

Head-on Collisions of Nonlinear Schrödinger waves

KDSMIL001 — June 2021

Abstract

We will study the collisions of two solitons in media with saturation nonlinearity properties, as described by a nonlinear Schrödinger equation. This will be done using two different numerical methods, split step and implicit finite difference/Crank-Nicolson, in order to compare their benefits and drawbacks.

1 Introduction

The equation we will investigate is

$$i\psi_t + \psi_{xx} + \frac{2|\psi|^2\psi}{1 + S \sin |\psi|^2} \quad (1.1)$$

with periodic boundary conditions and the initial conditions being two solitons separated by a sufficiently large distance (~ 40 units) given by

$$\psi(x, 0) = \sum_{j=1,2} A_j \operatorname{sech}[A_j(x - x_j)] e^{iv_j(x - x_j)} \quad (1.2)$$

We will investigate this equation for different values of the saturation of nonlinearity S , velocities v_j , and initial separations $|x_2 - x_1|$. We choose A_j to be the same for all runs.

The two methods we plan to use work in very different ways. The implicit finite difference method, in this case it's specifically a Crank-Nicolson method, is conceptually much simpler. All that we do is approximate the derivative of our function using finite difference methods and solve a matrix equation at each time step to find the value of the function at the next time step. This is conceptually simple, but when it comes to actually doing the computation, because our step in both space in time need to be small in order for our finite difference approximations to be accurate, that means inverting an incredibly large matrix, which is simple but takes a lot of time.

In contrast to that, the split step method is conceptually quite challenging, requiring many complicated steps to arrive at a way to solve for the next time step. This method of solving for the next time step, however, is computationally very simple; all it requires is doing the Fourier Fast Transform and Inverse Fourier Fast Transform and we have our solution. This means that, in theory, the split step method is much less computationally expensive.

2 Method

Our implementation of the split step method is fairly simple, below are the vital lines.

```

1 n=[0:N/2-1 -N/2:-1].^2;
2 n=exp(-1i*tau*4*pi^2/L^2*n);
3
4 psi=psi.*exp((1i*tau*2*abs(psi).^2)...
5             ./(1+S.*sin(abs(psi).^2)));
6 psi=ifft(n.*fft(psi));

```

The first 2 lines are done before we iterate and we do the last 3 for each time step, saving the solution to an array to be plotted at the end. Note the only computationally expensive things being done are the Fourier Fast Transforms, which are done very easily and quickly.

The finite difference implementation is slightly longer but also quite simple.

```

1 A=diag(s/2*ones(N-1,1),1)+diag((1i-s)*ones(N,1),0)...
2   +diag(s/2*ones(N-1,1),-1);
3 A(1,N)=s/2;
4 A(N,1)=s/2;
5 B=inv(A);
6
7 rhs(2:N-1)=-s/2*psi(1:N-2)+(1i+s)*psi(2:N-1)-...
8             (2*tau*abs(psi(2:N-1)).^2)./(1+S*sin...
9             (abs(psi(2:N-1)).^2)).*psi(2:N-1)-s/2*psi(3:N);
10 rhs(1)=-s/2*psi(N)+(1i+s)*psi(1)-(2*tau*abs(psi(1)).^2)...
11        /(1+S*sin(abs(psi(1)).^2)).*psi(1)-s/2*psi(2);
12 rhs(N)=-s/2*psi(N-1)+(1i+s)*psi(N)-(2*tau*abs(psi(N)).^2)
13        ...
14        /(1+S*sin(abs(psi(N)).^2)).*psi(N)-s/2*psi(1);
15 psi = B*rhs;

```

where $s = \tau/h^2$ and from line 6 we iterate through the time steps. The computationally expensive thing here is the multiplication of the matrix B. Normally this isn't a very hard thing to do but since we have on the order of 1000 spatial mesh points, it just takes a long time since matrix is then on the order of 1000×1000 .

The parameters that we could change to optimize each method were as follows: L is the length of the spatial interval, remained at 100 for all tests we made. N is the number of spatial mesh points, which was varied between methods. t_{Max} is the number of time steps taken in a run, which varied depending on velocities and spacing in order for the resulting plots to look nicer. $h = L/N$ is the spatial step and $\tau = h^2/3$ is the time step.

In order to compare the two methods on fair grounds we calibrated each so that the conserved quantity N , given by

$$I = \int_{-\infty}^{+\infty} |\psi|^2 dx \quad (2.1)$$

for one soliton for a small amount of time, stays within some chosen accuracy ε . We chose $\varepsilon = 1 \times 10^{-5}$ and for split step there was no problem staying within that accuracy

as well as keeping the runtime to a reasonable amount. In fact it conserved the quantity to an accuracy of around 1×10^{-12} consistently. For the finite difference method, it was a little bit more complicated. We could just about get the accuracy desired with $N = 2000$. Increasing N does improve the accuracy but it means the runtime goes to ridiculous amounts, which will be discussed later on.

To compare the two methods we simulated a number of collisions varying S , v_j , and $|x_2 - x_1|$ as stated before. The plots of the time evolutions of each variation are in section 3.

3 Results

Let us begin by comparing the two methods on a baseline. Going forward, unless specifically stated, these will be the parameters used: $A_1 = A_2 = 1$, $x_1 = -x_2 = -20$, $v_1 = -v_2 = 5$, $S = 0.5$, $h = L/N$, $\tau = h^2/3$.

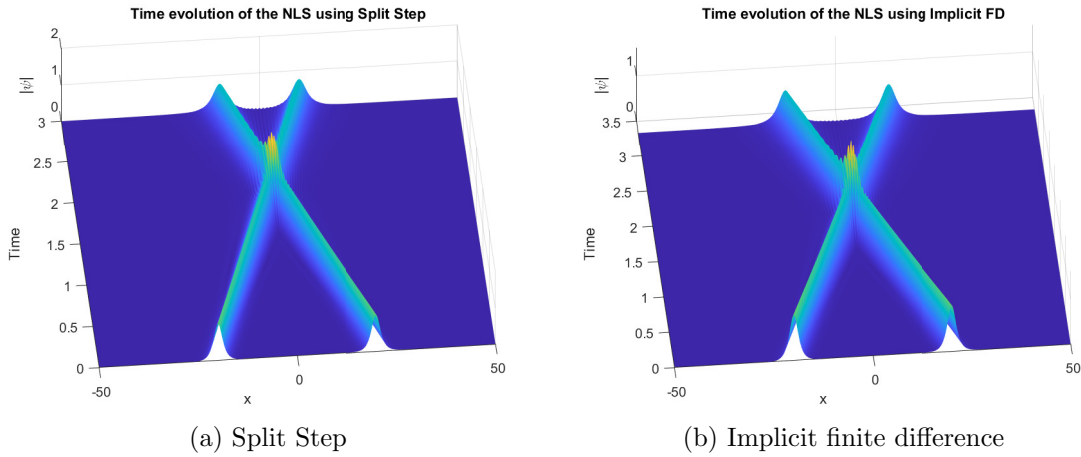
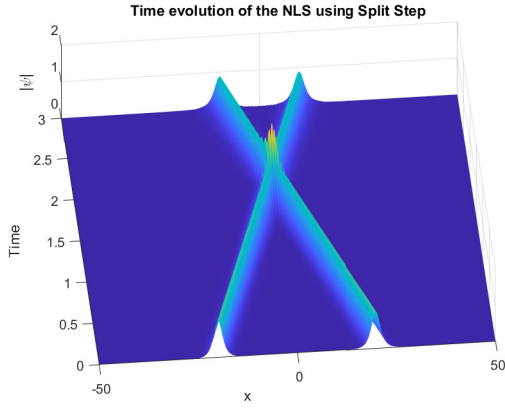
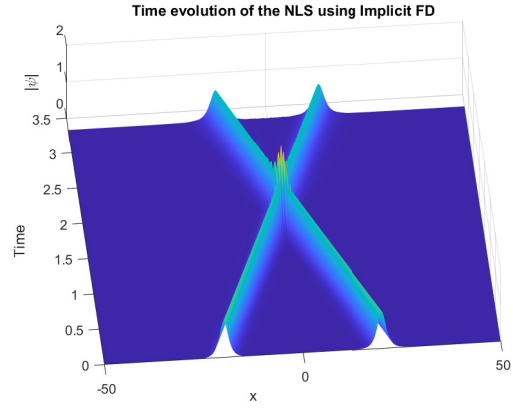


Figure 3.1: Time evolution of the colliding waves using the baseline parameters.

As an aside, there will be discrepancies in the final time on the Time axis for all of these comparisons. This is a result of the differences in τ and h to keep the accuracy in check.

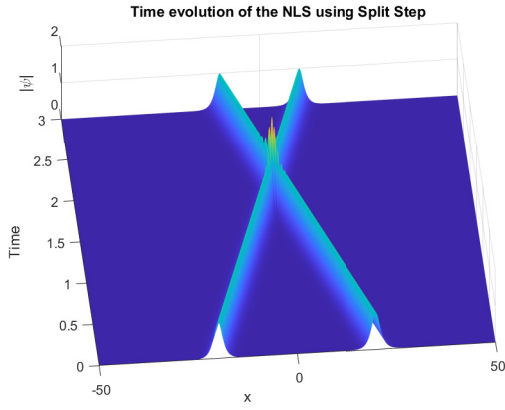


(a) Split Step

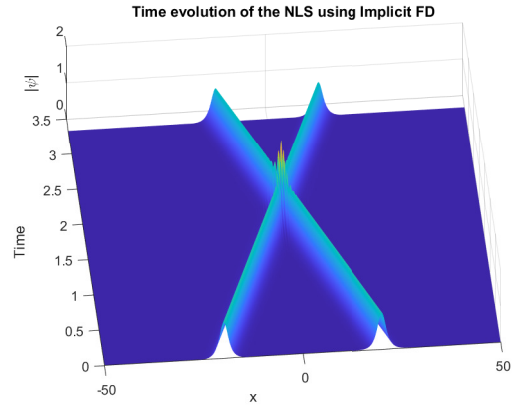


(b) Implicit finite difference

Figure 3.2: Time evolution of the colliding waves with $S = 0.2$.

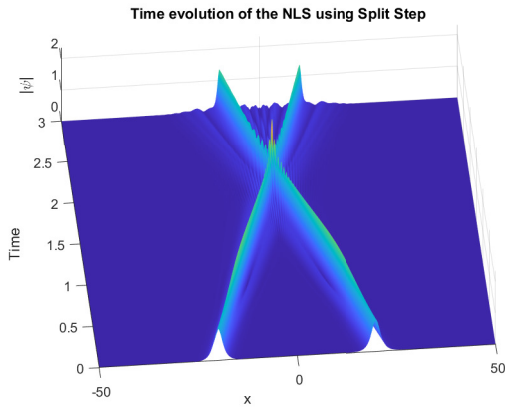


(a) Split Step

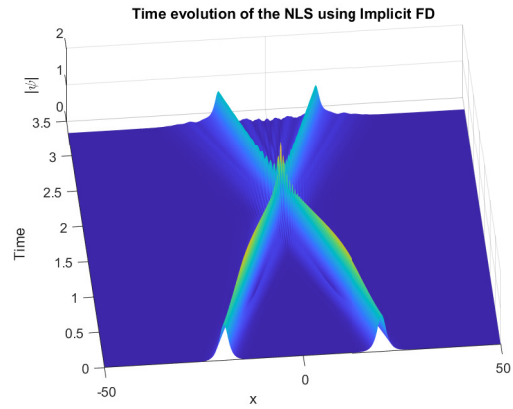


(b) Implicit finite difference

Figure 3.3: Time evolution of the colliding waves with $S = 0$.

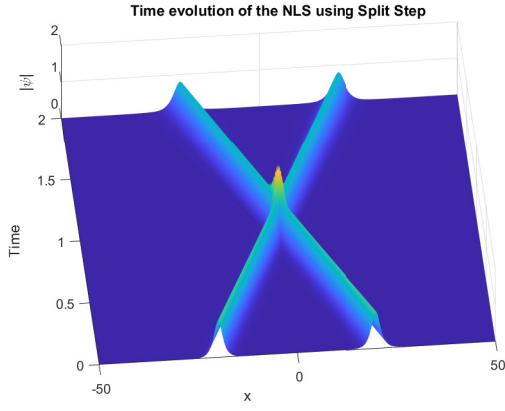


(a) Split Step

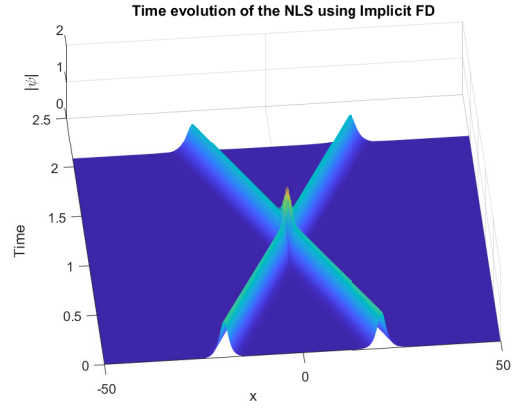


(b) Implicit finite difference

Figure 3.4: Time evolution of the colliding waves with $S = -0.5$.

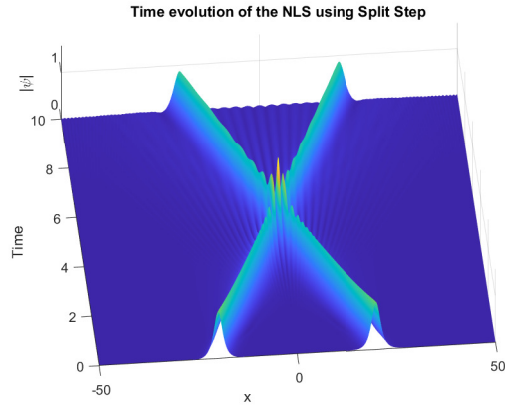


(a) Split Step

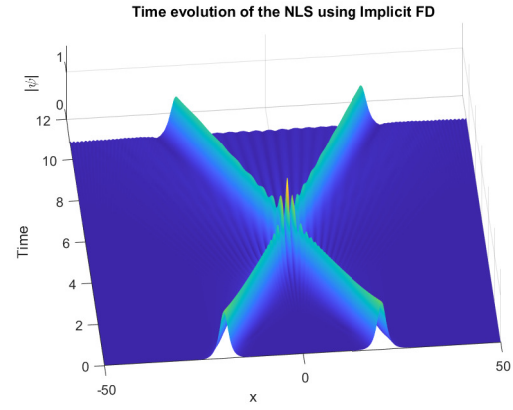


(b) Implicit finite difference

Figure 3.5: Time evolution of the colliding waves with $|v| = 10$.

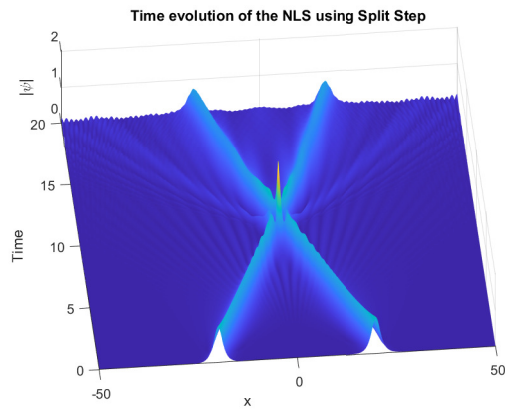


(a) Split Step

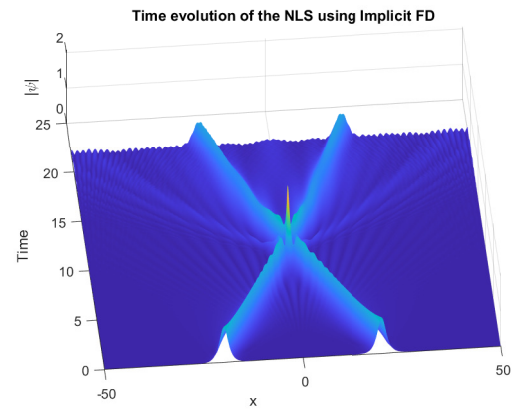


(b) Implicit finite difference

Figure 3.6: Time evolution of the colliding waves with $|v| = 2$.

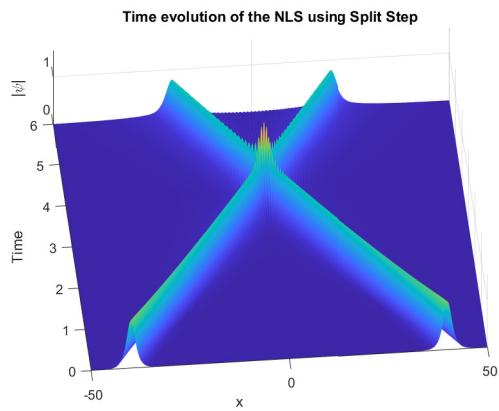


(a) Split Step

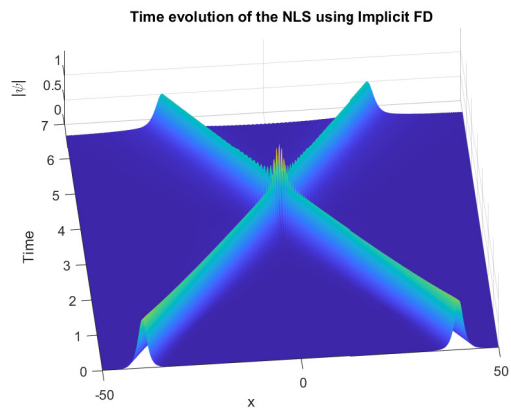


(b) Implicit finite difference

Figure 3.7: Time evolution of the colliding waves with $|v| = 1$.

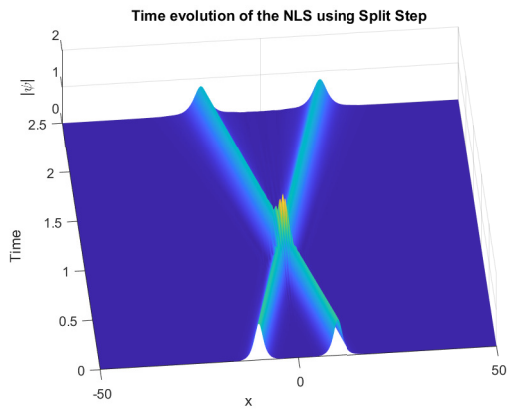


(a) Split Step

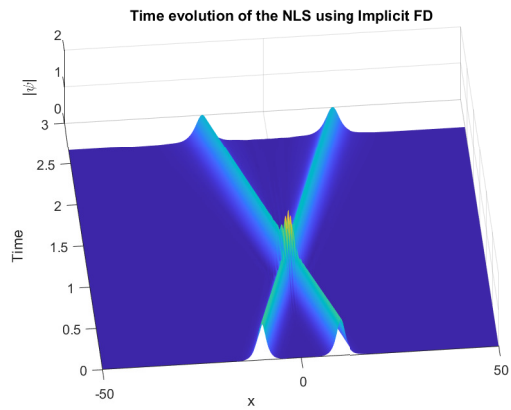


(b) Implicit finite difference

Figure 3.8: Time evolution of the colliding waves with $|x_2 - x_1| = 80$.

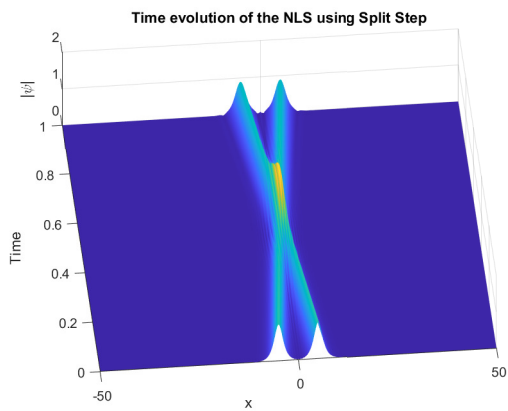


(a) Split Step

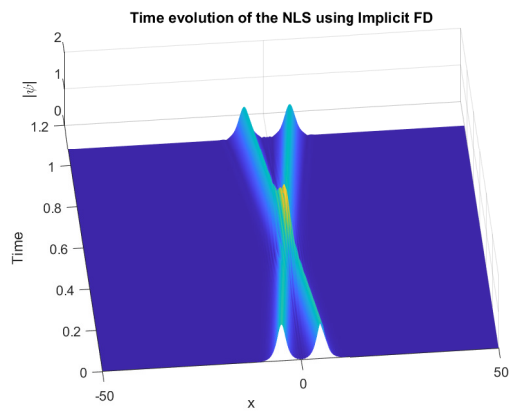


(b) Implicit finite difference

Figure 3.9: Time evolution of the colliding waves with $|x_2 - x_1| = 20$.



(a) Split Step



(b) Implicit finite difference

Figure 3.10: Time evolution of the colliding waves with $|x_2 - x_1| = 10$.

Looking purely at the results, it's clear that the two methods produce similar solutions with similar behaviour, most notably the fact that as the two solitons collide we see oscillations that do not line up with how we expect solitons to interact as they shouldn't interact at all. Nevertheless if we examine the shape of the solution before and after collision we see that for all variations apart from $S = 0$, we see that the solitons devolve into oscillations. This is most noticeable for $S = -0.5$ and $|v| = 1$. From this we can say that if there is some non-zero saturation of nonlinearity the solitons do, in fact, interact and result in strange, undesirable behaviour.

Regarding the differences between values of S , we see that for negative S the solution behaves quite strangely with growing amplitudes and noise around each soliton evolving over time. This is present for positive S but it seems to be less so, and also seems to be less prevalent for values of S closer to 0. Also note that the finite difference method seemed to consistently exaggerate deviations compared to split step.

The differences when it came to $|v|$ are interesting. Thinking about it before we ran the simulations we expected that larger velocities would result in instability since the program is making larger jumps for each time step, but as it turns out we appear to see the opposite happen. For $|v| = 10$ we see a very smooth solution, no oscillations beyond those seen when the solitons combine, but for smaller velocities, namely $|v| = 1$ the solution seems to devolve entirely. This leads us to believe that both methods are stable when it comes to different velocities and this is a result of the saturation of nonlinearity, since these runs were done with $S = 0.5$, and the lower velocities just amplified that noise. Again notice the finite difference method showing larger deviations.

And finally the variations in starting distance for the solitons. This seemed to have little to no effect on the shape and behaviour of the solutions. The most notable thing is that for $|x_2 - x_1| = 80$ we see oscillations start to form between the solitons at the end of the run, but we suspect this is simply because the program had to run for longer in order for the solitons to collide and as time goes on those oscillations grow.

Lastly let us look at the main difference that we noticed between the two methods: The runtime. Using the baseline parameters we found that when doing just the computation, no plotting or anything, the split step method took around 0.3 seconds, while the finite difference method took around 13 seconds. This difference, especially when it comes to looking at long timescales, is quite significant and leads us to greatly prefer the split step method.

4 Conclusion

We found that both the split step and finite difference methods are proficient at solving these nonlinear Schrödinger equations, but the comparison when it comes to computation time as well as general complexity of implementation shows us that the split step method is far superior. Other factors in favour of split step are its ability to conserve area under the integral, I , as well as its resilience against changing of parameters such as h and τ .