A contrast between Split-Step Fourier and Finite-Difference scheme in analyzing the soliton-collision of a coupled Nonlinear Schrodinger Equation found in a context of optical pulses

Themba Malusi

Department of Mathematics and Applied Mathematics

UNIVERSITY OF CAPE TOWN

Abstract

In this report, a coupled Nonlinear Schrodinger Equation (cNLSE) which is found in the context of optical pulses is analyzed using the Split Step and Finite Difference method. The investigation shows interesting dynamics regarding certain values for parameter α as well as a comparison between the two numeric schemes demonstrating the Split Step to be superior for this problem.

Contents

1 Introduction

The interaction of optical pulses in nonlinear media can be studied in a system of the coupled Nonlinear Schrödinger (cNLS) equation [Kat89] with the following nonlinearity:

$$\mathrm{i}\frac{\partial u}{\partial t}+\frac{\partial^2 u}{\partial x^2}+\left(|u|^2-\alpha\cos\left(|v|^2\right)\right)u=0 \tag{1.1}$$

$$\mathrm{i}\frac{\partial v}{\partial t}+\frac{\partial^2 v}{\partial x^2}+\left(|v|^2-\alpha\cos\left(|u|^2\right)\right)v=0 \tag{1.2}$$

where $u(x,t), v(x,t)$ are complex valued amplitudes of optical pulses and α is the coupling parameter and takes the interval $[-5,5]$. This is a system of Partial Differential Equations (PDEs) as they describes the relation of $u(x,t)\,, v(x,t)$ with regards to change over time and space. We observe when the coupling parameter is zero, *i.e.* $\alpha=0$, then the system reduces to a cubic Nonlinear Schrödinger equations and it is completely integrable [AS81] and has the so-called soliton solution defined by [AS81, ZA11, SH00]

$$u(x,t)\ ,v(x,t)=A\mathrm{sech}(A(x-\xi t))\mathrm{e}^{\mathrm{i}[\xi x+(A^2-\xi^2)t]} \tag{1.3}$$

where A is the amplitude of the wave, ξ is the regarded as the velocity of the wave. Henceforth, the aim of this report is to study the head-on collision of two such solitons (1.3), i.e the initial configuration of the form

$$u(x,0),\ v(x,0)=\sum_{k=1,2}\mathrm{sech}(x-x_k)\mathrm{e}^{\mathrm{i}v_k(x-x_k)} \tag{1.4}$$

with varying values for inter-soliton distance $|x_2-x_1|$ and the coupling parameters α by numerically advancing from an initial configuration (1.4) via the Split-Step Fourier scheme [WH86] and Finite Difference methods [DFP81].

2 Split-Step Fourier Method

The Split-Step Fourier scheme is a pseudo-spectral method. This method occurs in several applications, and is a useful numerical method when the equation can be split into two separate equations which are the linear and nonlinear parts, each of which can either be solved exactly, or each part is best solved by a different numerical method. Then, the linear and nonlinear part of coupled system (1.1)-(1.2) are given by

$$\text{Linear Part:}\begin{cases}\mathrm{i}\frac{\partial u}{\partial t}+\frac{\partial^2 u}{\partial x^2} &=0\\ \mathrm{i}\frac{\partial v}{\partial t}+\frac{\partial^2 v}{\partial x^2} &=0\end{cases} \tag{2.1}$$

and

$$\text{Nonlinear Part:}\begin{cases}\mathrm{i}\frac{\partial u}{\partial t}+\left(|u|^2-\alpha\cos\left(|v|^2\right)\right)u &=0\\ \mathrm{i}\frac{\partial v}{\partial t}+\left(|v|^2-\alpha\cos\left(|u|^2\right)\right)u &=0\end{cases} \tag{2.2}$$

Both these equations (2.1)-(2.2) are solved separately and then their solutions are advanced in time by taking small time steps τ for both solutions. For the linear solution (2.1), however needs to be Fourier transformed with the solution being advanced in Fourier space before inverse Fourier transforming back to the time domain.

2.1 Solving the Nonlinear Part

The nonlinear part is rewritten as

$$i\frac{\partial u}{\partial t} + \Big(|u|^2 - \alpha\cos\Big(|v|^2\Big)\Big)\, u = 0 \tag{2.3}$$

$$i\frac{\partial v}{\partial t} + \Big(|v|^2 - \alpha\cos\Big(|u|^2\Big)\Big)\, u = 0 \tag{2.4}$$

we first show that the nonlinearities $|u|^2 - \alpha\cos\Big(|v|^2\Big)$, $|u|^2 - \alpha\cos\Big(|v|^2\Big)$ are constant in time. Then, multiply (2.3) by the complex of conjugate of $\bar{u}$ [1] and multiply the complex conjugation of (2.3) by u and add these equations, then we get

$$\begin{aligned}\bar{u}\frac{\partial u}{\partial t} + u\frac{\partial \bar{u}}{\partial t} &= \mathrm{i}(|u|^2 - \alpha\cos\Big(|v|^2\Big))|u|^2 - \mathrm{i}(|u|^2 - \alpha\cos\Big(|v|^2\Big))|u|^2 \\ &= 0\end{aligned}$$

Thus,

$$\therefore \frac{\partial}{\partial t}u\bar{u} = 0 \implies |u|^2 = \text{constant in time.}$$

Following a similar procedure, one can show that

$$\therefore \frac{\partial}{\partial t}v\bar{v} = 0 \implies |v|^2 = \text{constant in time.}$$

Then the analytical solutions of (2.3)-(2.4) are given by

$$u(x,t) = u(x,0)\mathrm{e}^{\mathrm{i}(|u|^2 - \alpha\cos(|v|^2))\tau} \tag{2.5}$$

$$v(x,t) = v(x,0)\mathrm{e}^{\mathrm{i}(|v|^2 - \alpha\cos(|u|^2))\tau} \tag{2.6}$$

Then the solution advanced by taking small time step τ is written as

$$u(x,t_{n+1}) = u(x,t_n)\mathrm{e}^{i\left(|u(x,t_n)|^2 - \alpha\cos\left(|v(x,t_n)|^2\right)\right)\tau} \tag{2.7}$$

$$v(x,t_{n+1}) = v(x,t_n)\mathrm{e}^{i\left(|v(x,t_n)|^2 - \alpha\cos\left(|u(x,t_n)|^2\right)\right)\tau} \tag{2.8}$$

2.2 Solving the Linear Part

The most efficient way to solve the linear part

$$\mathrm{i}\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} = 0 \tag{2.9}$$

$$\mathrm{i}\frac{\partial v}{\partial t} + \frac{\partial^2 v}{\partial x^2} = 0 \tag{2.10}$$

is to use the Fourier series. The Fourier transform of $u(x,t)$ in space is given by

$$u(x,t) = \sum_{n=-\infty}^{\infty} \hat{u}_n(t)\mathrm{e}^{\mathrm{i}\mu_n x}, \quad \mu_n = \frac{2\pi n}{L} \tag{2.11}$$

[1] $\bar{a}$ is complex conjugate

and the inverse Fourier transform is

$$\hat{u}_n(t) = \frac{1}{L}\int_0^L u(x,t)\mathrm{e}^{-\mathrm{i}\mu_n x}\,\mathrm{d}x \tag{2.12}$$

Then substituting (2.11) into (2.9) and simplifying, we get that

$$\sum_{-\infty}^{\infty}\left[\mathrm{i}\frac{\partial \hat{u}_n}{\partial t} + (\mathrm{i}\mu_n)^2\hat{u}_n\right]\mathrm{e}^{\mathrm{i}\mu x} = 0 \tag{2.13}$$

$$\iff \mathrm{i}\frac{\partial \hat{u}_n}{\partial t} + (i\mu_n)^2\hat{u}_n = 0 \tag{2.14}$$

Therefore, the solution of ((2.14) is thus given by

$$\hat{u}_n(t) = \hat{u}_n(0)\mathrm{e}^{-\mathrm{i}\left(\frac{2\pi n}{L}\right)^2 t} \tag{2.15}$$

The solution of (2.9)-(2.10) advanced by small time steps τ in the Fourier space is

$$\hat{u}_n(t_{n+1}) = \hat{u}_n(t_n)\mathrm{e}^{-i\left(\frac{2\pi n}{L}\right)^2\tau} \tag{2.16}$$

$$\hat{v}_n(t_{n+1}) = \hat{v}_n(t_n)\mathrm{e}^{-i\left(\frac{2\pi n}{L}\right)^2\tau} \tag{2.17}$$

Thus, the Split-Step Fourier algorithm is shown below:

Algorithm 1: Split-Step Fourier Algorithm

```
Data: Define the initial configuration (1.4) at $t = 0$
$\alpha \leftarrow$ coupling constant;
$L \leftarrow$ Length of spatial interval;
$T \leftarrow$ Final time;
$\tau \leftarrow$ time steps;
$Nt \leftarrow T/\tau$;
$k \leftarrow [0 : \frac{N}{2} - 1 \quad -\frac{N}{2} : -1]$;
for $n = 1 : Nt$ do
    %% update time;
    $t = t + \tau$;
    %% solving the nonlinear part (2.7)-(2.8);
    $u \leftarrow u\exp\left(\mathrm{i}(|u|^2 - \alpha\cos\left(|v|^2\right))\tau\right)$;
    $v \leftarrow v\exp\left(\mathrm{i}(|v|^2 - \alpha\cos\left(|u|^2\right))\tau\right)$;
    %% convert into Fourier space and solve the linear part via (2.16)-(2.17);
    $u \leftarrow$ IFFT(exp$\left(-\mathrm{i}(2\pi k/L)^2\right)$ FFT($u$));
    $v \leftarrow$ IFFT(exp$\left(-\mathrm{i}(2\pi k/L)^2\right)$ FFT($v$));
end
```

3 Finite-Difference Method (FDM)

The Finite Difference method works by approximating the derivatives in the expression with finite differences. In our coupled-system (1.1)-(1.2), we have u_t, v_t, u_{xx}, v_{xx} that need to be approximated via finite differences. The way u_t, v_t is approximated determines what type of Finite Difference scheme is used which has various implications with regards to *accuracy*, *stability* and *implementation*.

1. The Forward Difference is an explicit scheme which means that the solution at each point at the latest time level can be expressed through the solutions of the previous time levels. Although this simplifies the implementation the scheme suffers from stability issues. In fact, for the PDEs (1.1)-(1.2) it can be shown that the Forward Difference has an exponential growth in error using the Von-Neumann stability analysis.

2. The Backwards Difference is an implicit scheme which means that a system of equations has to be solved in order to compute the solution at the next time level which makes the implementation non-trivial. However, this method has the superior property that it does not suffer from stability issues.

3. The Central Difference method has the advantage over both the Forward Difference and Backwards Difference in regards to the accuracy as the error is of $O(\tau^2)$ compared to $O(\tau)$ of the other methods. This means that the total error of the coupled-system (1.1)-(1.2), is of $O(\tau^2 + h^2)$ where τ is the time step and h is the space step. This method suffers from the same shortcoming as the Forward Difference method: Stability issues.

For this report, we choose to implement the Backwards-Difference scheme via the Crank-Nicolson scheme for the system (1.1)-(1.2) as it has good accuracy and has superior stability property as compared to the Forward Difference and Centered Difference schemes as they suffer to stability issues, *i.e.* there is a restriction to length of the time step-size.

3.1 Discretization

Choose mesh size $\Delta x := \frac{L}{N}$ with N number of sub-intervals, time step τ and denote grid points and time steps as:

$$x_j = -L/2 + (j-1)h, \quad j = 1, 2, \ldots, N+1, \quad t_n = n\tau, \quad n = 0, 1, 2, \ldots \tag{3.1}$$

Let us denote u_j^n and v_j^n by

$$u_j^n := u(x_j, t_n), \quad v_j^n := v(x_j, t_n) \tag{3.2}$$

Then the Crank Nicolson scheme for the coupled system (1.1)-(1.2) is written:

$$i\frac{u_j^{n+1} - u_j^n}{\tau} + \frac{1}{2}\left[\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{h^2} + \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{h^2}\right] + F_j^n u_j^n = 0 \tag{3.3}$$

$$i\frac{v_j^{n+1} - v_j^n}{\tau} + \frac{1}{2}\left[\frac{v_{j+1}^{n+1} - 2v_j^{n+1} + v_{j-1}^{n+1}}{h^2} + \frac{v_{j+1}^n - 2v_j^n + v_{j-1}^n}{h^2}\right] + G_j^n v_j^n = 0 \tag{3.4}$$

where

$$F_j^n := \left|u_j^n\right|^2 - \cos\left(\left|v_j^n\right|^2\right), \quad G_j^n := \left|v_j^n\right|^2 - \cos\left(\left|u_j^n\right|^2\right) \tag{3.5}$$

The Crank Nicolson scheme displayed in (3.3)-(3.4) has a total truncation error of $O(\tau^2 + h^2)$. The Crank-Nicolson scheme (3.3)-(3.4) is written in a *compact form* as

$$(\mathbf{I} - \beta\mathbf{L})\mathbf{U}^{n+1} = (\mathbf{I} + \beta\mathbf{L})\mathbf{U}^{n} + i\tau\mathbf{F}^{n}\mathbf{U}^{n} \tag{3.6}$$

$$(\mathbf{I} - \beta\mathbf{L})\mathbf{V}^{n+1} = (\mathbf{I} + \beta\mathbf{L})\mathbf{V}^{n} + i\tau\mathbf{G}^{n}\mathbf{V}^{n} \tag{3.7}$$

where $\beta = \frac{i\tau}{2h^2}$, $\mathbf{I}$ denotes an $N+1$ identity matrix and the matrix $\mathbf{L}$ is defined by

$$\mathbf{L} = \begin{pmatrix} \text{-2} & 1 & & & & \color{blue}{1} \\ 1 & \text{-2} & 1 & & & \\ & \ddots & \ddots & \ddots & & \\ & & \ddots & \ddots & \ddots & \\ & & & 1 & \text{-2} & 1 \\ \color{blue}{1} & & & & 1 & \text{-2} \end{pmatrix} \tag{3.8}$$

and where $\mathbf{F}^n = \left(F_1^n, \ldots, F_{N+1}^n\right)^T$ and $\mathbf{G}^n = \left(G_1^n, \ldots, G_{N+1}^n\right)^T$. The vector $\mathbf{U}^n, \mathbf{u}^{n+1}, \mathbf{V}^n, \mathbf{V}^{n+1}$ are row vectors of the unknown variables. The colored entries are the periodic boundary conditions imposed on the matrix. In the next section, we will prove that the Crank-Nicolson scheme in (3.3)-(3.4) are unconditionally stable, i.e. there is/are no restriction to the length of time steps.

3.2 Stability Analysis

As a rule Implicit schemes are unconditionally stable, *i.e.* there is usually no restriction to the length of the time step τ. To derive the stability condition for the Crank-Nicolson scheme, we need to disregard its nonlinear parts and to analyze only the linear parts. The Von-Neumann stability analysis [IK12] is used to make sense of the stability. The linear part of the coupled system (1.1)-(1.2) are decoupled and has similar structure, which means that we need to only analyze the stability of one of the linear parts, then the other will have the same stability property. To derive the stability condition for the above scheme (3.3)-(3.4), we need disregard the nonlinear part of (3.3) and to analyze only the linear part:

$$i\frac{u_j^{n+1} - u_j^n}{\tau} + \frac{1}{2}\left[\frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{h^2} + \frac{u_{j+1}^{n} - 2u_j^{n} + u_{j-1}^{n}}{h^2}\right] = 0 \tag{3.9}$$

Then let $u_j^n = a^n \mathrm{e}^{i\beta hj}$. For stability we required that $|a| \leq 1$. Then substitution of this approximation into (3.9) and factoring out common factors, gives

$$\begin{aligned} i\frac{a^n \mathrm{e}^{i\beta hj}(a-1)}{\tau} &+ \frac{1}{2}\frac{a^{n+1}\mathrm{e}^{i\beta hj}(\mathrm{e}^{-i\beta h} - 2 + \mathrm{e}^{i\beta h})}{h^2} \\ &+ \frac{1}{2}\frac{a^{n}\mathrm{e}^{i\beta hj}(\mathrm{e}^{-i\beta h} - 2 + \mathrm{e}^{i\beta h})}{h^2} = 0 \end{aligned} \tag{3.10}$$

Canceling common factors gives

$$i\frac{a-1}{\tau} - \frac{2a}{h^2} \cdot \sin^2\left(\frac{\beta h}{2}\right) - \frac{2}{h^2} \cdot \sin^2\left(\frac{\beta h}{2}\right) = 0 \tag{3.11}$$

Thus,

$$a = \frac{\mathrm{i} + \frac{2\tau}{h^2}\sin^2\left(\frac{\beta h}{2}\right)}{\mathrm{i} - \frac{2\tau}{h^2}\sin^2\left(\frac{\beta h}{2}\right)} \tag{3.12}$$

Therefore, this ratio is of the form $a = -z/z^*$, where z is the complex number and z^* is its complex conjugate.Then, its magnitude is one. Hence, we have that

$$|a| = 1 \tag{3.13}$$

This shows that the Crank-Nicolson scheme displayed in (3.3)-(3.4) is *unconditionally stable*.

4 Simulation Set-Up

To make the comparison between the methods consistent, we require the following quantity to be conserved

$$N = \int_{-\infty}^{\infty} \left(|u|^2 + |v|^2\right) \mathrm{d}x \tag{4.1}$$

to be chosen accuracy ε for one soliton for short time. Numerically, the power N is the function of time. Thereof, to calibrate the parameters (spatial/temporal steps) for both methods, we define

$$\Delta N = |(N(t) - N(0))/N(0)| \tag{4.2}$$

The number of spatial grid-points were fixed to $N = 1024$ and the simulation time $T = 1$ for both methods. The spatial length was chosen to be $L = 64$. Table 1 shows the convergence rate of the two numerical methods. The results shows each of the methods preserves the conserved quantity (4.1) very well; The accuracy of the Crank-Nicolson method is between 10^{-5} and 10^{-3}; The accuracy of Split-Step scheme is between 10^{-14} and 10^{-13}. . The conserved quantity was computed via the Composite Trapezoidal rule. All implementation was completed using MATLAB. All the source code can be found in the Appendix Section.

Table 1: Convergence analysis of Finite-Difference (3.3)-(3.4) and Split-Step method $((2.1) - (2.2))$ in time. Errors are computed in the maximum norm for (4.2)

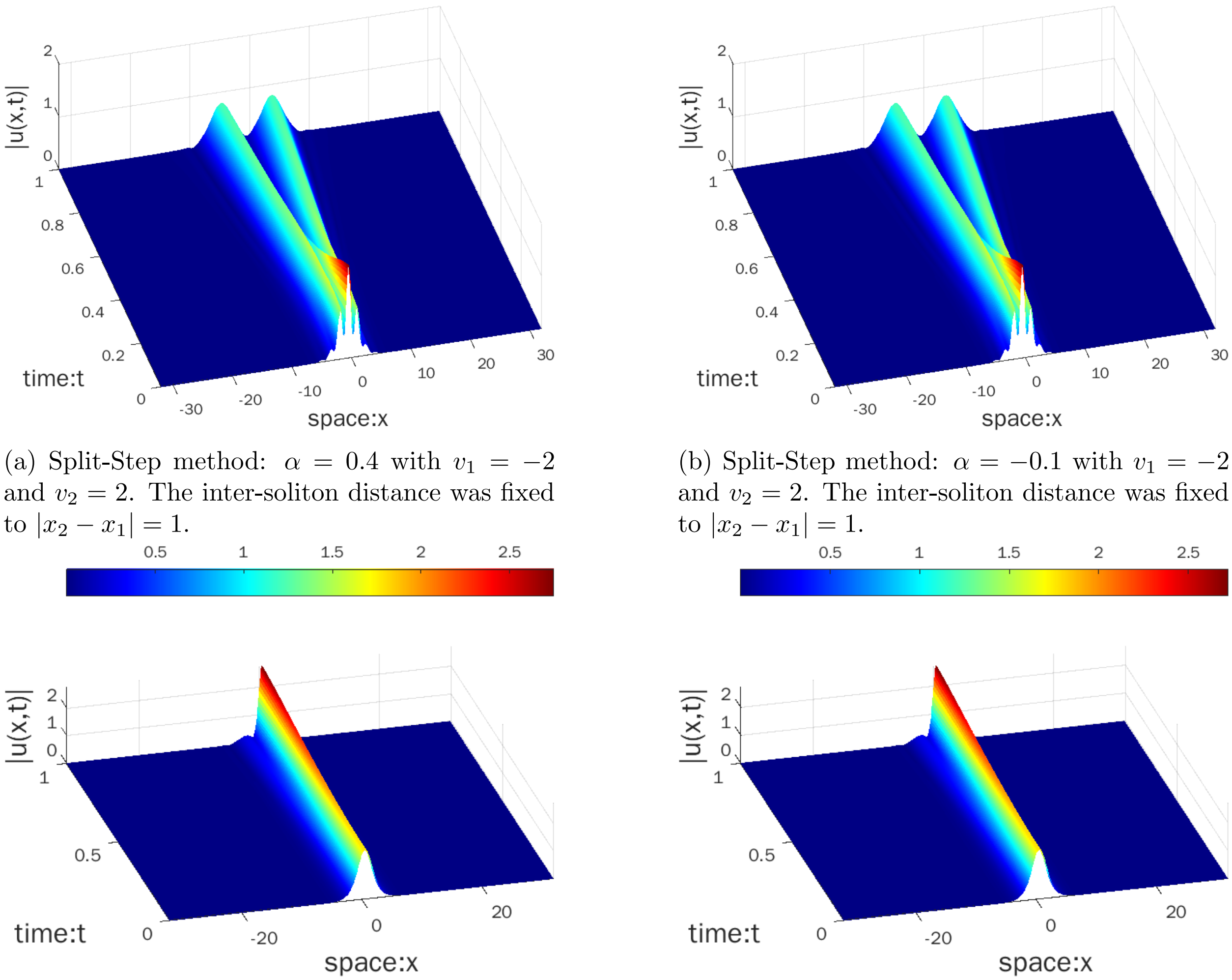
	Finite-Difference method	Split-Step method
$\tau = 0.01$	$4.0596E - 03$	$4.0596E - 03$
$\tau = 0.001$	$4.1168E - 05$	$1.1147E - 14$

In the next section, the results of multiple simulations using varying values for α and velocities v_1 and v_2 are portrayed for both methods.

5 Numerical Results

5.1 Results with a small negative α of -0.1

For these experiments, the coupling constant α was set to a small negative value of -0.1, with inter-soliton distance equal to 1, i.e —$|x_2 - x_1| = 1$. Figure 1a is the evolution of initial configuration (1.4) with velocity of 2 in colliding directions via the Crank-Nicholson method. The same evolution is observed in figure 1b, however using the Split-Step method.



(a) Split-Step method: $\alpha = 0.4$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 1$.

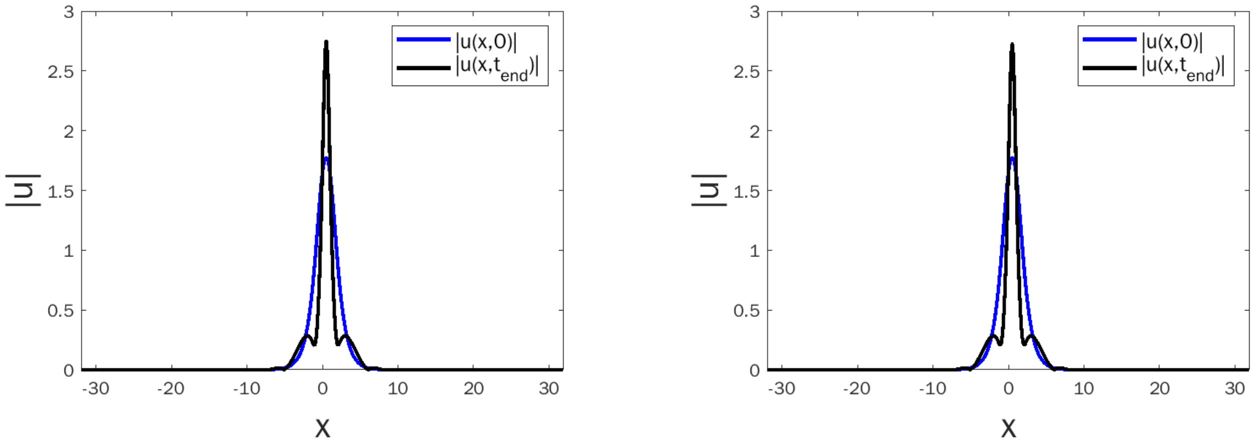
(b) Split-Step method: $\alpha = -0.1$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 1$.

(c) Crank-Nicolson method: $\alpha = -0.1$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 0.2$.

(d) Split-Step method: $\alpha = -0.1$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 0.2$.

The first observation to be made both methods produces colliding solitons that superimpose (red spikes on the figures 1a and 1b) to each other and decompose back to their original state. Figure 1c and 1d shows the evolution of the initial configuration for small values of inter-soliton distance, and it is observed that from the figures (1c and 1d) that initial configuration which is the sum of two solitons is superimposed and has formed one soliton; small inter-solitons distances raised the height of the solitons of the model. See Figures 2and 2b for clear view of this observation

The second observation to be made is that both the Crank-Nicolson and Split-Step schemes produces the same evolution of the initial configuration (1.3)-(1.4) (solitons) of the model (1.1)-(1.2)



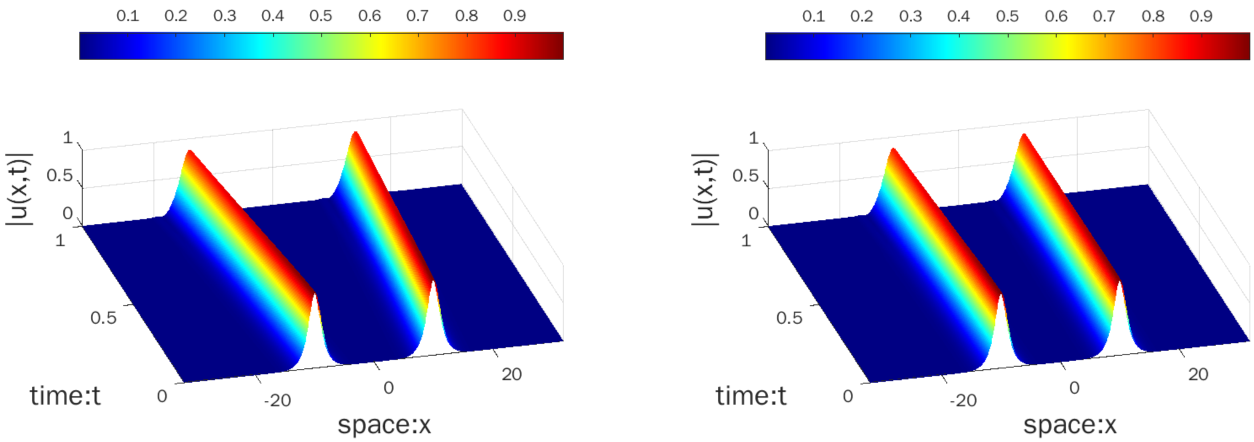
(a) Crank-Nicolson method: $\alpha = -0.1$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 0.2$. *Blue* graph is initial conditions and *black* is solution at $T = 1$, which is final time

(b) Split-Step method: $\alpha = -0.1$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 0.2$. *Blue* graph is initial conditions and *black* is solution at $T = 1$, which is final time.

but the key difference is in their runtime. The Running time for the Crank-Nicolson scheme is three times the running time of the Split-Step scheme. Note that both schemes were ran on the same time steps and same spatial steps. (This will be further illustrated on section 5.5). Hence, it is of this observations that other velocities were ran using the Split-Step method for different values of the coupling constant and for sufficiently fixed large value of inter-soliton distance.

5.2 Results with a small negative α of 0.4

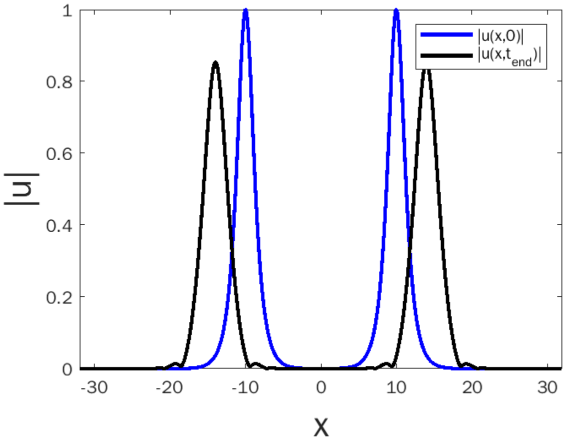
In this section, we ran two numerical simulations using the Split-Step scheme where we fixed the number of spatial steps to $N = 1024$ and spatial length to $L = 64$. The inter-soliton distance was kept constant for both experiments. Figure 3a and 3b shows colliding solitons with large and small magnitude of velocities on opposite directions, respectively.



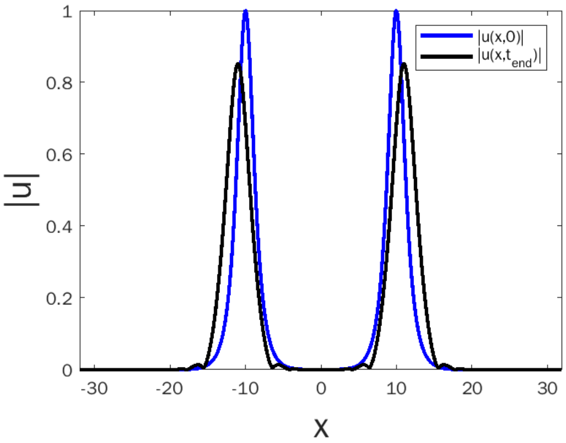
(a) Split-Step method: $\alpha = 0.4$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 20$.

(b) Split-Step method: $\alpha = 0.4$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 20$.

It can be observed from figures, a change in the value of α, has caused the height of the solitons to shrink. The same observation can be made rather for the Crank-Nicolson but the results for this method are not included since they are the same as the Split-Step method. From Figure 4a and 4b, the shrinkage of the height of the solitons is clearly observed.



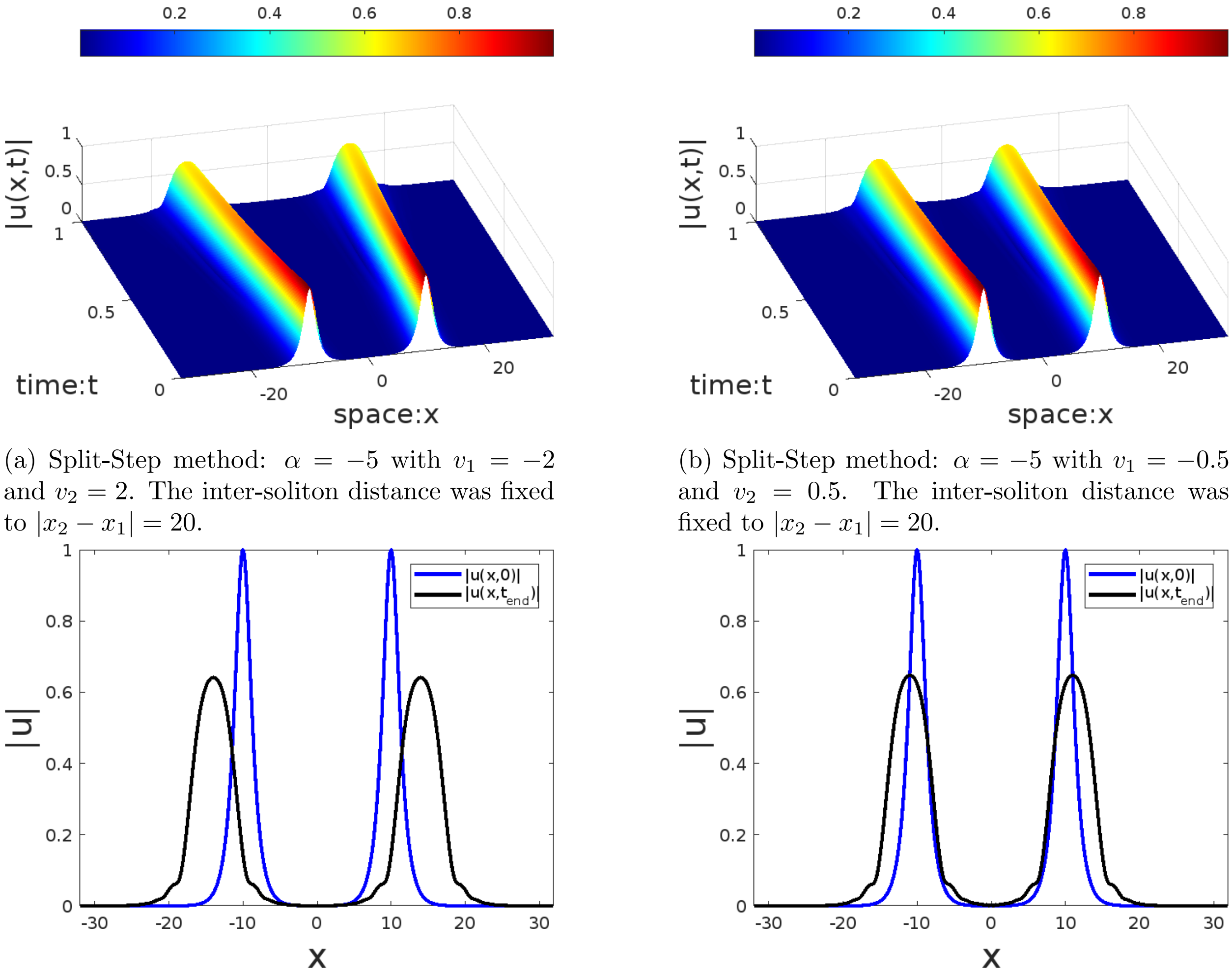
(a) Split-Step method: $\alpha = 0.4$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 20$. *Blue* graph is initial conditions and *black* is solution at $T = 1$, which is final time.



(b) Split-Step method: $\alpha = 0.4$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 20$. *Blue* graph is initial conditions and *black* is solution at $T = 1$, which is final time.

5.3 Results with a small negative α of -5

Previous configurations were kept the same, however simulations were now run using a value for $\alpha = -5$ and with small and large magnitude of velocities for the two solitons. A few observations can be made: The initial configuration has a large soliton height, and as the two solitons evolve in time the soliton height shrinks significantly. The superposition of the solitons can be clearly seen in figure 5a and 5b for both small and large magnitude of velocities. The shrinkage of the solitons shown in figures 5c and 5d.



(a) Split-Step method: $\alpha = -5$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 20$.

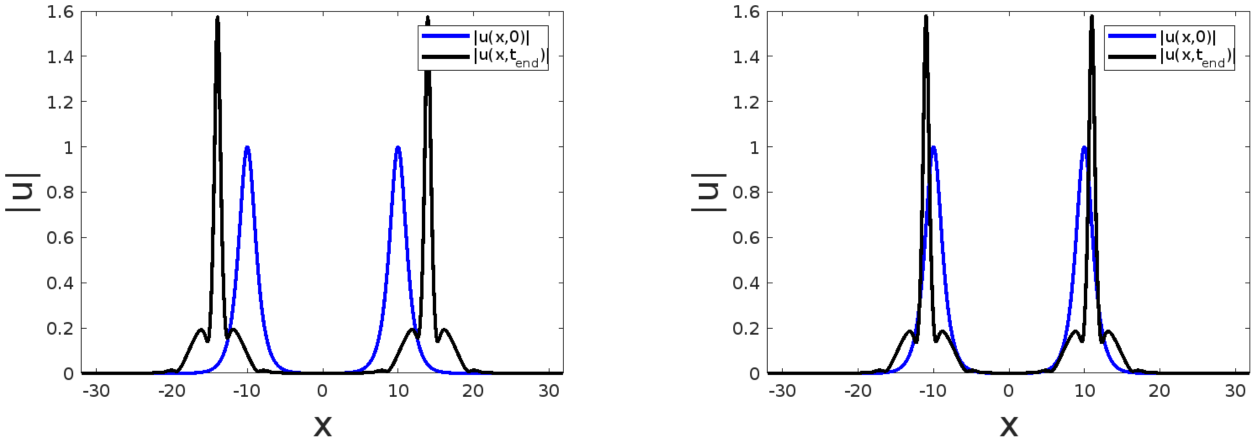
(b) Split-Step method: $\alpha = -5$ with $v_1 = -0.5$ and $v_2 = 0.5$. The inter-soliton distance was fixed to $|x_2 - x_1| = 20$.

(c) Split-Step method: $\alpha = -5$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 20$. *Blue* graph is initial conditions and *black* is solution at $T = 1$, which is final time.

(d) Split-Step method: $\alpha = -5$ with $v_1 = -0.5$ and $v_2 = 0.5$. The inter-soliton distance was fixed to $|x_2 - x_1| = 20$. *Blue* graph is initial conditions and *black* is solution at $T = 1$, which is final time.

5.4 Results with a large positive α of 5

Figure 6a and 6b represent the simulations ran with the value of $\alpha = 5$ using the *Split-Step method*. From these 2D plots, it can be deduced that increasing α resulted in the the solitons increasing in the height.



(a) Split-Step method: $\alpha = -5$ with $v_1 = -2$ and $v_2 = 2$. The inter-soliton distance was fixed to $|x_2 - x_1| = 20$. *Blue* graph is initial conditions and *black* is solution at $T = 1$, which is final time.

(b) Split-Step method: $\alpha = -5$ with $v_1 = -0.5$ and $v_2 = 0.5$. The inter-soliton distance was fixed to $|x_2 - x_1| = 20$. *Blue* graph is initial conditions and *black* is solution at $T = 1$, which is final time.

5.5 Run-Times

To make the comparison between the runtimes of the two schemes: Crank-Nicolson and Split-Step schemes, we fixed the number of spatial steps to $N = 1024$ and the length of spatial interval to $L = 64$. The table 2 shows the runtimes of both schemes for different values of time-steps.

τ	Crank-Nicolson runtime	Split-Step runtime	ratio $= \frac{\text{Crank-Nicolson runtime}}{\text{Split-Step runtime}}$
10^{-1}	0.1000	0.0135	7
10^{-2}	0.2502	0.0162	15
10^{-3}	2.2966	0.1675	14
10^{-4}	19.7777	1.6548	12
10^{-5}	190.4028	15.8815	12

Table 2: Comparison between the running time of the Crank-Nicolson and Split-Step scheme for fixed space $N = 1024$ and $L = 64$.

Column 4 shows the ratio of the runtime of the Crank-Nicolson scheme and the Split-Step scheme; this ratio shows that the runtime of the Split-Step for the same time-steps and fixed spatial steps, is at at least five times faster than the Crank Nicolson scheme. (This is due to the fact that for the Crank-Nicolson scheme, we had to find the inverse of the matrix $\mathbf{L}$, and since periodic conditions are imposed on the matrix, $\mathbf{L}$ which computationally increases the runtime of finding the inverse, $\mathbf{L}^{-1}$; this ultimately increases the runtime of the Crank-Nicolson scheme). Due to these findings, we conclude that the Split-Step scheme is computationally accurate and efficient than the Crank-Nicolson scheme. See figure 7, for further illustration of these findings.

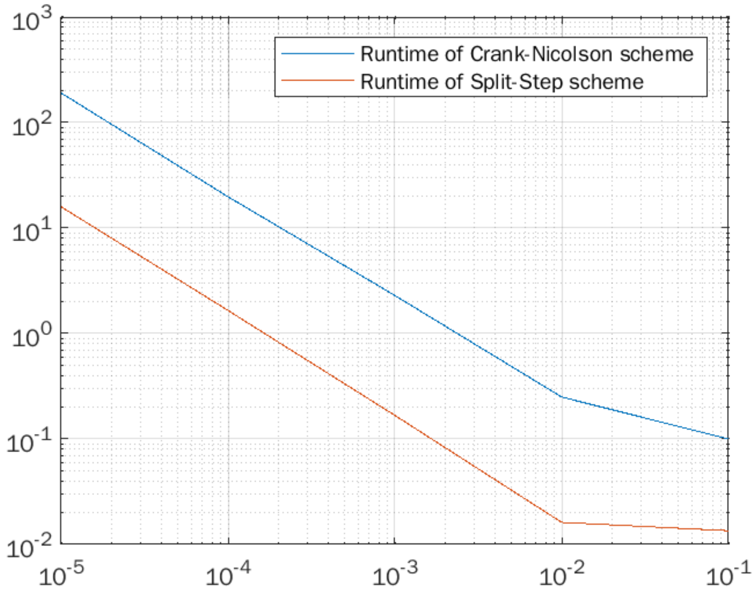


Figure 7: The CPU time of the Crank-Nicolson and Split-Step scheme for fixed spatial steps, $N = 1024$, and fixed spatial length, $L = 64$. This graphs shows the CPU time versus time steps

6 Discussion

In regards to numeric method both produced the same dynamics for either small or large values of α, but the *Split-Step method* converged lot more faster to the solution than the *Crank-Nicolson method*, which took a considerably lot of computational time to get to the solution. Both numeric schemes are *unconditional stable*, *i.e.* there is no restriction to the magnitude of time steps τ.

Both are relative lot easier to implement in MATLAB but there needs to be a careful consideration to the number of spatial steps for the *Crank-Nicolson method*, since further numerical experiments demonstrated that small values of N produces undesired results of the solution.

The Split Step is superior to the Finite Difference in this regard as the spatial interval and time step can be relatively large with no stability issues.

7 Conclusion

In this report the coupled Schrödinger Equation with a particular nonlinearity was investigated using the *Split Step* and *Finite Difference* method. In practice, it was found that the both Split Step method and Crank-Nicolson method does not suffer from stability issues, but the Split-Step method is lot more faster and can solve the equation on a larger spatial interval using a larger time step than the Crank-Nicolson method, which need more spatial steps to achieve the same accuracy and efficiency as the Split-Step method. Thus, it is advisable to solve problems like these using a spectral method like the Split Step over a Finite Difference method. Investigation of the equation itself also showed that the parameter α has a large effect on the dynamics of the solutions with smaller values producing solitons that collide with each other, superimpose and then decompose back into their original states whereas larger absolute values produces non-colliding solitons.

References

[AS81] Mark J Ablowitz and Harvey Segur. *Solitons and the inverse scattering transform.* SIAM, 1981.

[DFP81] Michel Delfour, Michel Fortin, and G Payr. Finite-difference solutions of a non-linear schrödinger equation. *Journal of computational physics*, 44(2):277–288, 1981.

[IK12] Eugene Isaacson and Herbert Bishop Keller. *Analysis of numerical methods.* Courier Corporation, 2012.

[Kat89] Tosio Kato. Nonlinear schrödinger equations. In *Schrödinger operators*, pages 218–263. Springer, 1989.

[SH00] Vladimir N Serkin and Akira Hasegawa. Novel soliton solutions of the nonlinear schrödinger equation model. *Physical Review Letters*, 85(21):4502, 2000.

[WH86] JAC Weideman and Herb. Split-step methods for the solution of the nonlinear schrödinger equation. *SIAM Journal on Numerical Analysis*, 23(3):485–507, 1986.

[ZA11] EV Zemlyanaya and NV Alexeeva. Numerical study of timeperiodic solitons in the damped-driven nls. *International Journal of Numerical Analysis & Modeling, Series B*, 2(2-3):248–261, 2011.

8 Appendix

8.1 Main Script code

This is the main script for running the simulations of both Split-Step and Crank-Nicolson method for coupled system (1.1)-(1.2).

```
%MainScript.m
% simulation the coupled system:
%                   i*u_t+u_xx +(abs(u)^2-alpha*cos(abs(v)^2))*u=0
%                   i*v_t+v_xx +(abs(v)^2-alpha*cos(abs(u)^2))*v=0
% with initial configuaration:
%           u(x,0) = sech(x-x1)*exp(1i*v1*(x-x1))+sech(x-x1)*exp(1i*v1*(x-x1))
%           v(x,0) = sech(x-x1)*exp(1i*v1*(x-x1))+sech(x-x1)*exp(1i*v1*(x-x1))
%for different values of alpha and v1, v2
%%clear all; close all; clc;
%======================== Global parameters ==================================
global alpha x1 x2 v1 v2 off
%======================== PARAMETERS =========================================
N = 1024; L =64; Tmax = 1;  tau = 0.001; alpha = 4;
%========================= Simulation Set-Up =================================
x1 = -10; x2 = 10;       % inter-soliton distance
v1 = -2; v2 = 2;         % velocity of the wave
%%v1 = -0.5; v2 = 0.5;  % velocity of the wave


%the solution via Split Step scheme
[x,tdata,  udata, vdata, RunTime] = Split_Step(tau,N,L,Tmax);

%the solution via Crank-Nicolson scheme
%%[x,tdata1, udata1, vdata1, RunTime1] = Crank_Nicolson(tau,N,L,Tmax);


%% computing the conserved quantity N via composite trapezium method
%%psi1 = abs(udata(:,1)).^2+abs(vdata(:,1)).^2; N0 = trapz(x,psi1); Nerr =[];
%%for nn = 1:16
%%  psi2 = abs(udata(:,nn)).^2+abs(vdata(:,nn)).^2; N1 = trapz(x, psi2);
%%  Nerr(nn) = abs((N1-N0)/N0);
%%end
%%max(Nerr)

%% plots of the results
[X,T]= meshgrid(x,tdata); surface(X,T,abs(udata')); colormap('jet');
colorbar("northoutside"); view(-10,75); zlabel('|u(x,t)|','fontsize',16);
xlim([-L/2 L/2]); xlabel('space:x','fontsize',16);
ylabel('time:t','fontsize',16);


%% the final profile and initial profile
figure(2)
plot(x,abs(udata(:,1)),'-b','linewidth',2,x,abs(udata(:,end)),'-k','linewidth',2)
xlabel('x','fontsize',16); ylabel('|u|','fontsize',16); grid on;
xlim([-L/2 L/2]); legend('|u(x,0)|','|u(x,t_{end})|', 'fontsize',16)
```

8.2 Split-Step method

The function script for the *Split-Step method* 1.

```
%% Solving the Coupled Nonlinear Schrodinger Equation found in the context of
%% optical pulses using the Finite-Difference Method (CN) and Split-Step Method.
function [x,tdata, udata, vdata, RunTime] = Split_Step(tau,N,L,Tmax)
  global alpha x1 x2 v1 v2 off
  tic;
  h  = L/N;               % time step
  x  = [-L/2:h:L/2-h]';   % x-vector
  Nt = round(Tmax/tau);   % number of time step

  %%wave number
  k  = (2*pi/L)*[0:N/2-1 -N/2:-1]';
  k2 = k.*k;
  %% initial configuration
%%  [u,v] = InitOneSoliton(x,off, v1);          %% One Soliton
  [u, v] = Initial_Cond(x,alpha,v1,v2,x1,x2); %% two solitons
  %% computing the exponent
  EXP = exp(-1i*k2*tau);
  %%main calculation
  udata(:,1) = u; vdata(:,1) = v; t =0;tdata(1) = t;
  for tt = 2:Nt
    %update time
    t = t + tau;
    %% solution of the nonlinear part
    F = abs(u).*abs(u)-alpha*cos(abs(v).*abs(v));
    G = abs(v).*abs(v)-alpha*cos(abs(u).*abs(u));
    u = u.*exp(1i*tau*F); v = v.*exp(1i*tau*G);
    %% solution of the linear part
    u = ifft(EXP.*fft(u)); v = ifft(EXP.*fft(v));
    udata(:,tt)= u; vdata(:,tt)= v; tdata(tt)= t;
  end
  RunTime = toc
end
```

8.3 Crank-Nicolson method

The function script for the *Crank-Nicolson method* (3.3)-(3.4).

```
%% Solving the Coupled Nonlinear Schrodinger Equation found in the context of
%% optical pulses using the Finite-Difference Method (CN) and Split-Step Method.

%% Crank-Nicolson scheme for the coupled Nonlinear Schrodinger equation
function [x, tdata, udata,vdata, RunTime] = Crank_Nicolson(tau,N,L,Tmax)
  global alpha x1 x2 v1 v2 off
  tic;
  h  = L/N;             % time step
  x  = [-L/2:h:L/2-h]'; % x-vector
  Nt = round(Tmax/tau);  % number of time step
  s  = 1i*tau/(2*h^2);     % courant coefficient

  %% The nonlinear function
  F = @(uu,vv) abs(uu).*abs(uu)-alpha*cos(abs(vv).*abs(vv));
  G = @(uu,vv) abs(vv).*abs(vv)-alpha*cos(abs(uu).*abs(uu));
  %% initial configuration
%%  [u,v] = InitOneSoliton(x,off, v1);          %% One Soliton
  [u, v] = Initial_Cond(x,alpha,v1,v2,x1,x2); %% two solitons
  %% creating the A, B matrices
  LL = diag(-2*ones(N,1),0) + diag(ones(N-1,1),1) + diag(ones(N-1,1),-1);
  LL(1,end)=1; LL(end, 1)=1; I = eye(size(LL)); A = I-s*LL; B = I+s*LL;
  Ainv = A\eye(size(A));
  %%main calculation
  udata(:,1) = u; vdata(:,1) = v; t =0;tdata(1) = t;
  for n = 2:Nt
    %update time
    t = t + tau;
    %update solution
    F1 = F(u,v); G1 = G(u,v);
    unew = Ainv*(B*u + 1i*tau*F1.*u); vnew = Ainv*(B*v + 1i*tau*G1.*v);
    u = unew; v = vnew;
    %% solution
    udata(:,n) = u; vdata(:,n) = v;tdata(n)=t;
  end
  RunTime = toc
end
```