Frost Summer Research

Computing Nonlinear Waves

By: Connor Leipelt, Kate Davis, and Olivia Hartnett

Introduction

We are focusing on two solutions, called solitons, to the one-component nonlinear Schrodinger equation (NLS):

$i\partial_t u = -\frac{1}{2}\partial_{xx} u + \gamma \lvert u \rvert^2 u,$	(1.1)	NLS equation
$u(x,t) = e^{-i\mu t} v(x),$	(1.2)	Complex Solution form
$-\frac{1}{2}v'' + \left(\gamma \lvert v \rvert^2 - \mu\right) v = 0.$	(1.3)	Time-independent version of NLS

The NLS equation possesses explicit soliton solution for two different choices of gamma:

$\gamma = -1$: $v(x) = A\,\mathrm{sech}\,(Ax)$ where $\mu = -A^2/2$ (this solution is called a "bright" soliton).
$\gamma = +1$: $v(x) = \sqrt{\mu}\,\tanh\,(\sqrt{\mu}x)$ (this solution is called a "dark" soliton).

Solitons are waves that do not change in shape over time (time independent).

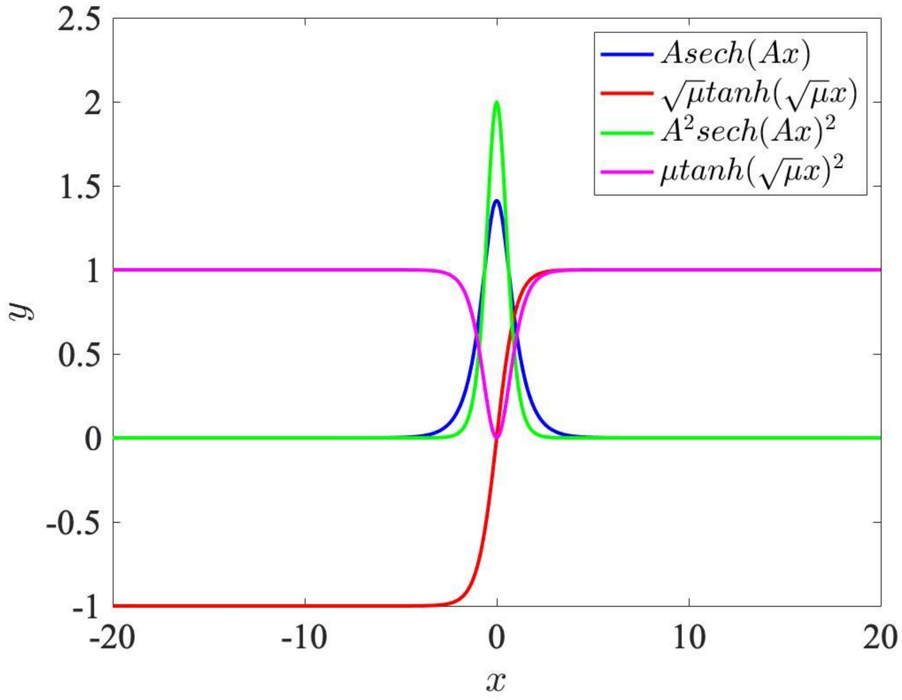
We use partial derivatives to check the solutions of our solitons with Equation 1.1.

Investigation

- We’re exploring different boundary conditions to solve the NLS
- Have focused mainly on using the numerically iterative Newton’s method for the existence problem and using Runge Kutta for the dynamical problem
- Certain boundary conditions cause instability when we work with smaller ranges of boundaries. We study this stability by computing the eigenvalues of the stability matrix.

Graphs of the Bright & Dark Solitons and Their Modulus Squared Functions

- These graphs use
  - A = √(2)
  - Bright γ = -1
  - Dark γ = 1
  - Bright $\boldsymbol{\mu}$ = -1
  - Dark $\boldsymbol{\mu}$ = 1



First Finite Difference Scheme (Newton’s Method for Systems)

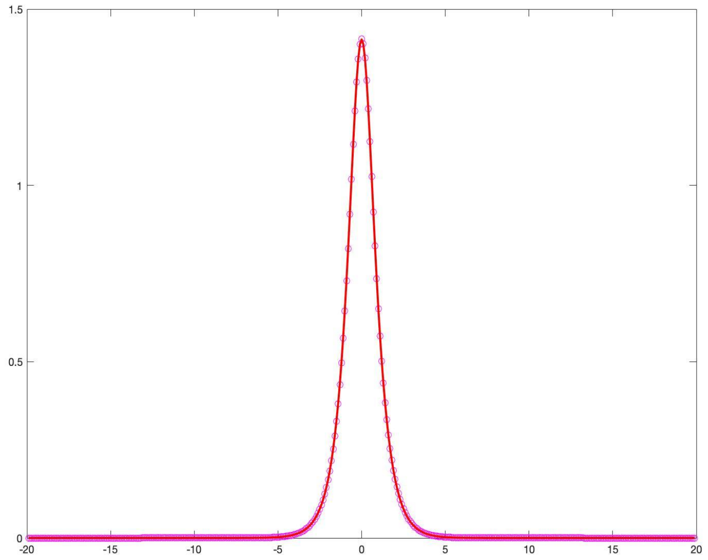
- We used Newton’s method for systems to find solutions to the NLS equation numerically.

- Using a symmetric spatial domain we introduced a finite number of grid points in the interval, then we chose a particular boundary condition and used Newton’s method to find our interior points that solve the NLS equation.

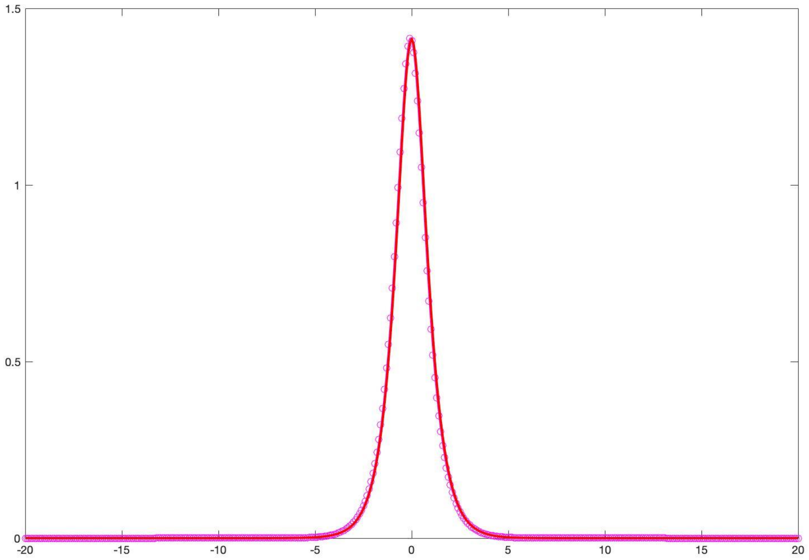
- We used two boundary conditions, Dirichlet and Neumann boundary conditions, with these boundary conditions we can explore stability.

Bright Soliton Dirichlet and Neumann B.C.’s

Dirichlet BC:

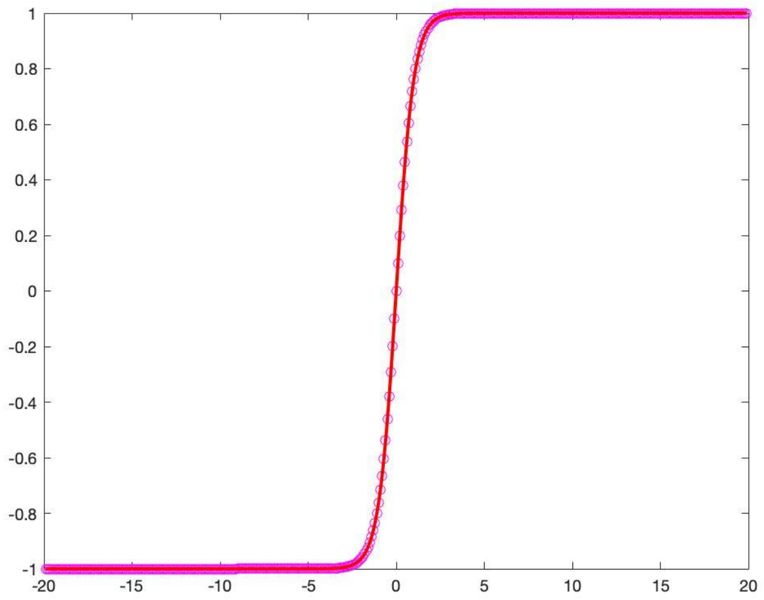


Neumann BC:

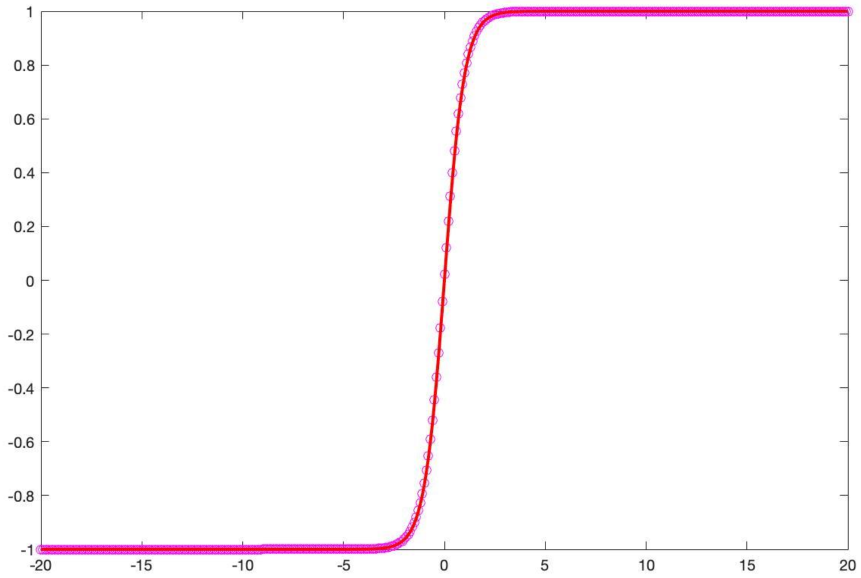


Dark Soliton Dirichlet and Neumann B.C.'s

Dirichlet BC:



Neumann BC:



Stability Analysis

- With this piece of the project we want to study the stability of the stationary solutions which are the bright and dark solitons.
- Using the linearization ansatz of the form

$$u(x,t) = e^{-i\mu t}\left[v(x) + \varepsilon\left(a(x)e^{\lambda t} + b^*(x)e^{\lambda^* t}\right)\right]. \tag{2.1}$$

and if we look into $\mathcal{O}(\varepsilon)$ we get an eigenvalue problem in this form

$$\tilde{\lambda}\begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} \\ -A_{12}^* & -A_{11} \end{pmatrix}\begin{pmatrix} a \\ b \end{pmatrix}, \tag{2.2}$$

- Where

$$A_{11} = -\frac{1}{2}\partial_{xx} + 2\gamma|v|^2 - \mu, \tag{2.3}$$

$$A_{12} = \gamma v^2. \tag{2.4}$$

Stability Analysis (continued)

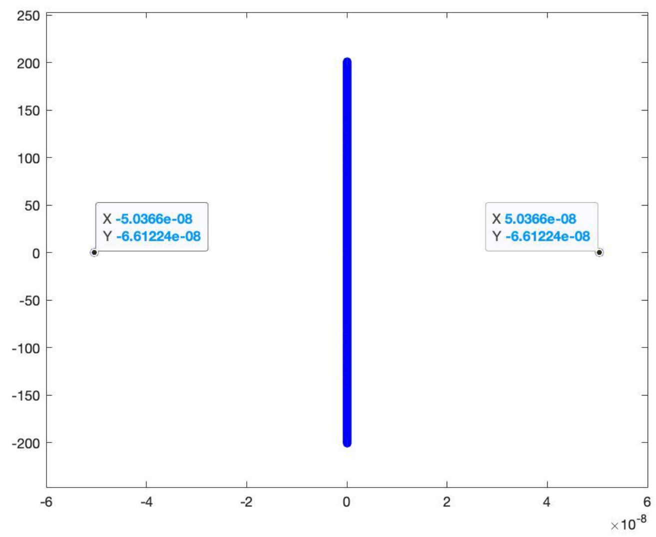
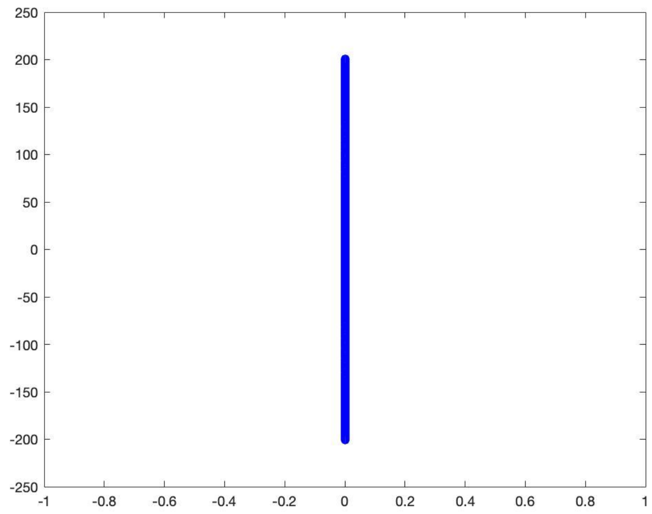
- From this matrix, we find its eigenvalues -- these eigenvalues may be complex

- If the eigenvalues are completely imaginary the solitons are stable. If the eigenvalues have real components that are significantly large, anything larger than 1x10^(-6), the solitons are unstable and will exhibit exponential growth breaking the system and exponential decay if the real component is less than -1x10^(-6).

- Our findings showed that both soliton solutions are stable.

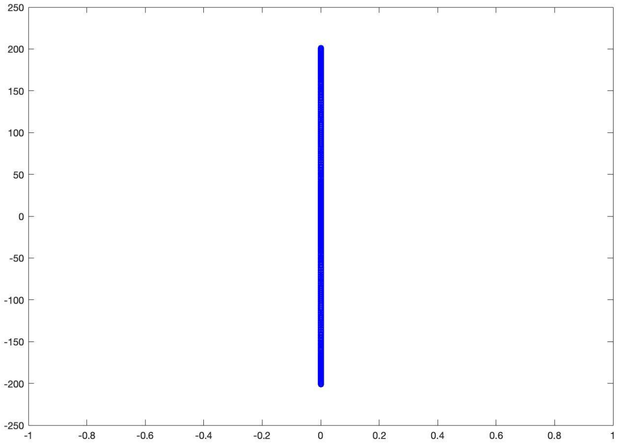
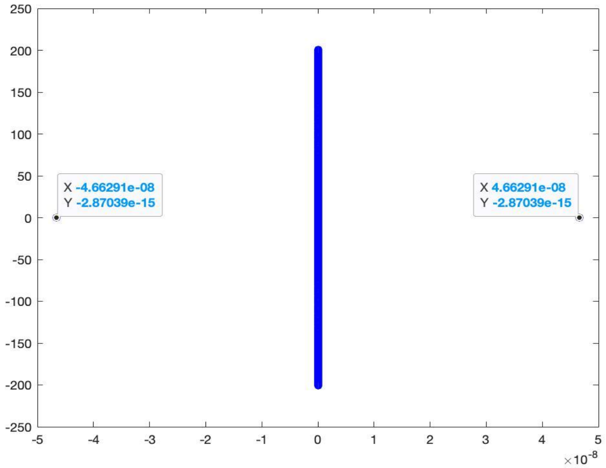
Stability Analysis Graph Findings

Since there are two solutions and two different possible boundary conditions, there are four total eigenvalue graphs. These are the graphs for the Dirichlet boundary condition.

	Bright Soliton	Dark Soliton
Dirichlet BC		

Neumann BC

These are the graphs for the Neumann boundary condition.

	Bright Soliton	Dark Soliton
Neumann BC		

Modulus-Squared Dirichlet Boundary Condition Implementation

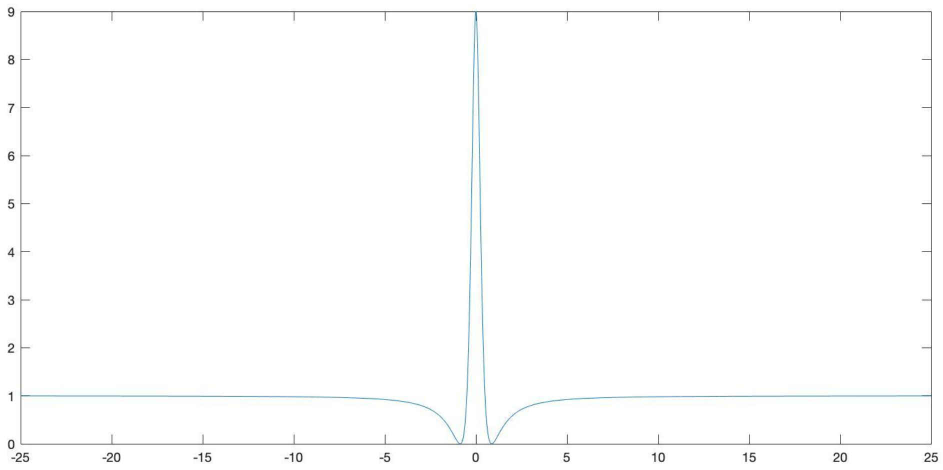
Now we have shifted our focus to using the Modulus-Squared Dirichlet (MSD) boundary condition which is used for time-dependent and complex valued partial differential equations. It helps us find a modulus squared value of the solution at the boundaries.

- reduces complexity, less expensive for many problems.
    - smaller domain → less grid points → less computations
- we can use this for the time-dependent version of the nonlinear schrodinger equation which has real and imaginary parts.

The MSD boundary condition is where the modulus square of the function evaluated at the boundary is a constant.

Peregrine Soliton solution for the NLS equation

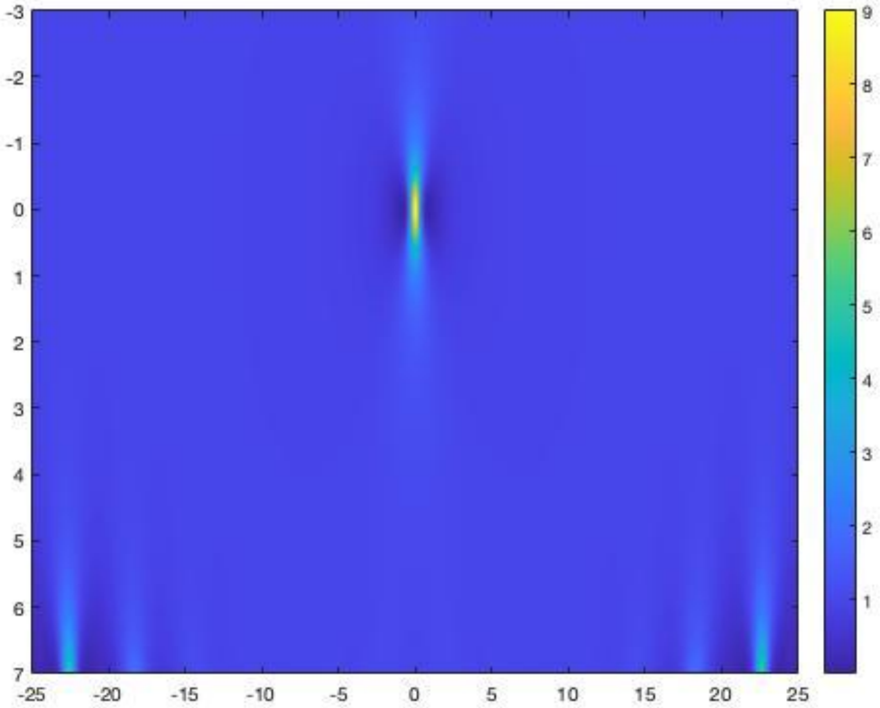
- The MSD boundary condition is used for the time-dependent version of the NLS equation and so we have decided to study the Peregrine soliton which is a valid solution to the time-dependent version of this problem.
- The MSD boundary condition is well suited for the Runge-Kutta numerical solver so we used it in this version.
- Graph of the Peregrine Soliton



Our goal is to compare the numerical versions of the Peregrine soliton when using Neumann BC vs MSD BC

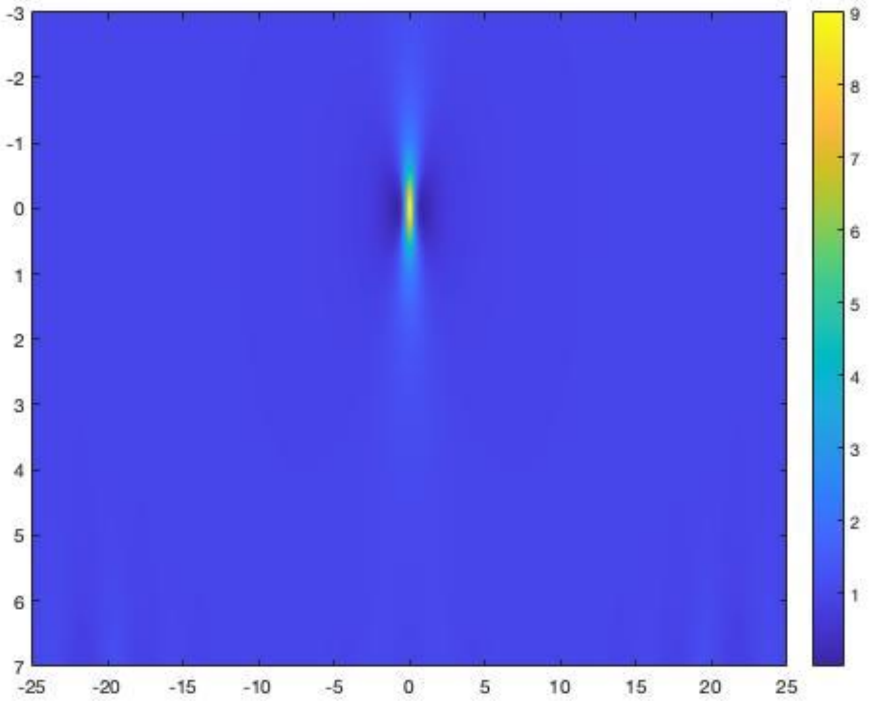
Peregrine Soliton with Neumann boundary condition

The graph vertically is going from -3 to 7 representing our tspan and has stability issues as the tspan reaches 6.



Peregrine Soliton with MSD boundary condition

Unlike the Neumann boundary condition, there are no areas of instability as we approach 6 on our tspan axis. This shows that the MSD boundary condition is a better fit than the Neumann boundary condition as it works for a larger tspan.



Implementing the MSD boundary condition to the Steady-State NLSE

- In our steady-state problem we used Newton's Method for systems to iteratively solve the dark soliton solution.

$$\nabla^2 \Psi_b \approx \left[\mathrm{Im}\left(i \, \frac{\nabla^2 \Psi_{b-1}}{\Psi_{b-1}} \right) + \frac{1}{a} \left(N_{b-1} - N_b \right) \right] \Psi_b \tag{3.2}$$

$$N_b = s \left| \Psi_b \right|^2 - V_b, \qquad N_{b-1} = s \left| \Psi_{b-1} \right|^2 - V_{b-1}. \tag{3.3}$$

- This means that we had to compute the second derivative with respect to our spatial variable at the boundaries and compute the partial derivatives for each boundary for the Jacobian.

$$\begin{aligned} \nabla^2 \Psi_b^R &\approx \left[A + \frac{1}{a} \left(N_{b-1} - N_b \right) \right] \Psi_b^R \\ \nabla^2 \Psi_b^I &\approx \left[A + \frac{1}{a} \left(N_{b-1} - N_b \right) \right] \Psi_b^I, \end{aligned} \tag{3.4}$$

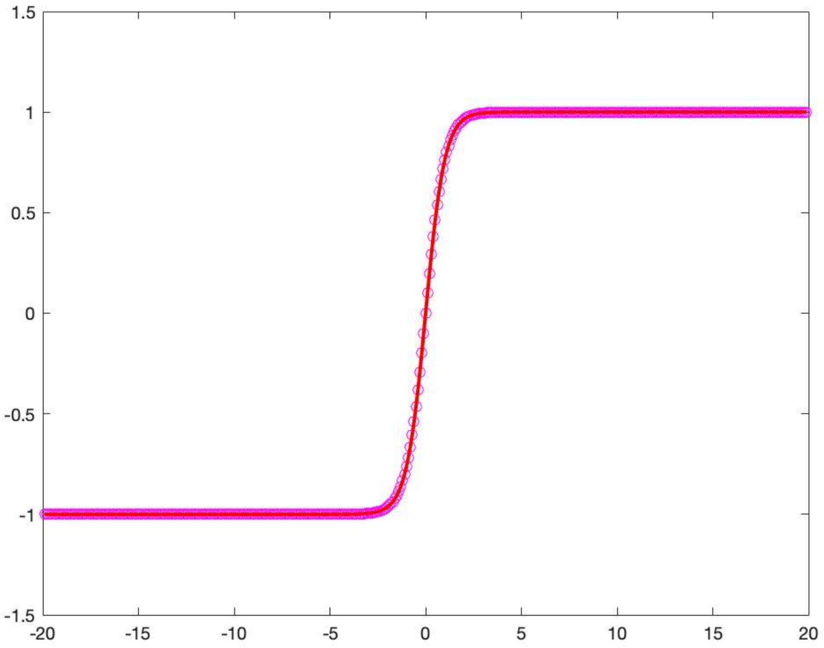
- We used equation 3.4 to compute the real and imaginary parts of the second derivative for each boundary.

$$A = \frac{\nabla^2 \Psi_{b-1}^R \, \Psi_{b-1}^R + \nabla^2 \Psi_{b-1}^I \, \Psi_{b-1}^I}{\left(\Psi_{b-1}^R \right)^2 + \left(\Psi_{b-1}^I \right)^2} \tag{3.5}$$

Applying the MSD boundary condition to the steady state NLS problem

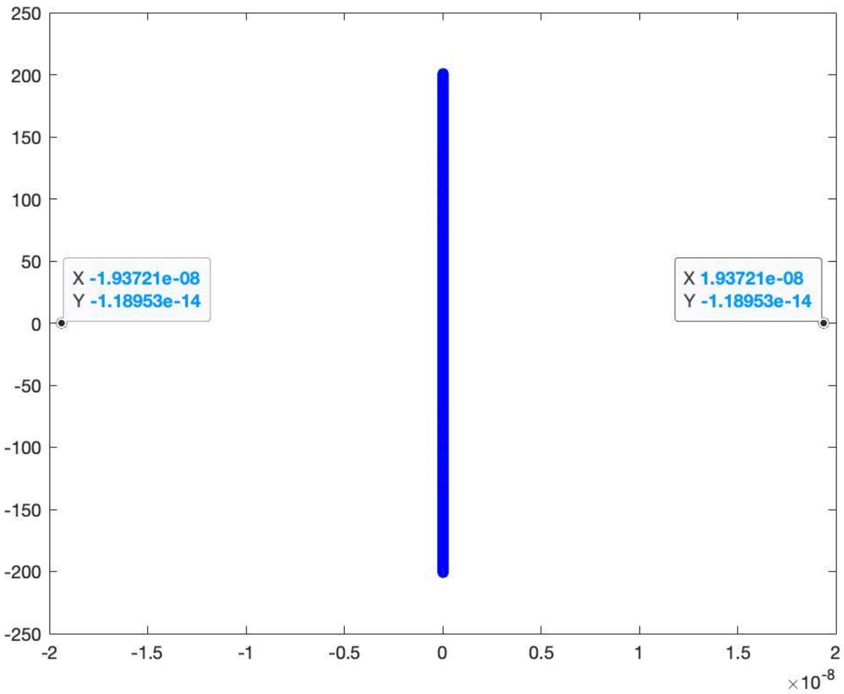
- We now want to apply the MSD boundary condition to our steady-state NLSE.
- We look into the dark soliton solution and see that the MSD boundary condition works well!

Dark soliton using MSD boundary condition



Stability Analysis for the Steady-State NLSE using MSD BC's

- As we tested for Dirichlet and Neumann boundary conditions, we want to see the stability of the steady-state NLSE while using the MSD boundary condition.
- On the right is a graph of our eigenvalues and we see that the NLSE with MSD is stable.



Challenges So Far

- Coding a complex-valued PDE using finite differences
- Constructing the associated Jacobian matrix that we'll use in Newton's method
- Understanding how Newton's method for systems iterates
- Understanding how the stability analysis works with the NLSE and how the eigenvalues can have complex parts
- Working with a time-dependent version of the NLSE in matlab
- Implementing the MSD boundary condition and understanding how and why the MSD boundary condition is a better fit for the time-dependent version of the NLSE

What We’ve Learned This Summer

- The history of solitons
- Understanding how different boundary conditions affect the stability and precision of numerically approximating soliton solutions to the non-linear Schrödinger equation
- Better understanding of Matlab
- How certain boundary conditions work better than others when shrinking the boundary that we are working with.
- Newton’s Method for Systems of Nonlinear Equations
- Stability Analysis of solutions to the NLSE