SHARMA, S., SENGUPTA, S. & SEN, A. 2015 Particle-in-cell simulation of large amplitude ion-acoustic solitons. *Phys. Plasmas* **22**, 022115.

SINGH, K., KAKAD, A., KAKAD, B. & KOURAKIS, I. 2022 Fluid simulation of dust-acoustic solitary waves in the presence of suprathermal particles. *Astron. Astrophys.* **666**, A37.

SINGH, K., KAKAD, A., KAKAD, B. & SAINI, N.S. 2021 Fluid simulation of ion acoustic solitary waves in electron-positron-ion plasma. *Eur. Phys. J. Plus* **136**, 14.

WASHIMI, H. & TANIUTI, T. 1966 Propagation of ion-acoustic solitary waves of small amplitude. *Phys. Rev. Lett.* **17**, 1966.

YOUNG, D.M. 2014 *Iterative Solutions of Large Linear Systems*. Elsevier.