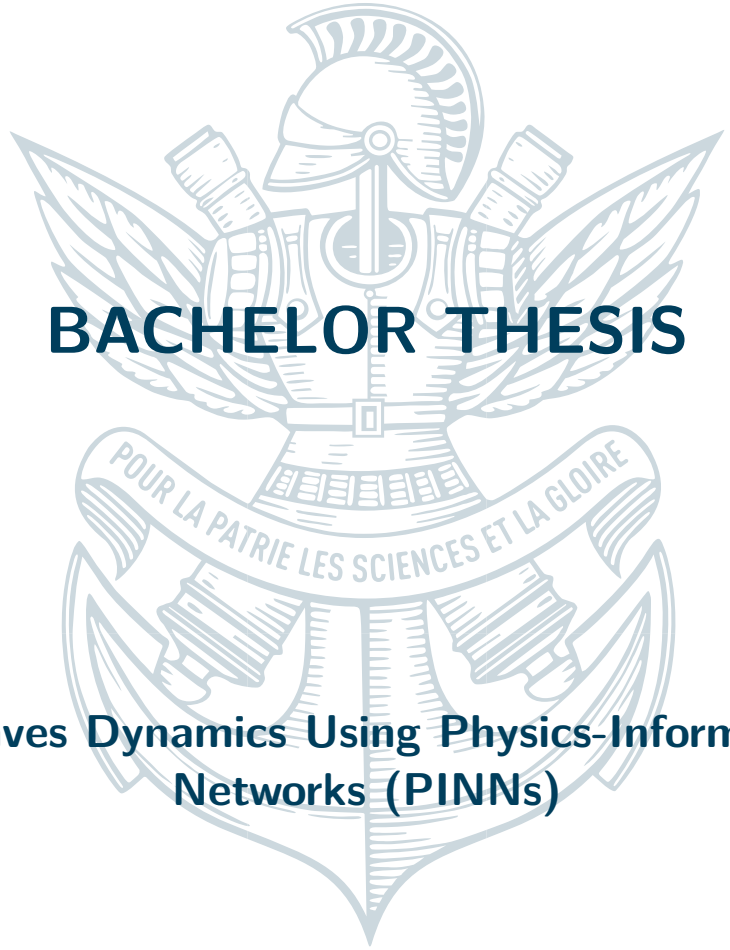
LadHyX



BACHELOR THESIS

Rogue Waves Dynamics Using Physics-Informed Neural Networks (PINNs)

March 20, 2025

Peter A. Sayegh[1]
Under the supervision of Prof. Christophe Josserand[2]

ÉCOLE POLYTECHNIQUE

IP PARIS

[1]Email: peter.sayegh@polytechnique.com
[2]Email: christophe.josserand@polytechnique.edu

Contents

Abstract

This report aims to get closer to understanding Rogue Waves dynamics and to leverage the flexibility of PINNs, a recent developing tool in Machine Learning, to solve nonlinear and complex partial differential equations. By introducing the Nonlinear Schrödinger Equation and some of its analytical solutions, we establish a theoretical foundation for modeling extreme wave events. We then develop a Physics-Informed Neural Network (PINN) framework tailored to approximate solutions of the equation under various initial and boundary conditions. Our approach is validated against known analytical solutions and numerical methods, demonstrating the effectiveness of PINNs in capturing rogue wave behavior. We also focus on the error of the PINN solution and the parameters that impact it more heavily. The results highlight the potential of deep learning techniques in advancing our understanding of nonlinear wave phenomena, paving the way for future explorations in physical oceanography and nonlinear optics.

1 Introduction

Rogue waves (RW), illustrated in Hokusai's painting in Figure 1, are rare, unpredictable surface waves that exceed twice the height of surrounding waves and can appear suddenly from unexpected directions. Their formation and evolution have been extensively analyzed in fluid mechanics, particularly in the seminal work of Dysthe et al. (2008) [2], which provides a comprehensive exploration of the underlying nonlinear and random processes. Unlike tsunamis, these localized phenomena—sometimes surpassing 30 meters—arise from nonlinear energy focusing and pose significant risks to even the largest ships.

The Nonlinear Schrödinger (NLS) equation serves as a model to study localized structures in a broad range of applications, going from Bose-Einstein condensates to superconductivity, including fiber optics communication and hydrodynamics, as N. Karjanto detailed in Ref. [6]. In the latter case, which we focus on in this work, the NLS models ocean surface waves evolution both in shallow and deep waters. It admits analytical solitary wave solutions referred to as solitons which can propagate over long distances without losing their integrity due to a precise balance between dispersion and nonlinearity.

Physics-Informed Neural Networks (PINNs) have recently emerged as a promising paradigm for solving PDEs, as initiated by Raissi et al. in 2019 [9], by integrating physical laws directly into the loss functions of neural networks. Unlike conventional numerical methods, PINNs are mesh-free and offer flexibility in representing complex solution spaces. By leveraging automatic differentiation, PINNs can compute derivatives of the neural network solution with respect to spatial and temporal variables, enabling efficient optimization and accurate solutions even for high-dimensional and irregular domains.

Figure 1: Katsushika Hokusai, *The Great Wave off Kanagawa*, 1831

2 Theoretical background

In this section, we investigate the physical concepts and computational tools that were key to our study and that we relied on for our numerical models.

2.1 The NLS Equation and Solitons in Fluid Dynamics

The one-dimensional (1D) NLS in the absence of viscosity and external forcing can be written as:

$$i\frac{\partial\psi}{\partial t} + \frac{\alpha}{2}\frac{\partial^2\psi}{\partial x^2} + g|\psi|^2\psi = 0 \tag{1}$$

where $\psi(x,t)$ is the complex wave function (considering $|\psi|^2$ represents the height of the wave), α is the dispersion coefficient, and g represents the strength and sign of the nonlinearity. The equation is said to be focusing if $g > 0$ and defocusing if $g < 0$, respectively with bright and dark solitons as analytical solutions.

In hydrodynamics, in the asymptotic limit ($L \to \infty$), the focusing NLS serves to model ocean surface waves in deep waters (i.e. of wavelength λ_{WW} much shorter than the ocean depth δ due to nonlinear effects and focusing current). In the inverse scenario ($\lambda_{WW} >> \delta$), for shallow waters and long wavelengths, one would use the defocusing NLS as a model.

Furthermore, regarding RW formation specifically, as they occur in deep waters, one would usually consider the focusing case. Its analytical solution, the Peregrine soliton corresponds to the RW; it sharpens until half of the time period then broadens and relaxes until then end of it where it fully collapses in the asymptotic limit ($T \to \infty$). Small-scale RWs were produced in a laboratory water wave tank in 2011, witnessing the accuracy of this simple model for RWs, as published in Ref. [7].

Conservation Laws: The 1D NLS conserves certain quantities, such as the total number of particles N, the Hamiltonian H (total energy), and momentum P:

$$N = \int |\psi|^2 \, dx.$$

$$H = \int \left(\frac{\alpha}{2}|\nabla\psi|^2 - \frac{g}{2}|\psi|^4 \right) dx,$$

where the first term represents the kinetic energy and the second term represents the potential energy due to nonlinearity.

$$P = \Im\left(\int (\psi^* \nabla\psi) \, dx \right),$$

Conservation of N, H, and P ensures the accuracy and stability of the numerical method. The analytical proof of their conservation is given in Appendix B.

2.2 Physics-Informed Neural Networks

The core idea behind PINNs is to approximate the solution of a PDE as a neural network, denoted $\psi_\theta(\mathbf{x}, t)$, where $\mathbf{x}$ and t are spatial and temporal variables, respectively, and θ represents the parameters of the neural network. The governing equations, boundary conditions, and initial conditions are encoded into the loss function of the network. For a generic PDE of the form:

$$\mathcal{D}[\psi] = f \quad \text{in a domain } \Omega,$$

where $\mathcal{D}$ is a differential operator, ψ is the solution, and f is a source term, the loss function is constructed as:

$$\mathcal{L}(\theta) = \lambda_r \|\mathcal{D}[\psi_\theta] - f\|^2 + \lambda_{BC}\mathcal{L}_{\text{BC}}(\theta) + \lambda_{IC}\mathcal{L}_{\text{IC}}(\theta),$$

where $\mathcal{L}_{\text{BC}}$ and $\mathcal{L}_{\text{IC}}$ are the boundary and initial conditions losses, respectively. By minimizing this loss function, the neural network learns a solution that satisfies both the governing equation and the physical constraints.

A key advantage of PINNs is the use of automatic differentiation, which enables the exact computation of derivatives of the network output with respect to its inputs, but also to its parameters θ for the loss function. This capability eliminates the need for numerical approximations of derivatives, enhancing accuracy and stability, particularly for high-dimensional problems.

In the context of RWs, the PINN framework is particularly appealing due to its flexibility and efficiency in handling multiscale phenomena. By incorporating the NLS equation, the PINN can learn wave dynamics, including the formation and evolution of solitons and rogue waves. Figure 2 displays a simplified scheme of the PINN training algorithm, proposed by K. Sun and X. Feng in Ref. [10].

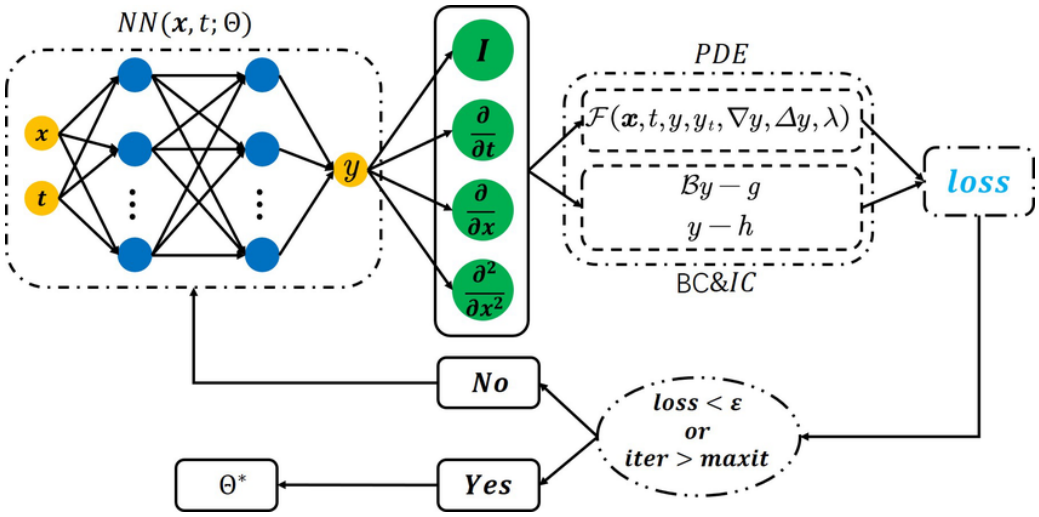


Figure 2: Simplified schematic of the PINNs model used to solve the NLS Equation

3 The inviscid one-dimensional NLS equation

To solve the defocusing 1D NLS, we have developed a spectral method, leveraging the efficiency of the Fast Fourier Transform (FFT) and spectral methods to handle spatial derivatives.

3.1 With spectral methods

3.1.1 The Split-Step Fourier/Strang spectral algorithm

The spectral algorithm involves the following steps:

Initialization:

- Define the spatial grid x and associated grid spacing Δx based on the domain size $2L$ and the number of grid points N.
- Compute the wavenumbers k using the FFT frequencies, scaled by π/L, and their squares k^2 for the dispersion term.

- Specify the initial condition $\psi(x, 0)$, typically chosen as a localized wave (e.g., Gaussian or hyperbolic secant profile) with or without a phase factor.
- Define the time step Δt, ensuring stability by satisfying the Courant-Friedrichs-Lewy (CFL) condition. For the 1D NLS, Δt can be chosen based on the relationship:

$$\Delta t \leq C \Delta x^2,$$

  where $C = 0.5$ is tuned empirically

One may rewrite equation 1 in the following form:

$$\frac{\partial \psi}{\partial t} = i(\hat{D} + \hat{N})\psi$$

where

$$\hat{D}\psi = \frac{\alpha}{2}\Delta\psi, \hat{N}\psi = g|\psi|^2\psi$$

We use a symmetric Strang splitting scheme for time evolution, where $\mathcal{F}$ denotes the Fourier Transform and $\mathcal{F}^{-1}$ the inverse transformation:

$$\psi(x, t + dt) = e^{-i\frac{dt}{2}\hat{N}}\mathcal{F}^{-1}\left(e^{-idt\hat{D}}\mathcal{F}(e^{-i\frac{dt}{2}\hat{N}}\psi(x, t))\right)$$

The algorithm proceeds in three steps: A Half-step nonlinear evolution in real space, then a Full-step linear evolution in Fourier space, and then another half-step nonlinear evolution in real space.

For the linear part, we work in Fourier space where the Laplacian becomes diagonal:

$$\mathcal{F}\{\Delta\psi\} = -k^2\hat{\psi}(k)$$

The time-evolution algorithm is outlined below:

Algorithm 1 Time Evolution Using the Spectral Method

Require: Initial wave function $\psi(x, 0)$, dispersion coefficient α, nonlinearity coefficient g, wavenumbers k, time step Δt, total time T.

1: Compute $N_t = T/\Delta t$ (total number of time steps).
2: **for** $n = 1$ to N_t **do**
3: **Half-step nonlinear:** $\psi \leftarrow e^{-i\frac{dt}{2}\hat{N}}\psi$ {Real space}
4: **FFT:** $\hat{\psi} \leftarrow \mathcal{F}\{\psi\}$ {To Fourier space}
5: **Full-step linear:** $\hat{\psi} \leftarrow e^{-idt\hat{D}}\hat{\psi}$ {Fourier space}
6: **IFFT:** $\psi \leftarrow \mathcal{F}^{-1}\{\hat{\psi}\}$ {Back to real space}
7: **Half-step nonlinear:** $\psi \leftarrow e^{-i\frac{dt}{2}\hat{N}}\psi$ {Real space}
8: **end for**

Boundary Conditions: The method implicitly assumes periodic boundary conditions due to the use of FFT: $\psi(L, t) = \psi(-L, t)$.

Complexity and Error Analysis

Since the FFT operations dominate, with a the complexity per time step of $O(N_x \log N_x)$, over $N_t = \frac{T}{dt}$ time steps, the total complexity becomes:

$$O\left(N_t \cdot N_x \log N_x\right).$$

Substituting $N_t \propto T \cdot N_x^2/L^2$, where $dt = 0.5 \cdot dx^2$ and $dx = 2L/N_x$, the total complexity scales as:

$$O\left(\frac{T}{2L^2} \cdot N_x^3 \log N_x\right).$$

The time evolution uses a Strang splitting method to alternate between the linear and nonlinear operators. The error of this method is known to scale as $O(dt^3)$. This arises because Strang splitting introduces local truncation errors proportional to dt^3 at each step, while ensuring global errors accumulate to the same order under smooth dynamics.

3.1.2 Verification with the analytical solution: dark solitons

Besides monitoring the conserved quantities of the NLS to validate our algorithm, one could also use exact solutions as follows. As mentioned by C. Josserand et Y. Pomeau in Ref. [4], the 1D defocusing NLS has analytical solutions, called dark solitons, of the form:

$$\psi(x,t) = \left[\nu \tanh\left(\nu(x - x_0 - \chi t)\right) + i\chi\right] e^{-it}, \text{ where } \nu^2 + \chi^2 = 1.$$

Here, the wavepacket is initially centered at x_0, has width $1/\nu$ and oriented velocity χ. Hence, since it is an analytical solution of the NLS, if it is given as initial condition of our simulation, the wavepacket moves at constant speed χ without deformation. We verify that this ansatz satisfies the defocusing NLS (with $\alpha = 1, g = -1$) is given in Appendix A. However, this function is not periodic in space which will make the FFT reinitialize the function by enforcing periodicity and the code will diverge or give inaccurate results. Instead, we test our algorithm with a product of two dark solitons symmetric with respect to 0 (at $\pm x_0$) and with opposite velocities ($\mp\chi$) as initial condition:

$$\psi_0(x) = \psi(x, t = 0) = (\nu \tanh(\nu(x - x_0)) + i\chi)(\nu \tanh(\nu(x + x_0)) - i\chi)$$

This function is pseudo-periodic and the FFT should be handled properly. Asymptotically, we have when $x \to \infty$, $\tanh(\nu(x + x_0) + \chi t) \to 1$ so $\psi \to (\nu + i\chi)(\nu - i\chi) = \nu^2 + \chi^2 = 1$. Solitons have the property to propagate without being deformed so one expects from the simulation that the two wavepackets will merge into a larger well before recovering and continue each in their direction. We plot our results and the verification tests hereafter in Figure 3.

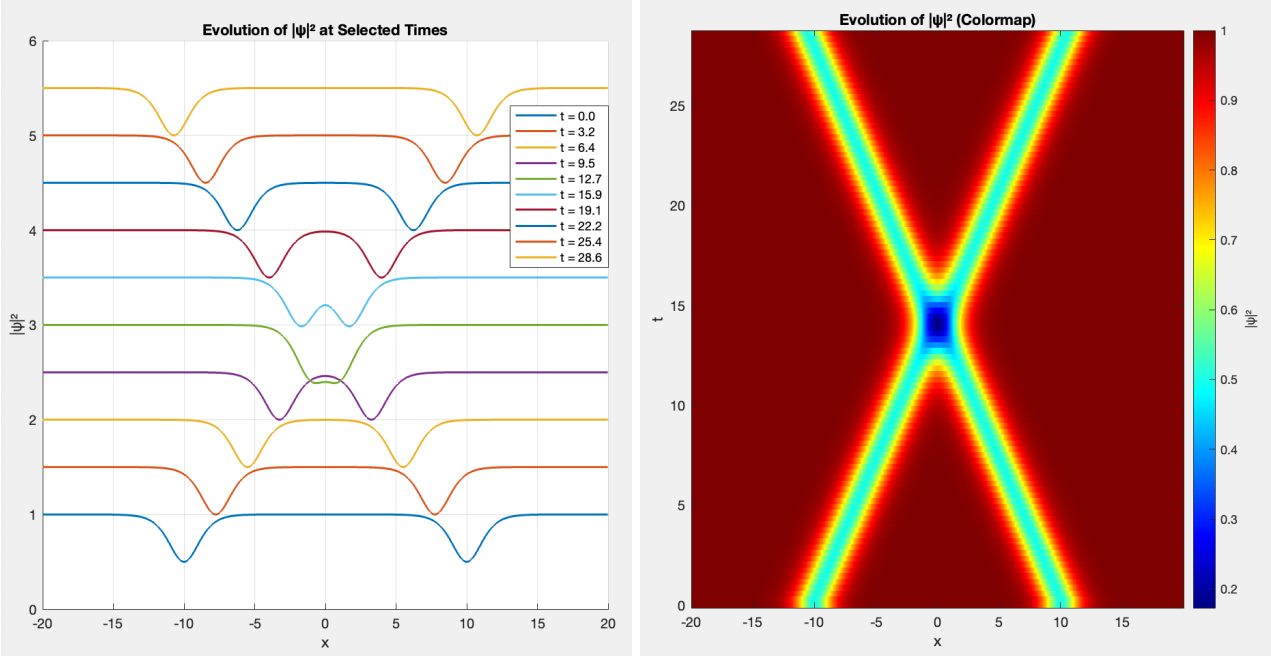


Figure 3: Evolution of two dark solitons in a symmetric setting at selected instants (left) and with a colormap 2D plot (right). $\nu = \chi = \frac{1}{\sqrt{2}}$, $L = 20, x_0 = L/2, N_x = 2048, \alpha = 1, g = -1, T = 28$

As one can notice, the wells always have exactly the same depth (or amplitude) except when they are merging, as predicted. The relative errors we obtain on the conserved quantities compared to the initial values are: $\epsilon_N = 8.8 \times 10^{-13}, \epsilon_H = 1.8 \times 10^{-7}, \epsilon_P = 1.1 \times 10^{-1}$. The energy error is higher than the mass one due to slight dispersion given by the Laplacian. The momentum error is also significantly higher than both, a reasonable explanation to that would be that the momentum is initially low as it is perfectly balanced between the two solitons moving at the same speed towards each other so at the end of the propagation time, they may have lost some of their momentum which explains the small gap since the beginning of the simulation. Theoretically, we would indeed expect a null momentum at all times.

From Figure 3, we observe that the peaks move in their direction at constant speed so one could track the position of the both minimizers of $|\psi|^2$ over time and study their evolution. One expects linear dependence with slopes $\pm\chi = \pm\frac{1}{\sqrt{2}} \approx \pm 0.707$ which are the velocities of the solitons. We showcase this evolution below in Figure 4.

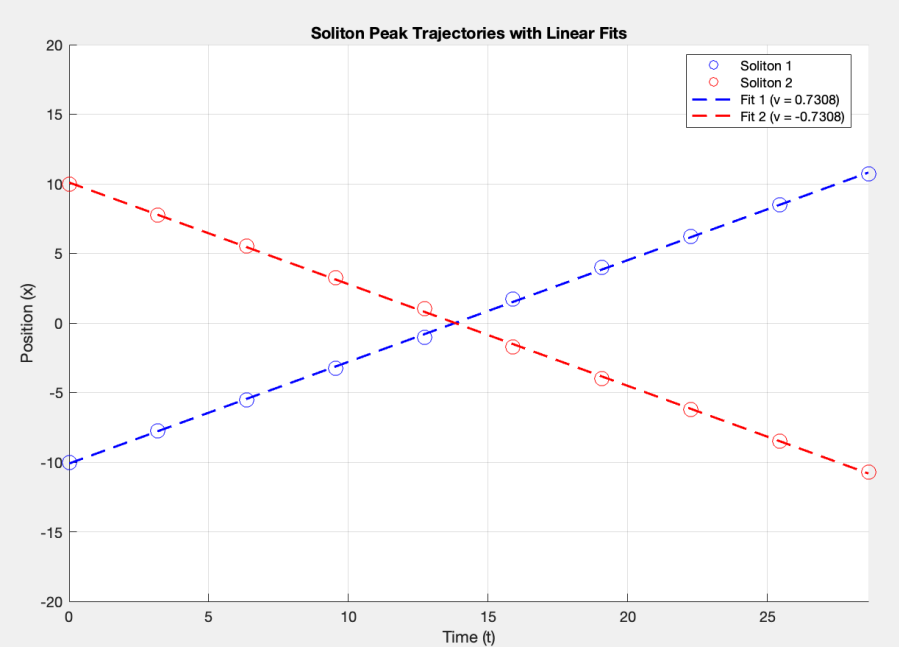


Figure 4: Evolution of the position of the minima of $|\psi|^2$ in time

We obtain velocities of ± 0.731 which gives a relative gap of 3.4% compared to $\pm\chi$ and which can be caused by the merge of the peaks slightly deviating from the linear fit.
The last check we perform relates to the periodicity of the numerical solution. Indeed, we are interested in what happens if we increase the propagation time such that it includes two periods and see if the solitons will reproduce the same behaviour another time. We include this outcome below in Figure 5.

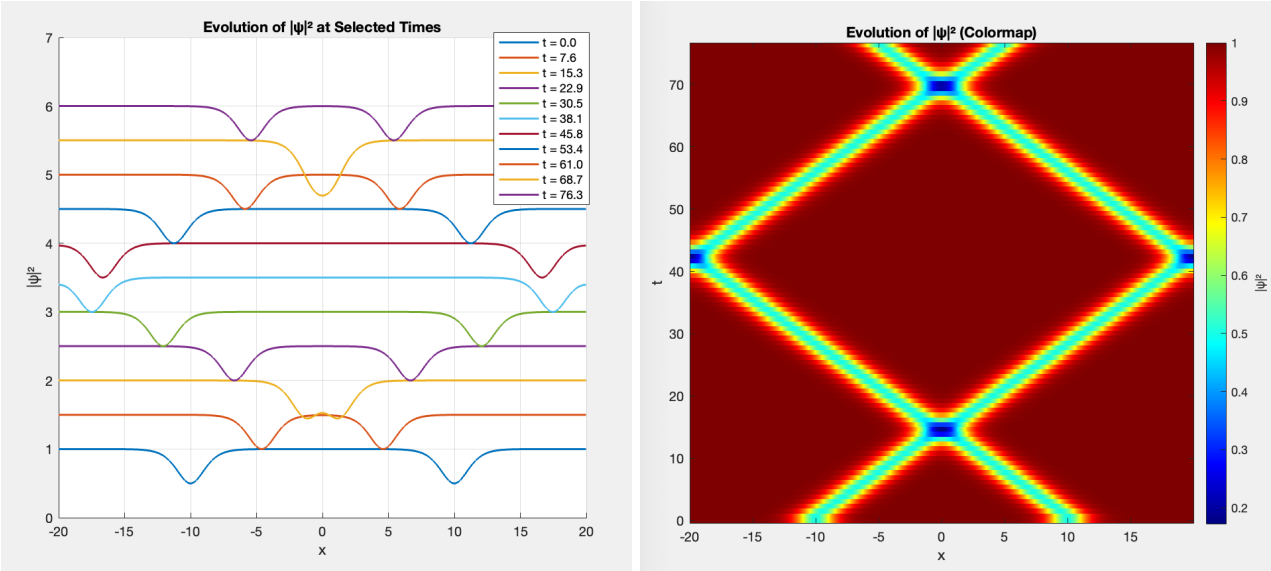


Figure 5: Extended evolution of two dark solitons in a symmetric setting at selected instants (left) and with a colormap 2D plot (right). $\nu = \chi = \frac{1}{\sqrt{2}}$, $L = 20, x_0 = L/2, N_x = 2048, \alpha = 1, g = -1, T = 76$

We observe that the localized wavepackets propagate the same way as before and merge a second time before recovering and separating again. The relative errors we obtain on the conserved quantities compared to the initial values are: $\epsilon_N = 2.3 \times 10^{-12}, \epsilon_H = 7.3 \times 10^{-5}, \epsilon_P = 6.7 \times 10^{-4}$. Notice that we obtain a drastically better momentum conservation and can be explained by the return of peaks closer to their initial position and oriented velocity, hence minimizing this relative deviation.

3.2 With PINNs

While spectral methods provide an efficient numerical approach to solving the 1D NLS equation, an alternative strategy using PINNs allows for a mesh-free approximation of the solution.

3.2.1 Network Architecture

The core of the PINN architecture is a feedforward fully connected neural network with multiple layers (also known as Multilayer Perceptron or MLP), each equipped with non-linear activation functions (tanh in our case). The network is trained to approximate the wave function $\psi(x,t)$, which is split into its real and imaginary components: $\psi(x,t) = u(x,t) + iv(x,t)$. Writing this in terms of the real and imaginary components, we obtain the coupled system that will define the PDE residuum loss:

$$\begin{cases} -v_t + \frac{\alpha}{2}u_{xx} + g(u^2+v^2)u = 0 \\ u_t + \frac{\alpha}{2}v_{xx} + g(u^2+v^2)v = 0 \end{cases}$$

The PINN maps the spatial and temporal coordinates (x,t) to the outputs (u,v) using a neural network:

$$f_\theta : (x,t) \mapsto (u_\theta(x,t), v_\theta(x,t))$$

where $\theta = \left\{W^{(k)}, b^{(k)}\right\}_{k=1}^{l}$ represents the trainable parameters (weights and biases) of the network. The architecture consists of:

- **Input Layer:** Two neurons corresponding to the spatial coordinate x and temporal coordinate t.
- **Feature Mapping:** The input x is transformed using periodic basis functions:

  $$\phi(x,t) = (\cos(\pi x/L), \sin(\pi x/L), t)$$

  This transformation ensures that the network inherently respects spatial periodic Dirichlet boundary conditions $\psi(L,t) = \psi(-L,t)$ according to Hao et al. (2024) in Ref. [3]. Also, this adds another dimension to the input layer. Note that if this feature mapping conflicts with Neumann zero-slope boundary conditions if these are imposed in the problem.
- **Hidden Layers:** l layers, each containing w neurons and a tanh activation function. Typically $l \approx 6-8$ and $w \approx 40-50$ depending on the complexity of the problem.
- **Output Layer:** Two neurons producing $u(x,t)$ and $v(x,t)$, without an activation function.

Each layer applies an affine transformation followed by the tanh activation function:

$$z^{(k+1)} = \tanh\left(W^{(k)}z^{(k)} + b^{(k)}\right)$$

where $W^{(k)} \in \mathbb{R}^{w \times w}$ is the weight matrix , $b^{(k)} \in \mathbb{R}^w$ is the bias vector, $z^{(k)}$ the activation vector for layer k.

The final output is computed as:

$$(u_{\theta^*}, v_{\theta^*}) = W^{(l)} z^{(l)} + b^{(l)}$$

where $W^{(l)}$ and $b^{(l)}$ correspond to the last layer.

3.2.2 The Total Loss function

As previously introduced, the total loss function $\mathcal{L}$ minimized by the PINN consists of a weighted sum of three main components:

$$\mathcal{L}(\theta) = \lambda_r \mathcal{L}_r(\theta) + \lambda_{IC} \mathcal{L}_{IC}(\theta) + \lambda_{BC} \mathcal{L}_{BC}(\theta),$$

where we remind:

- $\mathcal{L}_r$ is the NLS residual loss, $\mathcal{L}_{IC}$ is the initial condition (IC) loss, enforcing consistency with the initial wave profile, $\mathcal{L}_{BC}$ is the boundary condition (BC) loss, ensuring periodicity in space;
- $\lambda_r, \lambda_{IC}, \lambda_{BC}$ are weights for each source of loss ($\lambda_{BC} = 0$ when Dirichlet BCs are imposed since they are inherited from the feature mapping).

The residual loss is defined as:

$$\mathcal{L}_r(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} \left| -v_{\theta,t}(x_i, t_i) + \frac{\alpha}{2} u_{\theta,xx}(x_i, t_i) + g(u_\theta^2 + v_\theta^2) u_\theta \right|^2 + \left| u_{\theta,t}(x_i, t_i) + \frac{\alpha}{2} v_{\theta,xx}(x_i, t_i) + g(u_\theta^2 + v_\theta^2) v_\theta \right|^2.$$

where (x_i, t_i) are the collocation points sampled in the computational domain, N_r is the total number of collocation points, and u_θ, v_θ are the neural network outputs for the real and imaginary parts of ψ_θ.

One could rewrite the loss function as:

$$\mathcal{L}_r(\theta) = \frac{1}{N_r} \sum_{i=1}^{N_r} r_\theta(x_i, t_i)$$

with r_θ is the residual magnitude.

The IC loss is given by the initial wave profile $\psi(x, 0) = \psi_0(x) = u_0(x) + i v_0(x)$ and the IC batch size N_{IC}:

$$\mathcal{L}_{IC}(\theta) = \frac{1}{N_{IC}} \sum_{i=1}^{N_{IC}} |u_\theta(x_i, 0) - u_0(x_i)|^2 + |v_\theta(x_i, 0) - v_0(x_i)|^2 .$$

Finally, in case of Neumann zero-slope boundary conditions, the derivatives of the real and imaginary components must vanish at the boundaries:

$$\left.\frac{\partial \psi}{\partial x}\right|_{x=L} = \left.\frac{\partial \psi}{\partial x}\right|_{x=-L} = 0.$$

This is enforced by defining the following boundary condition loss function:

$$\mathcal{L}_{BC}(\theta) = \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} \left| \frac{\partial u_\theta}{\partial x}(L, t_i) \right|^2 + \left| \frac{\partial u_\theta}{\partial x}(-L, t_i) \right|^2 + \left| \frac{\partial v_\theta}{\partial x}(L, t_i) \right|^2 + \left| \frac{\partial v_\theta}{\partial x}(-L, t_i) \right|^2 .$$

where N_{BC} is the number of sampled boundary points.

The trainable parameters θ are optimized using Adam gradient descent, with updates performed as:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \mathcal{L}(\theta^{(t)}),$$

where η is the learning rate, which is progressively decayed over epochs, and the gradient with respect to θ is computed using back-propagation.

3.2.3 Optimization methods for training

Training a PINN on long time horizons often results in instability and slow convergence due to approximation errors accumulating over time. To address this, we employ two techniques: Causal training and Residual-Based Adaptive (RBA) Sampling.

Causal Training: Conventional PINNs minimize the PDE residual $\mathcal{L}_r(\theta)$ over the entire time domain simultaneously, often violating causality by attempting to optimize solutions at later times before accurately resolving earlier states.

To explicitly enforce causality, inspired by Wang et al. (2024) [12], we introduce a time-dependent weighting function:

$$\mathcal{L}_r(\theta) = \frac{1}{N_r} \sum_{j=1}^{N_r} w_j r_\theta(x_j, t_j),$$

where:

- $w_j = \exp\left(-\epsilon \sum_{k=1}^{j-1} \sum_{i=1}^{N_x} r_\theta(x_i, t_k)\right)$ is the **causality weight**, ensuring that later-time errors are only minimized if earlier residuals are sufficiently small.
- ϵ is a tunable causality parameter controlling the sharpness of the weighting decay.

This strategy prevents the network from prematurely focusing on later timesteps, ensuring that early-time dynamics are learned first.

RBA Sampling: Following Wu et al. [13], we dynamically select training points based on the magnitude of PDE residuals, allowing the network to focus on regions with large errors. We define the probability of selecting a training point (x, t) as:

$$p_\theta(x, t) = \frac{r_\theta(x, t)}{\sum_{i,j} r_\theta(x_i, t_j)},$$

where the residual magnitude $r_\theta(x, t)$ was defined earlier. Then, to dynamically adjust the importance of each training point based on the PDE residuals, we update the weights at each iteration as follows:

$$\omega^{(i+1)}(x,t) = \gamma\omega^{(i)}(x,t) + (1-\gamma)\frac{r(x,t)}{\max r(x,t)}$$

where γ is called a momentum parameter between 0 and 1 (typically 0.8-0.9) controlling the smoothing of weights updates.
The advantages of this approach are the focus on higher error regions and an improved convergence as fewer training iterations are needed since the model learns more efficiently.

The full training algorithm is summarized and outlined below:

Algorithm 2 PINN for the NLS Equation

Require: Spatial domain $[x_{\min}, x_{\max}]$, temporal domain $[t_{\min}, t_{\max}]$, network parameters θ, and batch sizes N_r, N_{IC}, N_{BC}

Ensure: Trained neural network f_θ approximating $\psi(x,t) = u(x,t) + iv(x,t)$

Initialize Model

Define PINN architecture $f_\theta : (x,t) \mapsto (u_\theta, v_\theta)$
Apply feature mapping $x \to (\cos(\pi x/L), \sin(\pi x/L))$ for periodic BCs
Initialize RBA weights $\omega(x,t)$ and set batch sizes
for epoch $= 1$ to max_epochs **do**
  Sample training points:
    $(x_r, t_r) \sim p_\theta(x,t) \propto \omega(x,t)$ (interior)
    $(x_0, 0)$ (initial condition)
    $(x_b = \pm L, t_b)$ (boundary condition)
  Compute predictions $\psi_\theta = f_\theta(x,t)$ and derivatives via AD
  Compute PDE residuals $r_{\theta,real}, r_{\theta,imag}$ and total residual r_θ
  Compute loss terms $\mathcal{L}_r(\theta), \mathcal{L}_{IC}(\theta), \mathcal{L}_{BC}(\theta)$
  Compute total loss $\mathcal{L}(\theta) = \lambda_r\mathcal{L}_r + \lambda_{IC}\mathcal{L}_{IC} + \lambda_{BC}\mathcal{L}_{BC}$
  Update parameters via Adam: $\theta \leftarrow \theta - \eta\nabla_\theta\mathcal{L}(\theta)$
  Update RBA weights:
    $\omega^{(i+1)}(x,t) = \gamma\omega^{(i)}(x,t) + (1-\gamma)\frac{r_\theta(x,t)}{\max r_\theta(x,t)}$
  if epoch % check_frequency $= 0$ **then**
    Compute validation metrics (error norms, conservation laws)
    Adjust learning rate: $\eta \leftarrow \eta \cdot$ decay_rate
  end if
end for
Return trained model f_θ.

3.2.4 Testing and Validation

To assess the accuracy and reliability of the trained PINN, we validate its predictions using:

1. Comparison with Reference Solutions: We evaluate the PINN predictions against analytical solutions (if available) for specific test cases like dark solitons, but also Spectral Method Solutions, using the Split-Step Fourier Method.

2. Error Metrics: We compute the following L^2-error norm:

$$\mathcal{E}_{L^2} = \frac{\|\psi_\theta - \psi_{\text{ref}}\|_2}{\|\psi_{\text{ref}}\|_2}$$

These quantify the relative deviation from reference solutions where $||\psi||_2 = (\int |\psi|^2 dx)^{\frac{1}{2}}$

3. Conserved quantities (Particle Number, Hamiltonian, and Momentum)

Case (a): dark soliton IC with Neumann zero-slope BC

We display the output of our PINN below (in Figure 6) along with the training metrics (i.e. the loss function and L^2 error function) evolution with respect to both time and number of iterations.

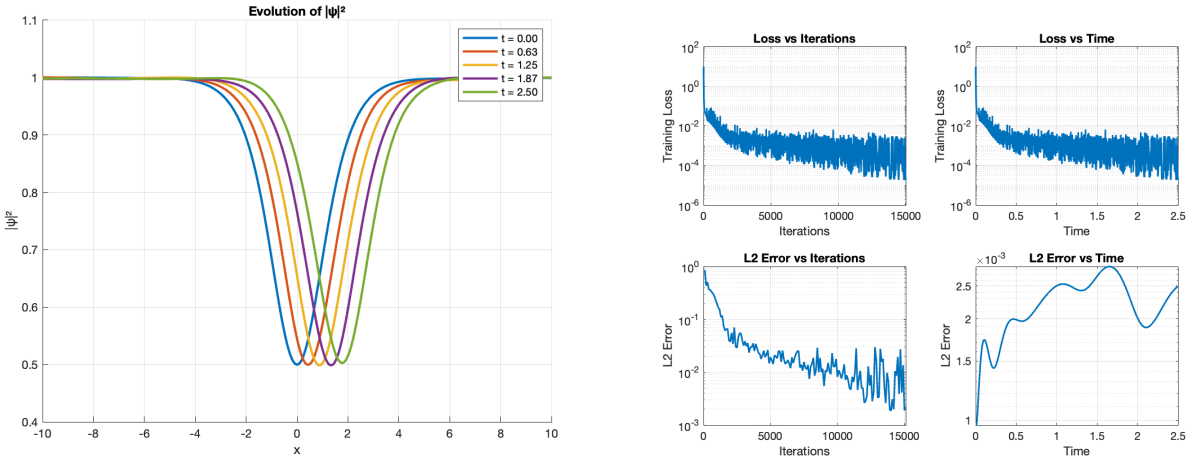


Figure 6: Evolution of $|\psi|^2$ (left) and training metrics (right) given by the PINN for $\alpha = 1, g = -1, L = 10, T = 2.5, N_x = 512, N_t = 256, l = 5, w = 48, \nu = \chi = \frac{1}{\sqrt{2}}, x_0 = 0, (\lambda_r, \lambda_{IC}, \lambda_{BC}) = (1, 10, 10), \epsilon = 5, \gamma = 0.8, (N_r, N_{IC}, N_{BC}) = (1024, 512, 512), N_{iter} = 15000$

Given the initial and boundary conditions, one expects the soliton to displace along the x-axis at speed χ with no deformation. We can still see that the wavepacket gets slightly shallower at the end of the simulation. We look at the training metrics to quantify the accuracy.

We observe that the L^2 error quickly increases in time and the training loss is progressively decreasing. We obtain an average error $\langle \mathcal{E}_{L^2} \rangle = 2.24 \times 10^{-3}$ and the training loss reaches the order of 10^{-5} by the end of the iterations.

Case (b): 2 dark solitons as IC, with Dirichlet periodic BCs

Now we are interested in the case of two dark solitons as the initial condition using feature mapping for periodicity and no boundary condition in the loss function. We can also compare the PINN output to the spectral method solution by monitoring the time-evolution of both the conserved quantities (previously denoted N, H, P) and the (spatial) average of the L^2-error.

Figure 7 illustrates the comparison between the PINN and spectral solutions over time. Initially, the PINN solution closely matches the spectral solution, revealing good agreement. However, as time progresses, discrepancies become more pronounced, indicating a gradual divergence. Additionally, symmetry in the solution is well-preserved at early times but degrades towards the end of the simulation period, as one can notice in the bottom curve of Figure 7, around $x = 0$ at $t = T$.

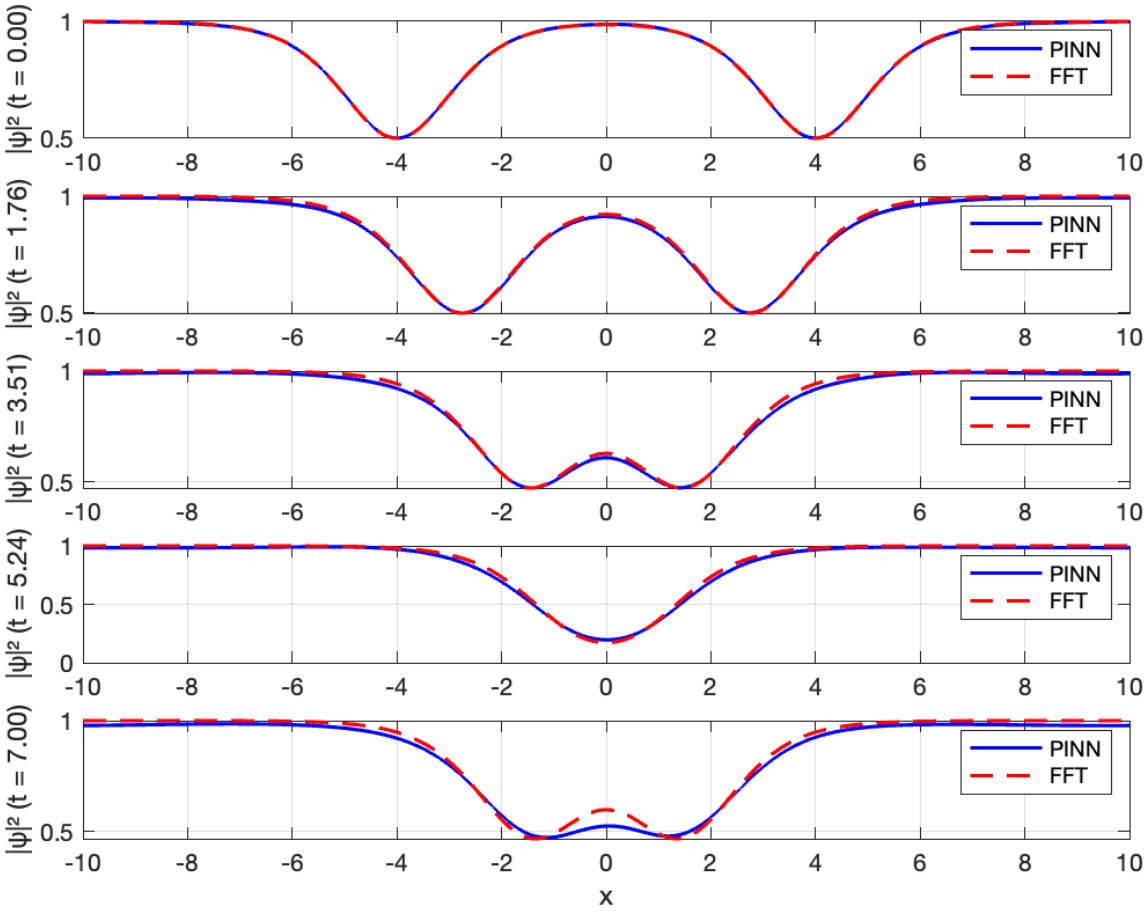


Figure 7: Evolution of $|\psi|^2$ given by the PINN and compared with the FFT-based method, for $\alpha = 1, g = -1, L = 10, T = 7, N_x = 600, N_t = 250, l = 8, w = 48, \nu = \chi = \frac{1}{\sqrt{2}}, x_0 = -4, (\lambda_r, \lambda_{IC}, \lambda_{BC}) = (1, 1, 0), \epsilon = 5, \gamma = 0.8, (N_r, N_{IC}, N_{BC}) = (1024, 300, 200), N_{iter} = 15000$

Then, looking at the conserved quantities in Figure 8, the particle number is clearly better conserved by the FFT-based method, yet PINNs still give a $\epsilon_N = 2.0\%$ relative deviation, $\epsilon_H = 1.2\%$ for the Hamiltonian, and finally, we see that both the FFT and the PINN preserve a null momentum, expected given the symmetry of the wavepacket. It is quite surprising that the PINN solution conserves the energy better than the mass. Regarding the L^2 error, it starts with an immediate loss in accuracy that progressively increases with time which restricts the time-domain to stay within the order of the unit.

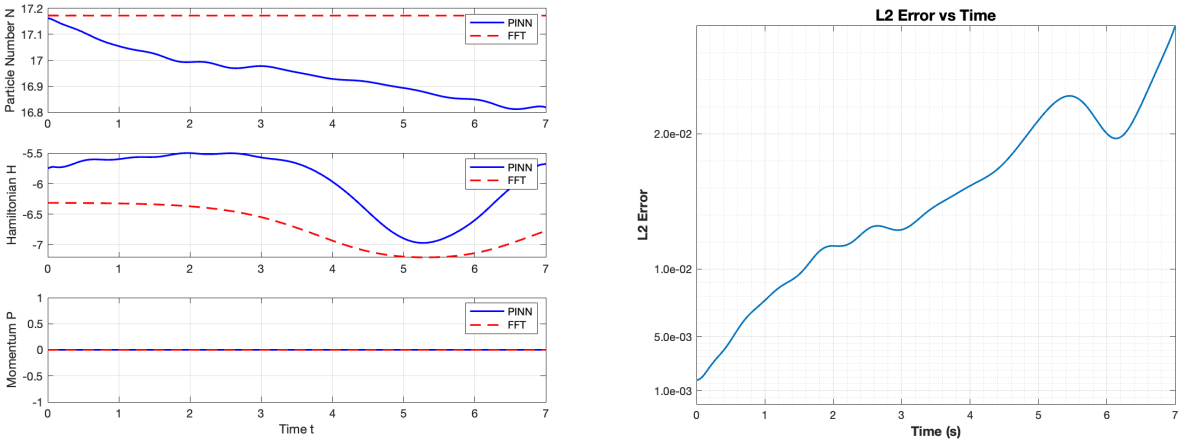


Figure 8: Time-evolution conserved quantities both with the spectral and the PINN algorithm (left) and L^2-error (right) corresponding to the previous simulation, i.e. case (b)

3.2.5 Complexity, error, and convergence analysis

Complexity of the PINN algorithm The complexity of the PINN algorithm can be analyzed based on the following components:

1. **Forward Evaluation:** The network consists of l layers, each with w neurons, leading to a per-layer complexity of $O(w^2)$. With l layers, this results in an overall feedforward pass complexity of $O(lw^2)$.

2. **Loss Function Computation:** The residual loss requires evaluating the PDE at N_r collocation points chosen in a grid of size $N_t \times N_x$, leading to a per-iteration complexity of $O(N_r l w^2)$.

The total complexity hence scales as: $O(N_{\text{iter}} N_r l w^2)$, where N_{iter} is the number of training iterations. For comparison, the spectral method using FFT has a complexity of $O(N_t N_x \log N_x)$[3].

Thus, PINNs have a significantly higher computational cost due to mainly the large number of training iterations required for convergence and the increased cost of evaluating derivatives compared to the efficient spectral differentiation in FFT.

Pointwise Error on PINNs We define the pointwise error at each point (x,t) by:

$$E(x,t) = |\psi_{PINN}(x,t) - \psi_{ref}(x,t)|^2$$

to evaluate where in the space-time domain the error is the highest, ψ_{ref} being given by either the explicit analytical solution like in case (a) or the output of the FFT-based spectral method as in (b). One can expect higher errors for later times as well as for positions close to 0 due to the solitons merging and interaction for instance. We display a 3D-plot of that error function in Figure 9, which confirms our predictions. Note that the error is globally higher for the two-soliton case which is not too surprising as one can expect that both solitons dynamics are harder to learn than a single one.

[3]Note that N_t, N_x are not the same in the spectral method and the PINNs sampling grid for automatic differentiation

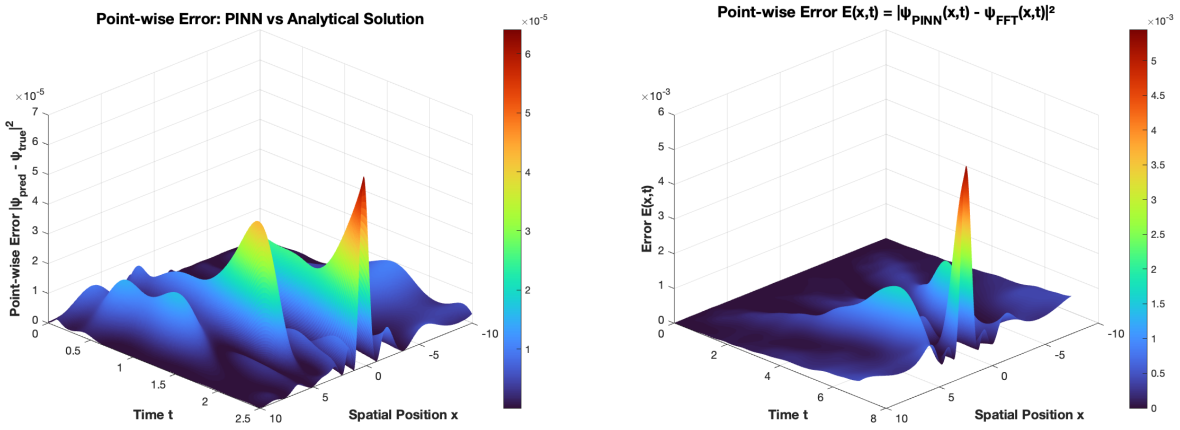


Figure 9: Surface plot of the pointwise error applied to both cases (a) (left) and (b) (right) of the previous simulation

Error dependence on network structure and resolution Here we are interested in finding the dependence of the error on the depth and width of the PINN we train (i.e. number of hidden layers l and number of neurons per layer w) as well as on the sampling resolution $(dx, dt) = \left(\frac{2L}{N_x}, \frac{T}{N_t}\right)$. For that purpose, we collect data of the temporal averaged L^2 error $\langle \mathcal{E}_{L^2} \rangle$ using case (b) as a reference and changing the structure on the one hand then changing the number of points on the other hand.

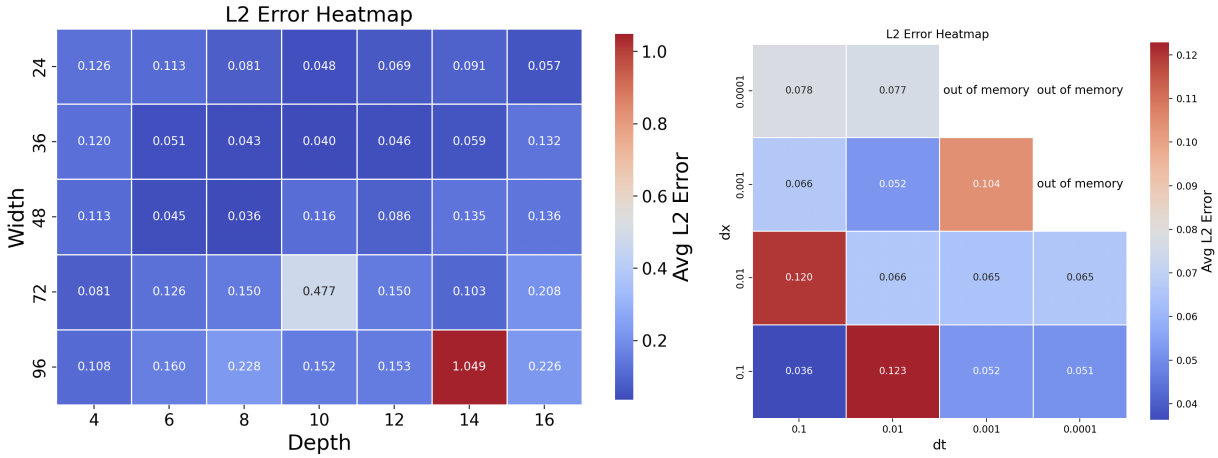


Figure 10: Error heatmap of the PINN solution against its depth and width with fixed resolution $(dx, dt) = (0.039, 0.028)$ (left) and against its resolution with fixed structure $(l, w) = (8, 48)$ (right)

One can first notice that both distributions in Figure 10 of data points are nontrivial revealing that the optimal structure and resolution are hard to find analytically. Interestingly, the left plot highlights that there's no straightforward relationship where merely increasing the size (depth or width) uniformly improves performance. Instead, certain specific combinations—such as 8 hidden layers with 48 neurons each—yield notably lower errors. Regarding the right plot, it is quite surprising that the lowest error is achieved with the least precise resolution, i.e. $dx = dt = 0.1$. Moreover, we could not evaluate the PINN accuracy for ultra precise resolution due to insufficient memory of the machine used to run our

simulations.

4 The Rogue Wave Equation

4.1 Mathematical Formulation

Building upon our analysis of the defocusing NLS equation and the methodologies developed for both spectral and PINN-based approaches, we now turn our attention to a modified form of the NLS equation that specifically governs RW phenomena. This extension is inspired by the recent work of Y. Chen et al. (2024)[1], who explored multi-view transfer learning to enhance the accuracy of PINNs for NLS equations. The modified defocusing NLS equation with time-dependent potential that supports RW solutions can be expressed as:

$$i\frac{\partial \psi}{\partial t} = -\frac{\alpha}{2}\frac{\partial^2 \psi}{\partial x^2} + V(x,t)\psi - g|\psi|^2\psi$$

where $V(x,t)$ is a spatio-temporal potential defined as:

$$V(x,t) = \frac{4(x^2 - (t-\frac{T}{2})^2) - 1}{(x^2 + (t-\frac{T}{2})^2 + 0.25)^2} - 2$$

The potential introduces a localized focusing effect that counterbalances the defocusing nonlinearity, enabling the emergence of RWs, as showcased by Wang and Yan (2021) [11]. As mentioned in their paper, the exact Rogue wave solution to this equation takes the form of a Peregrine Soliton:

$$\psi(x,t) = \left[1 - \frac{4(1+2i(t-\frac{T}{2}))}{4x^2 + 4(t-\frac{T}{2})^2 + 1}\right] e^{i(t-\frac{T}{2})}, t \in [0,T]$$ [4]

Theoretically, one expects the wavepacket to be localized around $x = 0$, sharpened between $t = 0$ and $t = T/2$ where it reaches its maximum amplitude, then broadened again until flat. We plot our results in Figure 11 which confirms our expectations.

For the spectral method, the time-dependent potential necessitates modifications to the Strang splitting scheme. Specifically, the nonlinear operator now includes both the quadratic nonlinearity and the spatio-temporal potential:

$$\hat{N}\psi = g|\psi|^2\psi - V(x,t)\psi$$

However, since the exact analytical solution is known, we do not need the spectral method here to validate the PINN solution, it also diverges if the initial condition is a single RW since it is not periodic. Regarding the PINN result, the residual loss function is updated to incorporate the potential term:

$$r_\theta(x,t) = \left|-v_{\theta,t} + \frac{\alpha}{2}u_{\theta,xx} - V(x,t)u_\theta + g(u_\theta^2 + v_\theta^2)u_\theta\right|^2 + \left|u_{\theta,t} + \frac{\alpha}{2}v_{\theta,xx} - V(x,t)v_\theta + g(u_\theta^2 + v_\theta^2)v_\theta\right|^2$$

The initial condition is set to match the analytical solution at $t = 0$:

[4] Note that the Peregrine soliton is conventionally taken at real times $t \in \mathbb{R}$ or $t \in \left[-\frac{T}{2}, \frac{T}{2}\right]$, but we do a shift by T/2 to avoid dealing with negative times, such that the maximum is obtained at $x = 0, t = T/2$

$$\psi(x,0) = \left[1 - \frac{4(1-iT)}{4x^2+T^2+1}\right] e^{-i\frac{T}{2}}$$

For boundary conditions, we impose Neumann zero-slope boundary conditions:

$$\frac{\partial \psi}{\partial x}\Big|_{x=L} = \frac{\partial \psi}{\partial x}\Big|_{x=-L} = 0.$$

The same causal training and residual-based adaptive sampling strategies are employed to enhance the network's performance, with particular attention to the region where the rogue wave forms.

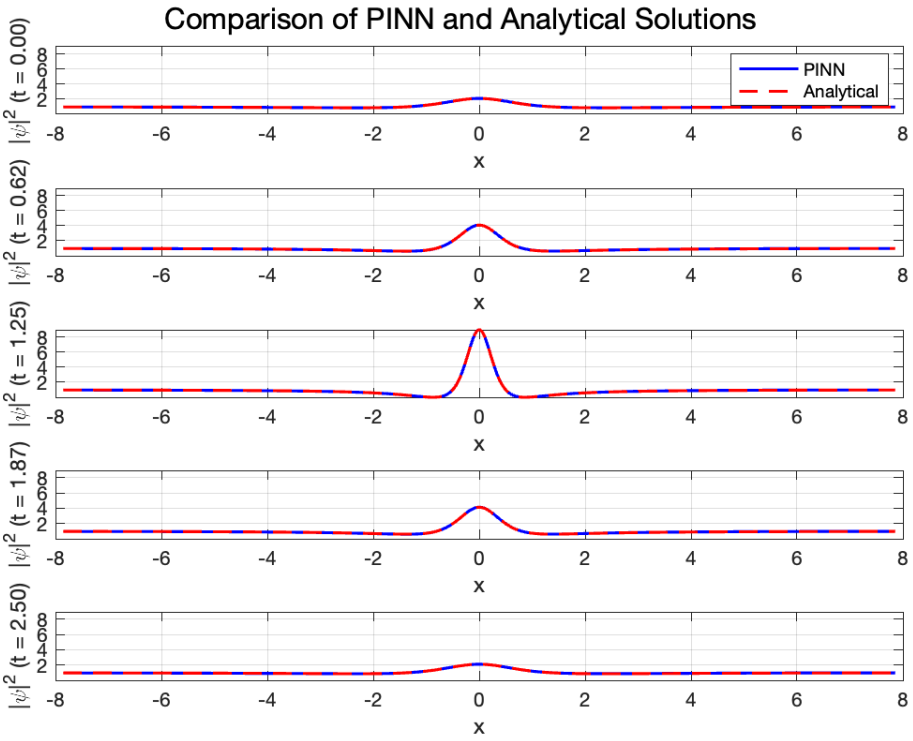


Figure 11: Evolution of $|\psi|^2$ given by the PINN and compared with the exact solution, for $\alpha = 1, g = -1, L = 2.5\pi, T = 2.5, N_x = 600, N_t = 250, l = 5, w = 48, (\lambda_r, \lambda_{IC}, \lambda_{BC}) = (1, 10, 10), \epsilon = 5, \gamma = 0.8, (N_r, N_{IC}, N_{BC}) = (1024, 300, 200), N_{iter} = 20000$

4.2 Error Analysis

After observing the quasi perfect match between the PINN solution and the analytical RW solution in Figure 11, we display the pointwise error between those two below as well as the training metrics plotted against time units and iterations in Figure 12.

Since the RW reaches its peak amplitude at $t = T/2$ and $x = 0$, it is predictable that the error would be higher closer to this instant and position. The L^2 error decreases log-linearly with respect to iterations and globally increases when computed against time at a sharper slope closer to the beginning of the simulation. The average error we obtain is $\langle \mathcal{E}_{L^2} \rangle = 4.84 \times 10^{-3}$.

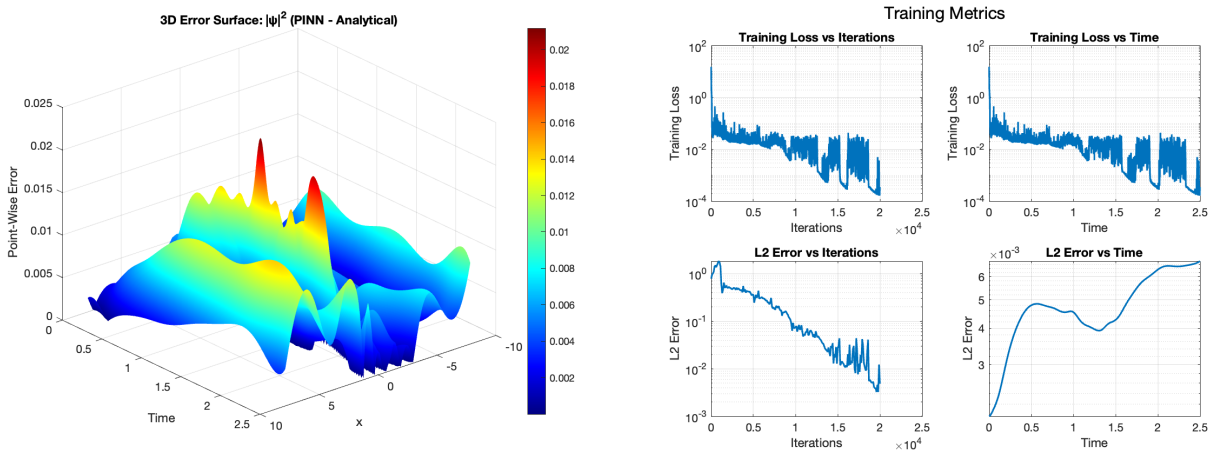


Figure 12: Pointwise Error and training metrics evolution in the previous Rogue Wave simulation

4.3 RW solution of the focusing NLS

In this work, we only treated the defocusing NLS so far, yet the Peregrine soliton also happens to be an exact solution of the focusing 1D NLS ($g = 1$) free of any external potential, as proposed by Howell Peregrine himself in 1983 [8]:

$$i\psi_t + \frac{1}{2}\psi_{xx} + |\psi|^2\psi = 0$$

Therefore, if we keep the previous simulation as a reference, one only needs to set the potential V to 0 and switch the sign of g to solve the focusing NLS admitting exactly the same solution. We plot the results and errors in Appendix D (Figures 13 and 14).
Note that this simulation required more training steps ($N_{iter} = 35000$) and collocation points ($N_r = 2048$) so one might infer that adding the external potential can facilitate the training which is coherent since the potential shapes the soliton. However, the average L^2 error is still an order above what we obtained in the defocusing case with external potential ($\langle \mathcal{E}_{L^2} \rangle = 4.29 \times 10^{-2}$). Regarding the pointwise error, as expected, it's significantly higher at times around T/2 and positions near zero.

5 Results and Discussion

5.1 Main findings

Our study demonstrates that PINNs can approximate solutions to the NLS equation and Rogue Wave solutions with reasonable accuracy, although they remain computationally expensive compared to spectral methods. The error between the PINN and reference solutions remains low at early times but accumulates over longer time horizons or at times where the wave dynamics are more complicated. PINN training also benefits from causal training and residual-based adaptive sampling, and possibly many more computational optimization methods, hence reducing instability.

5.2 The flexibility of PINNs

PINNs revealed to be highly practical to move from a simulation to another, one just needs to change the initial condition, the boundary condition (the latter depending on the spatial periodicity of the

former), and the residual loss functions and weights with some possible structure tuning depending on the complexity of the problem. Indeed, we also observed in Figure 10 that PINNs are likely more sensitive to their structure than their resolution to yield a decent accuracy. Their main drawbacks would be long runtime and large memory occupation although these can be mitigated through better hardware.

6 Conclusion

6.1 Summary of Findings

In this work, we have explored the application of Physics-Informed Neural Networks to solve the (1+1)-dimensional Nonlinear Schrödinger equation, comparing its performance against spectral methods based on the Fast Fourier Transform. While the latter provide computational efficiency and accuracy, PINNs offer flexibility in handling irregular domains and incorporating physics-based constraints.

Our study analyzed the error, convergence properties, and computational complexity of PINNs. The pointwise error between PINN and reference solutions revealed that while PINNs can approximate solutions with reasonable accuracy, the error accumulates over long time horizons. The impact of network depth, width, and resolution on error was also investigated, showing a nontrivial relationship between those and the accuracy of the solution.

A key takeaway from the complexity analysis is that PINNs are computationally more expensive than FFT-based methods due to their iterative training process, automatic differentiation overhead, and optimization complexity. However, PINNs remain promising for problems where spectral methods struggle, such as in cases involving complex boundary conditions or non-periodic ICs.

6.2 Limitations and Prospective Work

PINNs require significantly more iterations than spectral methods. The pointwise error between PINN and FFT solutions grows over time, limiting long-time stability. Moreover, PINN performance depends strongly on hyperparameters such as network depth, width, time resolution, learning rate, and more. To enhance the accuracy and efficiency of PINNs for solving the NLS equation, several optimization strategies are proposed:

- **Dynamic Loss Weighting:** Future work will explore adaptive reweighting of the loss function components to prioritize PDE residual minimization while ensuring boundary and initial conditions are accurately enforced.
- **Loss Function from Conserved Quantities:** Introducing additional regularization terms based on conservation laws, such as the Number of Particles (N), Hamiltonian (H), and Momentum (P), may improve long-term stability by reducing cumulative errors. The proposed loss function would be of the form:

  $$\mathcal{L}_{\text{CQ}} = \alpha_N |N(t) - N(0)|^2 + \alpha_H |H(t) - H(0)|^2 + \alpha_P |P(t) - P(0)|^2.$$

  Enforcing these physical invariants within the optimization framework could lead to faster and more accurate training.
- **Interaction of 2 or 3 Rogue Waves:** One might consider simulating the interaction of several RWs to check whether the PINN can capture these dynamics efficiently. This will imply giving a pseudo-symmetric initial condition to compare the solution with the spectral method result.

- **The full (2+1)-dimensional NLS for turbulence:** Furthermore, we may modify the PINN structure and training methods to solve the full NLS equation that governs water wave turbulence, as done in Reference [5]:

$$i\frac{\partial \psi}{\partial t} = -\frac{\alpha}{2}\Delta\psi - g|\psi|^{2n}\psi - i\nu\Delta^2\psi + f_{k_0}(\mathbf{x}, t),$$

  where n determines the nonlinearity order, ν represents viscosity-induced dissipation, and $f_{k_0}(\mathbf{x}, t)$ is an external forcing term, ($\mathbf{x} \in \mathbf{R}^2$, $n > 1$), for more realistic scenarios. Naturally, we expect to have more training steps, collocation points, optimization in the training processes. Then, from a more physical perspective, that would include studying the forcing sources (wind, change of density,...) as well as the ocean-air interface as a supplementary boundary condition.

By implementing these improvements, we aim to enhance the performance of PINNs for solving nonlinear wave equations, making them a more viable alternative to traditional numerical solvers in complex domains. Moreover, regarding RWs specifically, achieving a realistic and accurate simulation may drastically help oceanographers understanding the birth and impact of these extreme events.

Acknowledgements

Special thanks to Prof. Christophe Josserand for supervising the project and for providing key guidance and insights all along the internship. Thanks to PhD Matteo Tiritera who leveraged his expertise in PINNs to help me design and train one for my simulations. Finally, thank you to LadHyX, the laboratory that hosted me so kindly.

References

[1] Yikai Chen, Hongli Xiao, Xiao Teng, Wenjun Liu, and Long Lan. Enhancing accuracy of physically informed neural networks for nonlinear Schrödinger equations through multi-view transfer learning. *Information Fusion*, 102:102041, 2024.

[2] Kristian Dysthe, Harald E. Krogstad, and Peter Muller. Oceanic rogue waves. *Annual Review of Fluid Mechanics*, 40(1):287–310, 2008.

[3] Shixin Hao, Ulisses Braga-Neto, and Shuonan Dong. Structure preserving pinn for solving time dependent pdes with periodic boundary. *arXiv preprint arXiv:2404.16189*, 2024.

[4] C. Josserand and Y. Pomeau. Generation of vortices in a model of superfluid 4he by the kadomtsev–petviashvili instability. *Europhysics Letters*, 30(1):43–48, 1995.

[5] C. Josserand, Y. Pomeau, and S. Rica. Finite-time localized singularities as a mechanism for turbulent dissipation. *Physical Review Fluids*, 5:123701, 2020.

[6] Natanael Karjanto. The nonlinear schrödinger equation: A mathematical model with its wide-ranging applications. *arXiv preprint arXiv:1912.10683*, 2019.

[7] A. Lechuga. Generation of rogue waves in a wave tank. *Geophysical Research Abstracts*, 14(1), 2012.

[8] D. H. Peregrine. Water waves, nonlinear schrödinger equations and their solutions. *Journal of the Australian Mathematical Society B*, 25:16–43, 1983.

[9] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving pdes. *Journal of Computational Physics*, 378:686–707, 2019.

[10] Kuo Sun and Xinlong Feng. A second-order network structure based on gradient-enhanced physics-informed neural networks for solving parabolic partial differential equations. *Entropy*, 25(4):597, 2023.

[11] Li Wang and Zhenya Yan. Rogue wave formation and interactions in the defocusing nonlinear Schrödinger equation with external potentials. *Applied Mathematics Letters*, 111:106670, 2021.

[12] Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality for training physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 421:116813, 2024.

[13] Xiaoxiao Wu, Kyle S McFall, Haiyang You, and Hadi Meidani. A comprehensive study of non-adaptive and residual-based adaptive sampling for physics-informed neural networks. *Journal of Computational Physics*, 456:111033, 2022.

A Analytical Verification of Ansatz for the NLS

We consider the ansatz for a dark soliton in the defocusing NLS (i.e. $\alpha = 1, g = -1$):
$\psi(x,t) = (\nu \tanh(\nu(x - x_0 - \chi t)) + i\chi)\, e^{-it}$ with $\nu^2 + \chi^2 = 1$.

We have

$$\psi_t(x,t) = -i\left(\nu \tanh(\nu(x - x_0 - \chi t)) + i\chi\left(1 - \frac{\nu^2}{\cosh^2(\nu(x - x_0 - \chi t))}\right)\right) e^{-it}$$

$$|\psi(x,t)|^2 = \nu^2 \tanh^2(\nu(x - x_0 - \chi t)) + \chi^2 = 1 - \frac{\nu^2}{\cosh^2(\nu(x - x_0 - \chi t))}$$

Thus,

$$|\psi|^2\psi = \left(\nu \tanh(\nu(x - x_0 - \chi t)) + i\chi - \frac{\nu^3 \sinh(\nu(x - x_0 - \chi t))}{\cosh^3(\nu(x - x_0 - \chi t))} - i\frac{\nu^2\chi}{\cosh^2(\nu(x - x_0 - \chi t))}\right) e^{-it}$$

Then, one also has:

$$\psi_x(x,t) = \frac{\nu^2}{\cosh^2(\nu(x - x_0 - \chi t))} e^{-it}$$

Thus,

$$\psi_{xx}(x,t) = -2\frac{\nu^3 \sinh(\nu(x - x_0 - \chi t))}{\cosh^3(\nu(x - x_0 - \chi t))} e^{-it}$$

Hence,

$$\frac{1}{2}\psi_{xx} - |\psi|^2\psi = -\left(\nu \tanh(\nu(x - x_0 - \chi t)) + i\chi\left(1 - \frac{\nu^2}{\cosh^2(\nu(x - x_0 - \chi t))}\right)\right) e^{-it}$$

One can thus easily notice that:

$$\frac{1}{2}\psi_{xx} - |\psi|^2\psi = -i\psi_t$$

We hence checked that the given ansatz representing a dark soliton, solution of the 1D inviscid NLS, verfies the equation.

B Conservation Laws for the 1D Nonlinear Schrödinger Equation

The 1D Nonlinear Schrödinger Equation (NLS) is given by:

$$i\frac{\partial \psi}{\partial t} + \frac{\alpha}{2}\frac{\partial^2 \psi}{\partial x^2} + g|\psi|^2\psi = 0,$$

where $\psi(x,t)$ is the complex wavefunction, α is the dispersion coefficient, and g is the nonlinearity parameter.

In this section, we prove the conservation of the number of particles (mass) and the Hamiltonian for this equation.

B.1 Conservation of the Number of Particles

The number of particles is defined as:

$$N = \int |\psi(x,t)|^2 \, dx.$$

To prove that N is conserved, we compute its time derivative:

$$\frac{dN}{dt} = \frac{d}{dt} \int |\psi|^2 \, dx.$$

Expanding $|\psi|^2 = \psi\psi^*$, we have:

$$\frac{d}{dt}|\psi|^2 = \psi^* \frac{\partial \psi}{\partial t} + \psi \frac{\partial \psi^*}{\partial t}.$$

Substituting the NLS equation:

$$\frac{\partial \psi}{\partial t} = -i\frac{\alpha}{2}\frac{\partial^2 \psi}{\partial x^2} - ig|\psi|^2\psi,$$

and its complex conjugate:

$$\frac{\partial \psi^*}{\partial t} = i\frac{\alpha}{2}\frac{\partial^2 \psi^*}{\partial x^2} + ig|\psi|^2\psi^*,$$

into $\frac{dN}{dt}$, we obtain:

$$\begin{aligned}\frac{dN}{dt} &= \int \left(\psi^* \frac{\partial \psi}{\partial t} + \psi \frac{\partial \psi^*}{\partial t}\right) dx \\ &= \int \left[\psi^* \left(-i\frac{\alpha}{2}\frac{\partial^2 \psi}{\partial x^2} - ig|\psi|^2\psi\right) + \psi \left(i\frac{\alpha}{2}\frac{\partial^2 \psi^*}{\partial x^2} + ig|\psi|^2\psi^*\right)\right] dx.\end{aligned}$$

Grouping terms, we find:

$$\frac{dN}{dt} = -i\frac{\alpha}{2} \int \left(\psi^* \frac{\partial^2 \psi}{\partial x^2} - \psi \frac{\partial^2 \psi^*}{\partial x^2}\right) dx - ig \int \left(|\psi|^2\psi\psi^* - |\psi|^2\psi^*\psi\right) dx.$$

The nonlinear term cancels:

$$-ig \int \left(|\psi|^2\psi\psi^* - |\psi|^2\psi^*\psi\right) dx = 0.$$

The dispersive term simplifies using integration by parts:

$$\int \psi^* \frac{\partial^2 \psi}{\partial x^2} dx = \left[\psi^* \frac{\partial \psi}{\partial x}\right]_{-\infty}^{\infty} - \int \frac{\partial \psi^*}{\partial x}\frac{\partial \psi}{\partial x} dx,$$

and similarly for the conjugate term. Assuming boundary terms vanish for localized $\psi(x,t)$, we find:

$$-i\frac{\alpha}{2} \int \left(\psi^* \frac{\partial^2 \psi}{\partial x^2} - \psi \frac{\partial^2 \psi^*}{\partial x^2}\right) dx = 0.$$

Thus:

$$\frac{dN}{dt} = 0 \quad \Rightarrow \quad N \text{ is conserved.}$$

B.2 Conservation of the Hamiltonian

The Hamiltonian is defined as:

$$H = \int \left(\frac{\alpha}{2} \left| \frac{\partial \psi}{\partial x} \right|^2 - \frac{g}{2} |\psi|^4 \right) dx.$$

To prove that H is conserved, we compute its time derivative:

$$\frac{dH}{dt} = \frac{d}{dt} \int \left(\frac{\alpha}{2} \left| \frac{\partial \psi}{\partial x} \right|^2 - \frac{g}{2} |\psi|^4 \right) dx.$$

Split into two terms:

$$\frac{dH}{dt} = \frac{\alpha}{2} \frac{d}{dt} \int \left| \frac{\partial \psi}{\partial x} \right|^2 dx - \frac{g}{2} \frac{d}{dt} \int |\psi|^4 dx.$$

The spatial derivative term is:

$$\int \left| \frac{\partial \psi}{\partial x} \right|^2 dx = \int \frac{\partial \psi}{\partial x} \frac{\partial \psi^*}{\partial x} dx.$$

Taking the time derivative:

$$\frac{d}{dt} \int \left| \frac{\partial \psi}{\partial x} \right|^2 dx = \int \left(\frac{\partial}{\partial t} \frac{\partial \psi}{\partial x} \frac{\partial \psi^*}{\partial x} + \frac{\partial \psi}{\partial x} \frac{\partial}{\partial t} \frac{\partial \psi^*}{\partial x} \right) dx.$$

Using the product rule and commutation of partial derivatives:

$$\frac{\partial}{\partial t} \frac{\partial \psi}{\partial x} = \frac{\partial}{\partial x} \frac{\partial \psi}{\partial t}, \quad \frac{\partial}{\partial t} \frac{\partial \psi^*}{\partial x} = \frac{\partial}{\partial x} \frac{\partial \psi^*}{\partial t},$$

we rewrite:

$$\frac{d}{dt} \int \left| \frac{\partial \psi}{\partial x} \right|^2 dx = \int \frac{\partial}{\partial x} \left(\frac{\partial \psi}{\partial t} \frac{\partial \psi^*}{\partial x} + \frac{\partial \psi^*}{\partial t} \frac{\partial \psi}{\partial x} \right) dx.$$

Applying the divergence theorem:

$$\int \frac{\partial}{\partial x} \left(\frac{\partial \psi}{\partial t} \frac{\partial \psi^*}{\partial x} + \frac{\partial \psi^*}{\partial t} \frac{\partial \psi}{\partial x} \right) dx = \left[\frac{\partial \psi}{\partial t} \frac{\partial \psi^*}{\partial x} + \frac{\partial \psi^*}{\partial t} \frac{\partial \psi}{\partial x} \right]_{-\infty}^{\infty}.$$

Assuming $\psi(x,t)$ has zero-slope at the boundaries, the spatial derivatives vanish at $\pm L$:

$$\frac{d}{dt} \int \left| \frac{\partial \psi}{\partial x} \right|^2 dx = 0.$$

The nonlinear term is:

$$\int |\psi|^4 dx = \int (\psi \psi^*)^2 dx.$$

Taking the time derivative:

$$\frac{d}{dt} \int |\psi|^4 dx = \int 2|\psi|^2 \frac{d}{dt} |\psi|^2 dx.$$

Using $\frac{d}{dt}|\psi|^2 = \psi^* \frac{\partial \psi}{\partial t} + \psi \frac{\partial \psi^*}{\partial t}$, we write:

$$\frac{d}{dt}\int |\psi|^4 dx = 2\int |\psi|^2 \left(\psi^* \frac{\partial \psi}{\partial t} + \psi \frac{\partial \psi^*}{\partial t}\right) dx.$$

Substitute the NLS equation:

$$\frac{\partial \psi}{\partial t} = -i\alpha \frac{\partial^2 \psi}{\partial x^2} - ig|\psi|^2\psi, \quad \frac{\partial \psi^*}{\partial t} = i\alpha \frac{\partial^2 \psi^*}{\partial x^2} + ig|\psi|^2\psi^*.$$

The nonlinear terms cancel when substituted into the integral:

$$2\int |\psi|^2 \left(\psi^* \frac{\partial \psi}{\partial t} + \psi \frac{\partial \psi^*}{\partial t}\right) dx = 0.$$

Combining the two terms:

$$\frac{dH}{dt} = \frac{\alpha}{2}\frac{d}{dt}\int \left|\frac{\partial \psi}{\partial x}\right|^2 dx - \frac{g}{2}\frac{d}{dt}\int |\psi|^4 dx = 0.$$

Thus, H is conserved.

B.3 Conservation of the Momentum

The momentum is defined as:

$$P = \Im\left(\int \psi^* \nabla \psi \, dx\right),$$

where $\psi(x,t)$ is the wavefunction, ψ^* is its complex conjugate, and ∇ represents the spatial derivative.

Differentiating P with respect to time t, we get:

$$\frac{dP}{dt} = \Im\left(\frac{d}{dt}\int \psi^* \nabla \psi \, dx\right).$$

Using the product rule, this becomes:

$$\frac{dP}{dt} = \Im\left(\int \frac{\partial \psi^*}{\partial t} \nabla \psi \, dx + \int \psi^* \nabla \frac{\partial \psi}{\partial t} \, dx\right).$$

From the NLS equation, we have:

$$\frac{\partial \psi}{\partial t} = i\left(\frac{\alpha}{2}\nabla^2 \psi + g|\psi|^2 \psi\right),$$

and its complex conjugate:

$$\frac{\partial \psi^*}{\partial t} = -i\left(\frac{\alpha}{2}\nabla^2 \psi^* + g|\psi|^2 \psi^*\right).$$

Substituting these into $\frac{dP}{dt}$, we obtain:

$$\frac{dP}{dt} = \Im\left(\int \left[-i\left(\frac{\alpha}{2}\nabla^2 \psi^* + g|\psi|^2\psi^*\right)\nabla\psi + \psi^* \nabla\left(i\left(\frac{\alpha}{2}\nabla^2\psi + g|\psi|^2\psi\right)\right)\right] dx\right).$$

Linear Terms: The linear dispersive term expands as:

$$-i\frac{\alpha}{2}\int \nabla^2\psi^*\nabla\psi\,dx + i\frac{\alpha}{2}\int \psi^*\nabla\nabla^2\psi\,dx.$$

Using integration by parts and assuming periodic or decaying boundary conditions (so boundary terms vanish), these terms cancel. Thus, the linear contribution to $\frac{dP}{dt}$ is zero.

Nonlinear Terms:

$$-ig\int |\psi|^2\psi^*\nabla\psi\,dx + ig\int \psi^*\nabla\left(|\psi|^2\psi\right)\,dx.$$

The second term expands further:

$$ig\int \psi^*\left(\nabla|\psi|^2\psi + |\psi|^2\nabla\psi\right)\,dx.$$

Again, using integration by parts, terms involving $\nabla|\psi|^2$ cancel due to symmetry, and the remaining terms cancel under boundary conditions.

Since both the linear and nonlinear contributions vanish, we conclude:

$$\frac{dP}{dt} = 0.$$

Thus, the momentum P is conserved.

C Code used for simulations

All the code necessary to obtain these results and plots can be found on the following GitHub repository: https://github.com/peter-sayegh/PINNs-NLSE-RW.git, assuming the jax library is installed already.

D Results and Error Data for the focusing RW simulation

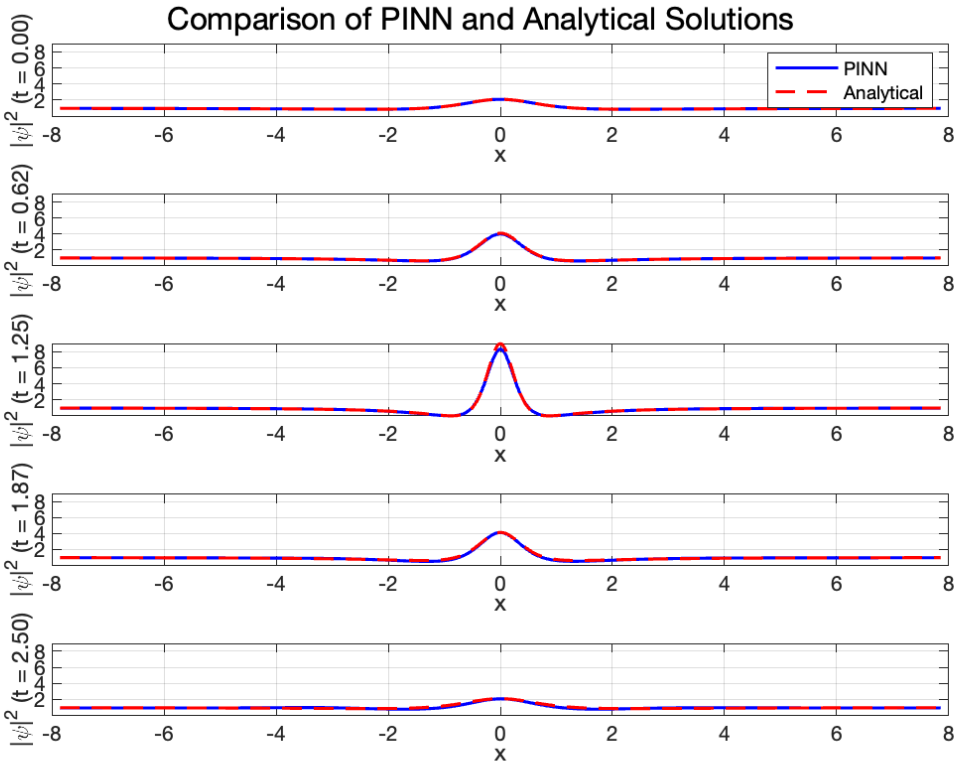


Figure 13: Evolution of $|\psi|^2$ given by the PINN and compared with the exact solution, for $\alpha = 1, g = 1$(focusing case),$L = 2.5\pi, T = 2.5, N_x = 600, N_t = 250, l = 5, w = 48, (\lambda_r, \lambda_{IC}, \lambda_{BC}) = (1, 10, 10), \epsilon = 5, \gamma = 0.8, (N_r, N_{IC}, N_{BC}) = (2048, 300, 200), N_{iter} = 35000$

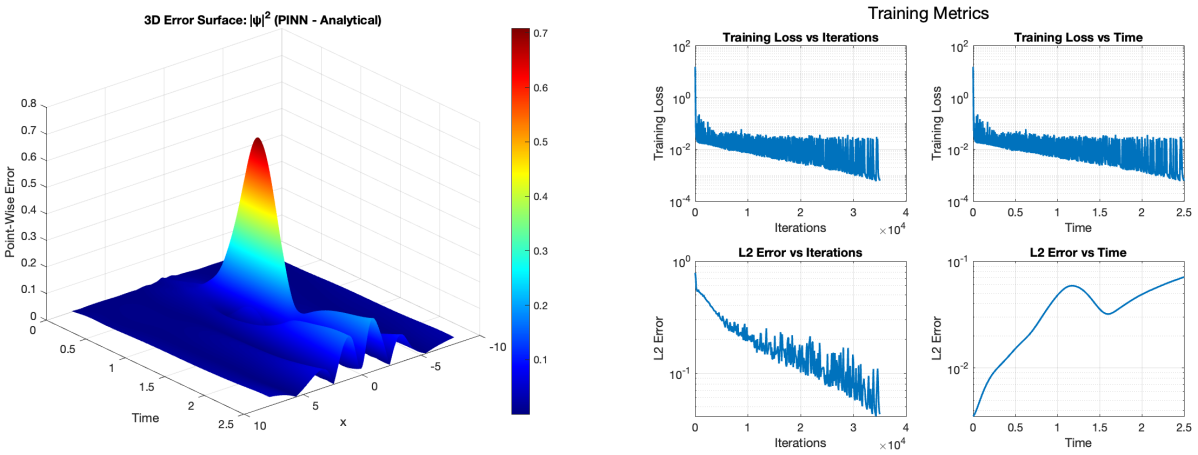


Figure 14: Pointwise Error and training metrics evolution in the **focusing** Rogue Wave simulation

Statement of Academic Integrity Regarding Plagiarism

I, the undersigned ..... Sayegh, Peter Albert ..... [family name, given name(s)], hereby certify on my honor that:

1. The results presented in this report are the product of my own work.

2. I am the original creator of this report.

3. I have not used sources or results from third parties without clearly stating thus and referencing them according to the recommended rules for providing bibliographic information.

Declaration to be copied below:

I hereby declare that this work contains no plagiarized material.

Date ..... 14.03.2025 .....

Signature